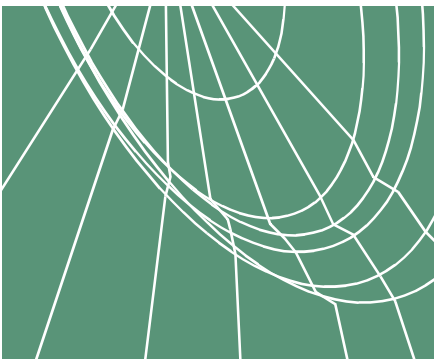




MSC.Marc Volume A



Theory and User Information

Version 2003

Copyright © 2003 MSC.Software Corporation

All rights reserved. Printed in U.S.A.

Corporate

MSC.Software Corporation
2 MacArthur Place
Santa Ana, CA 92707
Telephone: (714) 540-8900
Fax: (714) 784-4056

Asia Pacific

MSC Japan Ltd.
Entsuji-Gadelius Building
2-39, Akasaka 5-chome
Minato-ku, Tokyo 107-0052, JAPAN
Telephone: (81) (3) 3505 0266
Fax: (81) (3) 3505 0914

Europe

MSC.Software GmbH
Am Moosfeld
81829 München, GERMANY
Telephone: (49) (89) 431 987 0
Fax: (49) (89) 436 1716

Worldwide Web

www.mscsoftware.com

Part Number: MA*V2003*Z*Z*Z*DC-VOL-A

Disclaimer

THE CONCEPTS, METHODS, AND EXAMPLES PRESENTED IN THE DOCUMENTATION ARE FOR ILLUSTRATIVE AND EDUCATIONAL PURPOSES ONLY, AND ARE NOT INTENDED TO BE EXHAUSTIVE OR TO APPLY TO ANY PARTICULAR ENGINEERING PROBLEM OR DESIGN. USER ASSUMES ALL RISKS AND LIABILITY FOR RESULTS OBTAINED BY THE USE OF THE COMPUTER PROGRAMS DESCRIBED HEREIN. IN NO EVENT SHALL MSC.SOFTWARE CORPORATION BE LIABLE TO ANYONE FOR ANY SPECIAL, COLLATERAL, INCIDENTAL, INDIRECT OR CONSEQUENTIAL DAMAGES ARISING OUT OF, RESULTING FROM, OR IN CONNECTION WITH USE OF THE CONTENTS OR INFORMATION IN THE DOCUMENTATION.

MSC.SOFTWARE CORPORATION ASSUMES NO LIABILITY OR RESPONSIBILITY FOR ANY ERRORS THAT MAY APPEAR IN THE DOCUMENTATION. THE DOCUMENTATION IS PROVIDED ON AN “AS-IS” BASIS AND ALL EXPRESS AND IMPLIED CONDITIONS, REPRESENTATIONS AND WARRANTIES, INCLUDING ANY IMPLIED WARRANTY OF MERCHANTABILITY OR FITNESS FOR A PARTICULAR PURPOSE, ARE DISCLAIMED, EXCEPT TO THE EXTENT THAT SUCH DISCLAIMERS ARE HELD TO BE LEGALLY INVALID.

MSC.SOFTWARE CORPORATION RESERVES THE RIGHT TO MAKE CHANGES IN SPECIFICATIONS AND OTHER INFORMATION CONTAINED IN THE DOCUMENTATION WITHOUT PRIOR NOTICE.

Trademarks

MSC, Dytran, MARC, and Patran are registered trademarks of MSC.Software Corporation or its subsidiaries in the United States and/or other countries. MSC., MSC.Dytran, MSC.Marc, and MSC.Patran are trademarks of MSC.Software Corporation.

NASTRAN is a registered trademark of the National Aeronautics and Space Administration. MSC.Nastran is an enhanced proprietary version developed and maintained by MSC.Software Corporation. All other trademarks are the property of their respective owners.

Third Party Software Program Credits

METIS is copyrighted by the regents of the University of Minnesota.

NT-MPICH is developed by Lehrstuhl für Betriebssysteme der RWTH Aachen. Copyright © 1992-2003 Lehrstuhl für Betriebssysteme der RWTH Aachen.

Government Use

Use, duplication, or disclosure by the U.S. Government is subject to restrictions as set forth in FAR 12.212 (Commercial Computer Software) and DFARS 227.7202 (Commercial Computer Software and Commercial Computer Software Documentation), as applicable.

C O N T E N T S

MSC.Marc Volume A: Theory and User Information

Preface

- About This Manual, xvii
- Purpose of Volume A, xviii
- Contents of Volume A, xviii
- How to Use This Manual, xvix

Chapter 1

The MSC.Marc System

- MSC.Marc Programs, 1-3
 - MSC.Marc for Analysis, 1-3
 - MSC.Marc Mentat or MSC.Patran for GUI, 1-3
- Structure of MSC.Marc, 1-5
 - Procedure Library, 1-5
 - Material Library, 1-5
 - Element Library, 1-5
 - Program Function Library, 1-5
- Features and Benefits of MSC.Marc, 1-6

Chapter 2

Program Initiation

- MSC.Marc Host Systems, 2-2
- Workspace Requirements, 2-3
 - MSC.Marc Workspace Requirements, 2-3
- File Units, 2-6
- Program Initiation, 2-9
- Examples of Running MSC.Marc Jobs, 2-12

Chapter 3

Data Entry

Chapter 4

Introduction to Mesh Definition

- Input Conventions, 3-2
 - Input of List of Items, 3-3
 - Examples of Lists, 3-5
- Parameters, 3-6
- Model Definition Options, 3-7
- History Definition Options, 3-8
- REZONE Option, 3-10
- Direct Input, 4-3
 - Element Connectivity Data, 4-3
 - Nodal Coordinate Data, 4-7
 - Activate/Deactivate, 4-7
- User Subroutine Input, 4-8
- MESH2D, 4-9
 - Block Definition, 4-9
 - Merging of Nodes, 4-10
 - Block Types, 4-10
 - Symmetry, Weighting, and Constraints, 4-14
 - Additional Options, 4-15
- MSC.Marc Mentat, 4-16
- FXORD Option, 4-17
 - Major Classes of the FXORD Option, 4-17
 - Recommendations on Use of the FXORD Option, 4-22
- Incremental Mesh Generators, 4-23
- Bandwidth Optimization, 4-24
- Rezoning, 4-25
- Substructure, 4-26
 - Technical Background, 4-27
 - Scaling Element Stiffness, 4-29
- BEAM SECT Parameter, 4-30
 - Orientation of the Section in Space, 4-30
 - Definition of the Section, 4-30

Chapter 5

Structural Procedure Library

- Error Analysis, 4-33
- Local Adaptivity, 4-34
 - Number of Elements Created, 4-34
 - Boundary Conditions, 4-35
 - Location of New Nodes, 4-36
 - Adaptive Criteria, 4-38
- Global Remeshing, 4-41
 - Remeshing Criteria, 4-44
 - Remeshing Techniques, 4-46
 - 3-D Surface Extraction and Meshing, 4-51
- Linear Analysis, 5-3
 - Accuracy, 5-5
 - Error Estimates, 5-5
 - Adaptive Meshing, 5-5
 - Fourier Analysis, 5-6
- Nonlinear Analysis, 5-10
 - Geometric Nonlinearities, 5-15
 - Updated Lagrangian Procedure, 5-23
 - Nonlinear Boundary Conditions, 5-27
 - Buckling Analysis, 5-30
 - Perturbation Analysis, 5-31
 - Computational Procedures for Elastic-Plastic Analysis, 5-38
 - CREEP, 5-52
 - AUTO THERM CREEP (Automatic Thermally Loaded Elastic-Creep/Elastic-Plastic-Creep Stress Analysis), 5-57
 - Viscoelasticity, 5-58
 - Viscoplasticity, 5-59
- Fracture Mechanics, 5-61
 - Linear Fracture Mechanics, 5-61
 - Nonlinear Fracture Mechanics, 5-63
 - Numerical Evaluation of the J-integral, 5-65
 - Modeling Considerations, 5-67
 - Dynamic Crack Propagation, 5-68
 - Dynamic Fracture Methodology, 5-69

- Dynamics, 5-70
 - Eigenvalue Analysis, 5-70
 - Transient Analysis, 5-74
 - Harmonic Response, 5-84
 - Spectrum Response, 5-88
- Rigid-Plastic Flow, 5-90
 - Steady State Analysis, 5-91
 - Transient Analysis, 5-91
 - Technical Background, 5-91
- Superplasticity, 5-93
- Soil Analysis, 5-97
 - Technical Formulation, 5-98
- Design Sensitivity Analysis, 5-102
 - Theoretical Considerations, 5-104
- Design Optimization, 5-106
 - Approximation of Response Functions Over the Design Space, 5-107
 - Improvement of the Approximation, 5-110
 - The Optimization Algorithm, 5-110
 - MSC.Marc User Interface for Sensitivity Analysis and Optimization, 5-111
- Transfer Axisymmetric Analysis Data to 3-D Analysis, 5-115
 - Load and Displacement Boundary Conditions Transfer from Axisymmetric Analysis to 3-D using Curve Shift, 5-115
- Steady State Rolling Analysis, 5-118
 - Kinematics, 5-118
 - Inertia Effect, 5-120
 - Rolling Contact, 5-121
 - Steady State Rolling with MSC.Marc, 5-121
- References, 5-123

Chapter 6

Nonstructural Procedure Library

- Heat Transfer, 6-3
 - Thermal Contact, 6-5
 - Convergence Controls, 6-5
 - Steady State Analysis, 6-6
 - Transient Analysis, 6-6
 - Temperature Effects, 6-8
 - Initial Conditions, 6-10
 - Boundary Conditions, 6-10
 - Radiation Viewfactors, 6-12
 - Conrad Gap, 6-19
 - Channel, 6-21
 - Output, 6-22
- Hydrodynamic Bearing, 6-25
 - Technical Background, 6-27
- Electrostatic Analysis, 6-30
 - Technical Background, 6-31
- Magnetostatic Analysis, 6-33
 - Technical Background, 6-34
- Electromagnetic Analysis, 6-39
 - Technical Background, 6-40
- Piezoelectric Analysis, 6-44
 - Technical Background, 6-45
 - Strain Based Piezoelectric Coupling, 6-47
- Acoustic Analysis, 6-49
 - Rigid Cavity Acoustic Analysis, 6-49
 - Technical Background, 6-49
 - Coupled Acoustic-Structural Analysis, 6-51
 - Technical Background, 6-51

- Fluid Mechanics, 6-56
 - Finite Element Formulation, 6-59
 - Penalty Method, 6-62
 - Steady State Analysis, 6-63
 - Transient Analysis, 6-63
 - Solid Analysis, 6-63
 - Solution of Coupled Problems in Fluids, 6-64
 - Degrees of Freedom, 6-64
 - Element Types, 6-65
- Coupled Analyses, 6-67
 - Thermal Mechanically Coupled Analysis, 6-69
 - Fluid/Solid Interaction – Added Mass Approach, 6-71
 - Coupled Thermal-Electrical Analysis (Joule Heating), 6-73
 - Coupled Electrical-Thermal-Mechanical Analysis, 6-76
- References, 6-79
- Linear Elastic Material, 7-3
- Composite Material, 7-6
 - Layered Materials, 7-7
 - Material Preferred Direction, 7-10
 - Material Dependent Failure Criteria, 7-14
 - Interlaminar Shear for Thick Shell and Beam Elements, 7-20
 - Interlaminar Stresses for Continuum Composite Elements, 7-21
 - Progressive Composite Failure, 7-22
- Gasket, 7-23
 - Constitutive Model, 7-23
- Nonlinear Hypoelastic Material, 7-30
- Thermo-Mechanical Shape Memory Model, 7-31
 - Transformation Induced Deformation, 7-32
 - Constitutive Theory, 7-34
 - Phase Transformation Strains, 7-35
 - Experimental Data Fitting for Thermo-mechanical Shape Memory Alloy, 7-37

Chapter 7

Material Library

- Mechanical Shape Memory Model, 7-43
 - Experimental Data Fitting for Mechanical Shape Memory Alloy, 7-47
- Elastomer, 7-48
 - Updated Lagrange Formulation for Nonlinear Elasticity, 7-59
- Time-independent Inelastic Behavior, 7-61
 - Yield Conditions, 7-64
 - Mohr-Coulomb Material (Hydrostatic Stress Dependence), 7-70
 - Buyukozturk Criterion (Hydrostatic Stress Dependence), 7-72
 - Powder Material, 7-73
 - Workhardening Rules, 7-76
 - Flow Rule, 7-83
 - Constitutive Relations, 7-84
 - Time-independent Cyclic Plasticity, 7-88
- Time-dependent Inelastic Behavior, 7-92
 - Creep (Maxwell Model), 7-97
 - Oak Ridge National Laboratory Laws, 7-101
 - Swelling, 7-103
 - Viscoplasticity (Explicit Formulation), 7-104
 - Time-dependent Cyclic Plasticity, 7-105
 - Viscoelastic Material, 7-107
 - Narayanaswamy Model, 7-120
- Temperature Effects and Coefficient of Thermal Expansion, 7-126
 - Piecewise Linear Representation, 7-126
 - Temperature-Dependent Creep, 7-128
 - Coefficient of Thermal Expansion, 7-128
- Time-Temperature-Transformation, 7-130
- Low Tension Material, 7-133
 - Uniaxial Cracking Data, 7-133
 - Low Tension Cracking, 7-134
 - Tension Softening, 7-134
 - Crack Closure, 7-135
 - Crushing, 7-135
 - Analysis, 7-135

- Soil Model, 7-136
 - Elastic Models, 7-136
 - Cam-Clay Model, 7-137
 - Evaluation of Soil Parameters for the Critical State Soil Model, 7-139
- Damage Models, 7-152
 - Ductile Metals, 7-152
 - Elastomers, 7-155
- Nonstructural Materials, 7-158
 - Heat Transfer Analysis, 7-158
 - Thermo-Electrical Analysis, 7-158
 - Electrical-Thermal-Mechanical Analysis, 7-158
 - Hydrodynamic Bearing Analysis, 7-158
 - Fluid/Solid Interaction Analysis – Added Mass Approach, 7-159
 - Electrostatic Analysis, 7-159
 - Magnetostatic Analysis, 7-159
 - Electromagnetic Analysis, 7-159
 - Piezoelectric Analysis, 7-159
 - Acoustic Analysis, 7-160
 - Fluid Analysis, 7-160
- References, 7-161

Chapter 8

Contact

- Numerical Procedures, 8-3
 - Lagrange Multipliers, 8-3
 - Penalty Methods, 8-3
 - Hybrid and Mixed Methods, 8-4
 - Direct Constraints, 8-4
- Definition of Contact Bodies, 8-5
- Numbering of Contact Bodies, 8-9
- Motion of Bodies, 8-10
 - Initial Conditions, 8-12
- Detection of Contact, 8-13
 - Shell Contact, 8-15
 - Neighbor Relations, 8-17

- Implementation of Constraints, 8-18
- Friction Modeling, 8-21
 - Glue Model, 8-28
- Separation, 8-29
 - Release, 8-30
- Coupled Analysis, 8-31
- Thermal Contact, 8-34
- Element Considerations, 8-35
 - 2-D Beams, 8-36
 - 3-D Beams, 8-36
 - Shell Elements, 8-38
- Dynamic Impact, 8-39
- Global Adaptive Meshing and Rezoning, 8-40
- Adaptive Meshing, 8-41
- Result Evaluation, 8-42
- Tolerance Values, 8-43
- Workspace Reservation, 8-45
- Mathematical Aspects of Contact, 8-46
 - Lagrange Multiplier Procedure, 8-46
 - Direct Constraint Procedure, 8-47
 - Solution Strategy for Deformable Contact, 8-56
 - Friction Modeling, 8-58
 - Iterative Penetration Checking, 8-60
 - Instabilities, 8-62
- References, 8-63

Chapter 9

Boundary Conditions

- Loading, 9-3
 - Loading Types, 9-3
 - Fluid Drag and Wave Loads, 9-6
 - Mechanical Loads, 9-8
 - Cavity Pressure Loading, 9-9
 - Thermal Loads, 9-14
 - Initial Stress and Initial Plastic Strain, 9-15
 - Heat Fluxes, 9-16
 - Mass Fluxes and Restrictors, 9-17
 - Electrical Currents, 9-18
 - Electrostatic Charges, 9-18
 - Acoustic Sources, 9-19
 - Piezoelectric Loads, 9-19
 - Magnetostatic Currents, 9-19
 - Electromagnetic Currents and Charges, 9-20
- Kinematic Constraints, 9-21
 - Boundary Conditions, 9-21
 - Transformation of Degree of Freedom, 9-22
 - Shell Transformation, 9-23
 - Tying Constraint, 9-26
 - Rigid Link Constraint, 9-35
 - Shell-to-Solid Tying, 9-36
 - Insert, 9-37
 - Support Conditions, 9-37
 - Cyclic Symmetry, 9-38
 - MSC.Nastran RBE2 and RBE3, 9-41

Chapter 10

Element Library

- Truss Elements, 10-18
- Membrane Elements, 10-19
- Continuum Elements, 10-20
- Beam Elements, 10-21
- Plate Elements, 10-22
- Shell Elements, 10-23

- Heat Transfer Elements, 10-24
 - Acoustic Analysis, 10-24
 - Electrostatic Analysis, 10-24
 - Fluid/Solid Interaction, 10-24
 - Hydrodynamic Bearing Analysis, 10-25
 - Magnetostatic Analysis, 10-25
- Electromagnetic Analysis, 10-26
- Soil Analysis, 10-27
- Fluid Analysis, 10-28
- Piezoelectric Analysis, 10-29
- Special Elements, 10-30
 - Gap-and-Friction Elements, 10-30
 - Pipe-bend Element, 10-30
 - Curved-pipe Element, 10-30
 - Shear Panel Element, 10-30
 - Cable Element, 10-31
 - Rebar Elements, 10-31
- Incompressible Elements, 10-32
 - Large Strain Elasticity, 10-32
 - Large Strain Plasticity, 10-33
 - Rigid-Plastic Flow, 10-33
- Constant Dilatation Elements, 10-34
- Reduced Integration Elements, 10-35
- Continuum Composite Elements, 10-36
- Fourier Elements, 10-37
- Semi-infinite Elements, 10-38
- Cavity Surface Elements, 10-39
- Assumed Strain Formulation, 10-40
- Follow Force Stiffness Contribution, 10-41
- Explicit Dynamics, 10-42
- Adaptive Mesh Refinement, 10-43
- References, 10-44

Chapter 11

Solution Procedures for Nonlinear Systems

- Considerations for Nonlinear Analysis, 11-3
 - Behavior of Nonlinear Materials, 11-4
 - Scaling the Elastic Solution, 11-4
 - Load Incrementation, 11-4
 - Selecting Load Increment Size, 11-7
 - Fixed Load Incrementation, 11-7
 - Automatic Load Incrementation, 11-7
 - Residual Load Correction, 11-16
 - Restarting the Analysis, 11-17
- Full Newton-Raphson Algorithm, 11-19
- Modified Newton-Raphson Algorithm, 11-21
- Strain Correction Method, 11-22
- The Secant Method, 11-24
- Direct Substitution, 11-26
 - Load Correction, 11-26
- Arc-length Methods, 11-27
- Remarks, 11-35
- Convergence Controls, 11-36
- Singularity Ratio, 11-39
- Solution of Linear Equations, 11-41
 - Direct Methods, 11-41
 - Iterative Methods, 11-42
 - Preconditioners, 11-42
 - Storage Methods, 11-43
 - Nonsymmetric Systems, 11-43
 - Complex Systems, 11-44
 - Iterative Solvers, 11-44
 - Basic Theory, 11-44
- Flow Diagram, 11-47
- References, 11-48

Chapter 12

Output Results

- Workspace Information, 12-3
- Increment Information, 12-7
 - Summary of Loads, 12-7
 - Timing Information, 12-7
 - Singularity Ratio, 12-7
 - Convergence, 12-7
- Selective Printout, 12-9
 - Options, 12-9
 - User Subroutines, 12-11
- Restart, 12-12
- Element Information, 12-13
 - Solid (Continuum) Elements, 12-14
 - Shell Elements, 12-14
 - Beam Elements, 12-16
 - Heat Transfer Elements, 12-17
 - Gap Elements, 12-17
 - Linear and Nonlinear Springs, 12-17
 - Hydrodynamic Bearing, 12-17
- Nodal Information, 12-18
 - Stress Analysis, 12-18
 - Reaction Forces, 12-18
 - Residual Loads, 12-18
 - Dynamic Analysis, 12-18
 - Heat Transfer Analysis, 12-19
 - Rigid-Plastic Analysis, 12-19
 - Hydrodynamic Bearing Analysis, 12-19
 - Electrostatic Analysis, 12-19
 - Magnetostatic Analysis, 12-19
 - Electromagnetic Analysis, 12-19
 - Piezoelectric Analysis, 12-19
 - Acoustic Analysis, 12-19
 - Contact Analysis, 12-19
- Post File, 12-21
- Forming Limit Parameter (FLP), 12-22
- Program Messages, 12-26

Chapter 13

Parallel Processing

Appendix A

Finite Element Technology in MSC.Marc

Appendix B

Finite Element Analysis of NC Machining Processes

Index

- MSC.Marc HyperMesh Results Interface, 12-27
- MSC.Marc SDRC I-DEAS Results Interface, 12-28
- MSC.Marc ADAMS Results Interface, 12-29
- Status File, 12-31

- Different Types of Machines, 13-2
- Supported and Unsupported Features, 13-2
- Matrix Solvers, 13-3
- Contact, 13-4
- Domain Decomposition, 13-5
 - Running a Parallel Job, 13-5

- Governing Equations of Various Structural Procedures, A-1
- System and Element Stiffness Matrices, A-5
- Load Vectors, A-7
- References, A-8

- General Description, B-1
- NC Files (Cutter Shape and Cutter Path Definition), B-2
- Intersection Between Finite Element Mesh and Cutter, B-3
- Deactivation of Elements, B-3
- References, B-4

Preface

- [About This Manual](#)
- [Purpose of Volume A](#)
- [Contents of Volume A](#)
- [How to Use This Manual](#)

About This Manual

This manual is MSC.Marc Volume A, the first in a series of six volumes documenting the MSC.Marc Finite Element program. The documentation of MSC.Marc is summarized below. You will find references to these documents throughout this manual.

MSC.Marc Documentation

TITLE	VOLUME
<i>Theory and User Information</i>	Volume A
<i>Element Library</i>	Volume B
<i>Program Input</i>	Volume C
<i>User Subroutines and Special Routines</i>	Volume D
<i>Demonstration Problems</i>	Volume E

Purpose of Volume A

The purpose of this volume is:

1. To help you define your finite element problem by describing MSC.Marc's capabilities to model physical problems.
2. To identify and describe complex engineering problems and introduce MSC.Marc's scope and capabilities for solving these problems.
3. To assist you in accessing MSC.Marc features that are applicable to your particular problems and to provide you with references to the rest of the MSC.Marc literature.
4. To provide you with the theoretical basis of the computational techniques used to solve the problem.

Contents of Volume A

This volume describes how to use MSC.Marc. It explains the capabilities of MSC.Marc and gives pertinent background information. The principal categories of information are found under the following titles:

- [Chapter 1 The MSC.Marc System](#)
- [Chapter 2 Program Initiation](#)
- [Chapter 3 Data Entry](#)
- [Chapter 4 Introduction to Mesh Definition](#)
- [Chapter 5 Structural Procedure Library](#)
- [Chapter 6 Nonstructural and Coupled Procedure Library](#)
- [Chapter 7 Material Library](#)

Chapter 8	Contact
Chapter 9	Boundary Conditions
Chapter 10	Element Library
Chapter 11	Solution Procedures for Nonlinear Systems
Chapter 12	Output Results
Chapter 13	Parallel Processing
Appendix A	Finite Element Technology in MSC.Marc

The information in this manual is both descriptive and theoretical. You will find engineering mechanics discussed in some detail. You will also find specific instructions for operating the various options offered by MSC.Marc.

How to Use This Manual

Volume A organizes the features and operations of the MSC.Marc program sequentially. This organization represents a logical approach to problem solving using Finite Element Analysis. First, the database is entered into the system, as described in Chapter 3. Next, a physical problem is defined in terms of a mesh overlay. Techniques for mesh definition are described in Chapter 4. Chapters 5 and 6 describe the various structural analyses that can be performed by MSC.Marc, while Chapter 7 describes the material models that are available in MSC.Marc. Chapter 8 describes the contact capabilities. Chapter 9 discusses constraints, in the form of boundary conditions. Chapter 10 explains the type of elements that can be used to represent the physical problem. Chapter 11 describes the numerical procedures for solving nonlinear equations. Finally, the results of the analysis, in the form of outputs, are described in Chapter 12.

This volume is also designed as a reference source. This means that all users will not need to refer to each section of the manual with the same frequency or in the same sequence.

CHAPTER

1

The MSC.Marc System

- MSC.Marc Programs
- Structure of MSC.Marc
- Features and Benefits of MSC.Marc

The MSC.Marc system contains a series of integrated programs that facilitate analysis of engineering problems in the fields of structural mechanics, heat transfer, and electromagnetics. The MSC.Marc system consists of the following programs:

- [MSC.Marc for Analysis](#)
- [MSC.Marc Mentat or MSC.Patran for GUI](#) (For a detailed description of the supported functionalities by MSC.Patran, refer to the *MSC.Patran Marc Preference Guide*.)

These programs work together to:

- Generate geometric information that defines your structure (MSC.Marc and MSC.Marc Mentat or MSC.Patran)
- Analyze your structure (MSC.Marc)
- Graphically depict the results (MSC.Marc and MSC.Marc Mentat or MSC.Patran)

Figure 1-1 shows the interrelationships among these programs. [MSC.Marc Programs](#) discusses the MSC.Marc component programs.

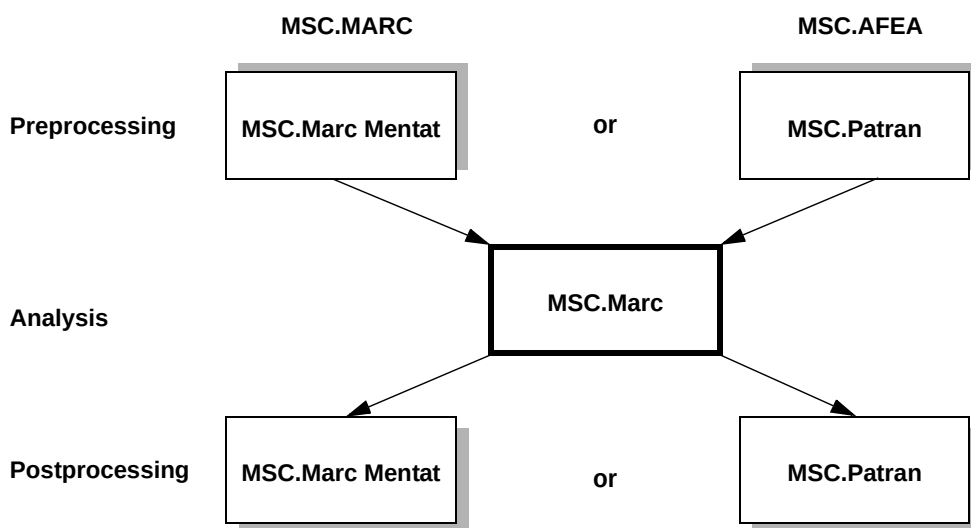


Figure 1-1 The MSC.Marc System

MSC.Marc Programs

MSC.Marc for Analysis

You can use MSC.Marc to perform linear or nonlinear stress analysis in the static and dynamic regimes, to perform heat transfer analysis and electromagnetic analysis. The nonlinearities may be due to either material behavior, large deformation, or boundary conditions. An accurate representation accounts for these nonlinearities.

Physical problems in one, two, or three dimensions can be modeled using a variety of elements. These elements include trusses, beams, shells, and solids. Mesh generators, graphics, and postprocessing capabilities, which assist you in the preparation of input and the interpretation of results, are all available in MSC.Marc. The equations governing mechanics and implementation of these equations in the finite element method are discussed in Chapters [5](#), [6](#), [7](#), [8](#), and [11](#).

MSC.Marc Mentat or MSC.Patran for GUI

MSC.Marc Mentat is an interactive computer program that prepares and processes data for use with the finite element method. Interactive computing can significantly reduce the human effort needed for analysis by the finite element method. Graphical presentation of data further reduces this effort by providing an effective way to review the large quantity of data typically associated with finite element analysis.

An important aspect of MSC.Marc Mentat is that you can interact directly with the program. MSC.Marc Mentat verifies keyboard input and returns recommendations or warnings when it detects questionable input. MSC.Marc Mentat checks the contents of input files and generates warnings about its interpretation of the data if the program suspects that it may not be processing the data in the manner in which you, the user, have assumed. MSC.Marc Mentat allows you to graphically verify any changes the input generates.

MSC.Marc Mentat can process both two- and three-dimensional meshes to do the following:

- Generate and display a mesh
- Generate and display boundary conditions and loadings
- Perform postprocessing to generate contour, deformed shape, and time history plots

The data that is processed includes:

- Nodal coordinates

- Element connectivity

- Nodal boundary conditions

- Nodal coordinate systems

- Element material properties

- Element geometric properties

- Element loads

- Nodal loads/nonzero boundary conditions

- Element and nodal sets

Structure of MSC.Marc

MSC.Marc has four comprehensive libraries, making the program applicable to a wide range of uses. These libraries contain structural procedures, materials, elements, and program functions. The contents of each library are described below.

Procedure Library

The structural procedure library contains procedures such as static, dynamic, creep, buckling, heat transfer, fluid mechanics, and electromagnetic analysis.

The procedure library conveniently relates these various structural procedures to physical phenomena while guiding you through modules that allow, for example, nonlinear dynamic and heat-transfer analyses.

Material Library

The material library includes many material models that represent most engineering materials. Examples are the inelastic behavior of metals, soils, and rubber material. Many models exhibit nonlinear properties such as plasticity, viscoelasticity, and hypoelasticity. Linear elasticity is also included. All properties may depend on temperature.

Element Library

The element library contains 180 elements. This library lets you describe any geometry under any linear or nonlinear loading conditions.

Program Function Library

The program functions such as selective assembly, user-supplied subroutines, and restart, are tailored for user-friendliness and are designed to speed up and simplify analysis work. MSC.Marc allows you to combine any number of components from each of the four libraries and, in doing so, puts at your disposal the tools to solve almost any structural mechanics problem.

Features and Benefits of MSC.Marc

Since the mid-1970s, MSC.Marc has been recognized as the premier general purpose program for nonlinear finite element analysis. The program's modularity leads to its broad applicability. All components of the structural procedure, material, and element libraries are available for use, allowing virtually unlimited flexibility and adaptability.

MSC.Marc has helped analyze and influence final design decisions on

Automotive parts

Nuclear reactor housings

Biomedical equipment

Offshore platform components

Coated fiberglass fabric roof
structures

Rocket motor casings

Ship hulls

Elastomeric motor mounts

Space vehicles

Electronic components

Steam-piping systems

Engine pistons

Tires

Jet engine rotors

Welding, casting, and quenching
processes

Large strain metal extrusions

MSC.Marc's clients gained the following benefits not attainable through other numerical or experimental techniques. These benefits include:

- Accurate results for both linear and nonlinear analysis

- Better designs, which result in improved performance and reliability

- The ability to model complex structures and to incorporate geometric and material nonlinear behavior

- Documentation, technical support, consulting, and education provided by MSC.Software Corporation

- Availability of MSC.Marc on most computers from workstations to supercomputers

- Efficient operation

CHAPTER

2

Program Initiation

- [MSC.Marc Host Systems](#)
- [Workspace Requirements](#)
- [File Units](#)
- [Program Initiation](#)
- [Examples of Running MSC.Marc Jobs](#)

Chapter 2 explains how to execute MSC.Marc on your computer. MSC.Marc runs on many types of machines. All MSC.Marc capabilities are available on each type of machine; however, program execution can vary among machine types. The allocation of computer memory depends on the hardware restrictions of the machine you are using.

MSC.Marc Host Systems

MSC.Marc runs on most computers. [Table 2-1](#) summarizes the types of machines and operating systems on which MSC.Marc currently runs.

Table 2-1 MSC.Marc Computer Versions

Computer	Machine Type	Operating System
HP Compaq Alpha	All machines	OSF1/Tru64
Hewlett-Packard	All machines	HP-UX
IBM	RS6000	AIX
Silicon Graphics	All machines	IRIX
Sun	All machines	Solaris
Intel Based	Pentium (etc.) Itanium2	Windows-NT/2000/XP, Linux

Workspace Requirements

Computing the amount of workspace required by MSC.Marc is a complex function of many variables. The most efficient method is to use the default values for the allocation. The program dynamically acquires memory if necessary and if available. In some situations, it is advantageous to initially allocate an amount of memory as described below. The following sections discuss workspace requirements for MSC.Marc.

MSC.Marc Workspace Requirements

The workspace used by MSC.Marc is allocated in separate parts. One part is referred to as *general memory* and contains a major portion of the data related to the model such as element connectivity, node and element sets, nodal vectors, stress and strain tensors, assembled stiffness, and mass matrices and decomposed operator matrix for certain matrix solvers. The initial amount of memory for this part can be entered by the user and the program dynamically allocates more memory if necessary. Other parts of the dynamically allocated workspace can not be influenced by the user. This includes data for contact bodies, kinematic boundary conditions, transformations, the so-called incremental backup, solver workspace for certain solvers among other things. These are all allocated separately. The *incremental backup* is an extra copy of stress tensors and similar quantities and is used for the Newton-Raphson iterations in a non-linear analysis. If the cut-back feature is activated, more data is stored in this part to allow for redoing the increment with a modified time step if a failure occurs.

General Memory stored in common/space/

Basic Data	Element Data (1)	Vector Data	Assembled Stiffness Matrix (2)
			Decomposed Stiffness Matrix (3)

Second Group of Memory stored in individual dynamically allocated vectors

Contact Data	Boundary Conditions	Transformation	Tying	Attach
Incremental Backup (4)		Decomposed Stiffness Matrix (5)		

1. Element Data memory allocation is reduced if [ELSTO](#) is used.
2. Assembled stiffness matrix memory allocation is reduced if out-of-core solver is used.

3. Decomposed Stiffness matrix is for solver type 0, 2, and 4.
4. Incremental Back memory allocation is reduced if **IBOOC** is used.
5. Decomposed Stiffness matrix is for solver type 6 and 8.

During the analysis, some of the blocks may grow because of changes in the model. In particular because of local or global adaptive meshing, the Element Data, Vector Data, Assembled Stiffness Matrix, and Decomposed Stiffness Matrix may grow in size because of the addition of elements and nodes. The Assembled Stiffness and Decomposed Stiffness matrices also expand due to changes in the bandwidth of the system. Changes in the bandwidth occur when deformable-deformable or self-contact occur. These changes can have a dramatic effect of the amount of memory required. The program dynamically requests additional memory. If this memory is not available, it activates one or more of the out-of-core options.

The amount of memory that can be used by an analysis is limited by the hardware employed in the operating system and by internal restrictions within MSC.Marc. On 32 bit systems, it is, in general, not possible to allocate more than 2 GB of memory for a process. If this limit is reached, the memory allocation request by MSC.Marc fails. Also, a failure to allocate can occur if other processes are using large amounts of memory. A typical output when a memory request fails is

```
memory request of 250000000 words failed
```

This occurs when MSC.Marc sends a memory allocation request (using the C function `malloc`) and the system refuses the request. There is also an internal restriction in Marc. It uses standard Fortran integers as pointers in vectors, and the largest amount of memory that can be addressed with this is 8 GB. This limit applies to each part of the separately allocated memory. For example, in a job where the general memory requires 4 GB, the solver needs 6 GB and the incremental backup needs 4 GB. This job can be run provided that the machine has at least 14 GB of available memory. For a parallel analysis, the 8 GB restriction, as well as the 2 GB limit for 32 bit system, is for each domain of the job since each domain corresponds to a separate process.

The memory allocation for the general memory can be affected by the user. The **SIZING** parameter specifies the amount of memory that is initially allocated for this part. This amount is allocated regardless of whether it is used or not. The default is set to a small value and the memory grows as needed. There is also an optional upper bound for this part of the memory allocation. The variable `MAXSIZE` is defined in the include file (`include.bat` for Windows) in the *tools* directory of the MSC.Marc installation. It specifies the maximum memory allowed for the general memory in terms of number of million 4 byte words (multiplied by 4 it gives Mbytes). If the general memory goes beyond this value, the job stops. This option can be used to avoid use of excessive paging space on a machine with limited memory.

For large problems, you may want to get an estimate of the workspace requirements for running a job without actually executing the analysis. To do this, insert the [STOP](#) parameter to exit the program normally after the workspace is allocated. Marc prints out a summary of the memory needed. For setting the appropriate sizing to avoid memory growth of the general memory (which can be inefficient) one should look at the entry on general memory of the summary printout. Please note that the [SIZING](#) parameter value in a parallel run refers to the current domain; while in a nonparallel run, it refers to the complete model. Chapter 12 [Output Results](#) describes the MSC.Marc output related to memory in more detail.

By default, the data is stored in-core. There are three out-of-core storage options in MSC.Marc.

- Out-of-core element data storage (the [ELSTO](#) parameter)
- Out-of-core storage of incremental backup (the [IBOOC](#) parameter)
- Out-of-core matrix solution

The out-of-core element data option stores element arrays (strains, stresses, temperatures, etc.) on a file (Fortran unit 3). Data connected with storage of all element quantities occupy a large amount of space for the more complex shell or three-dimensional elements. Putting this data out-of-core leads to a slow-down of the execution (disk access is, in general, slower than memory access) but the effect of this is usually not severe. The [ELSTO](#) parameter is used to set this option. The out-of-core option for incremental backup stores this data in a file (Fortran unit 29). If the element data is out-of-core, the incremental backup is automatically out-of-core as well. The slow-down related to incremental backup out-of-core is much less than the one related to element data out-of-core.

The out-of-core matrix solution has a more severe effect on the execution speed. Not all matrix solvers support this option. It is not supported by the iterative sparse solver (MSC.Marc solver number 2) and the hardware vendor provided solvers by Sun and HP (solver 6). Only the multifrontal direct solver (solver 8) supports out-of-core matrix solution with parallel processing.

In the case that a memory allocation request fails, MSC.Marc automatically switches to out-of-core storage. If the allocation fails during allocation for the general memory part, it first puts the element data out-of-core. If this is not sufficient, it puts the matrix solution part out-of-core. This can be done in separate stages, and MSC.Marc tries to keep as much of the data as possible in-core. If the memory allocation for incremental backup fails, this part is put out-of-core. The incremental backup part can also be put out-of-core to free up memory for the matrix solver.

File Units

MSC.Marc uses auxiliary files for data storage in various ways. Particular FORTRAN unit numbers are used for certain program functions (for example, [ELSTO](#), [RESTART](#)). [Table 2-2](#) lists these file unit numbers.

Note: On most systems, these files are references by file names, as well as by the file unit numbers.

Several of these files are necessary for solving most problems. The program input file and program output file are always required.

Table 2-2 FORTRAN File Units Used by MSC.Marc

File name	Unit	Description	File Type
jidname.log	0	Analysis sequence log file	sequential access, formatted
jidname.t01	1	Usually contains mesh data	random access, formatted
jidname.t02	2	OOO* solver scratch file	random access, binary
jidname.t03	3	Element data storage (see ELSTO parameter)	random access, binary
jidname.dat	5	Data input file	sequential access, formatted
jidname.out	6	Output file	sequential access, formatted
jidname.t08	8	Restart file, written out	sequential access, binary
ridname.t08	9	Restart file to be read in from a previous job	sequential access, binary
jidname.t11	11	OOO* solver scratch file	sequential access, binary
jidname.t12	12	OOO* solver scratch file	sequential access, binary
jidname.t13	13	OOO* solver scratch file	sequential access, binary
jidname.t14	14	OOO* solver scratch file	random access, binary
jidname.t15	15	OOO* solver scratch file	sequential access, binary
jidname.t16	16	Post file, written out	sequential access, binary
ridname.t16	17	Post file to be read in from a previous job	sequential access, binary
jidname.t18	18	Mesh optimization correspondence table	sequential access, formatted
jidname.fem	18	From MSC.Marc to external mesher	sequential access, formatted
jidname.t19	19	Post file, written out	sequential access, formatted
ridname.t19	20	Post file to be read in from a previous job	sequential access, formatted
jidname_j_.dat	21	Temporary input file when cut-back is used.	sequential access, formatted
jidname.t22	22	Subspace iteration scratch file	random access, binary
jidname.t23	23	Fluid-solid interaction file	sequential access, binary
pidname.t19	24	Heat data input file	sequential access, formatted
pidname.t16	25	Heat data input file	sequential access, binary

*OOO denotes Out-Of-Core solution.

Table 2-2 FORTRAN File Units Used by MSC.Marc (Continued)

File name	Unit	Description	File Type
jidname.t29	29	Incremental backup file when ELSTO , IBOOC is used, or insufficient memory exists.	sequential access, binary
sidname.t31	31	Substructure master data file	random access, binary
jidname.t32	32	Secant method file	sequential access, binary
jidname.t33	33	Lanczos scratch file	sequential access, binary
sidname.t35	35	Substructure file	sequential access, binary
sidname.t36	36	Substructure file	sequential access, binary
material.mat	38	Material data base file	sequential access, formatted
jidname.g	39	Intergraph post file	sequential access, formatted
jidname.unv	40	I-DEAS Universal post file	sequential access, formatted
jidname.t41	41	Post file – Domain Decomposition	sequential access, binary
ridname.t42	42	Post file – Domain Decomposition	sequential access, formatted
jidname.opt	45	Duplicate load case data file during design optimization run	sequential access, formatted
jidname.t46	46	Design optimization scratch file	sequential access, binary
jidname.trk	47	New particle tracking file	sequential access, formatted
ridname.trk	48	Old particle tracking file	sequential access, formatted
userspecified	49	User default file (see <i>MSC.Marc Volume C: Program Input</i> , Appendix C: Default File)	sequential access, formatted
jidname.vfs	50	Viewfactors	sequential access, formatted
jidname.lck	51	Locking of post file	sequential access, formatted
jidname.cnt	52	Dynamic control file	sequential access, formatted
jidname.mfd	52	rebar - Mentat interface	sequential access, formatted
jidname-bbc.mfd	52	beam-beam contact - Mentat interface	sequential access, formatted
jidname.seq	53	Sequence option	sequential access, formatted
jidname.rst	54	Load case data	sequential access, formatted
jidname.mesh	55	User supplied mesh	sequential access, formatted
jidname.feb	55	From 3-D mesher to MSC.Marc	sequential access, formatted
jidname.pass	56	Auto restart command line	sequential access, formatted
jidname.rms	57	2-D outline file for remeshing	sequential access, formatted
jidname.domesht jidname.donemesh	59	Lock files indicating meshing status “do mesh” and “done mesh”	sequential access, formatted
jidname.sltrk	60	New streamline tracking file	sequential access, formatted
ridname.sltrk	61	Old streamline tracking file	sequential access, formatted
jidname.sts	67	Analysis progress reporting file	sequential access, formatted

*OOC denotes Out-Of-Core solution.

Table 2-2 FORTRAN File Units Used by MSC.Marc (Continued)

File name	Unit	Description	File Type
bbctch.noconv	80	beam-beam contact information	sequential access, formatted
jidname.t81	81	multifrontal OOC scratch file	random access, binary
jidname.t82	82	multifrontal OOC scratch file	random access, binary
jidname.t83	83	multifrontal OOC scratch file	sequential access, binary
jidname.t84	84	multifrontal OOC scratch file	sequential access, binary
jidname.t85	85	multifrontal DDM scratch file	sequential access, binary
jidname.t86	86	multifrontal DDM scratch file	sequential access, binary
jidname.t87	87	multifrontal DDM scratch file	sequential access, binary
jidname.t88	88	multifrontal DDM scratch file	sequential access, binary
jidname.t89	89	multifrontal DDM scratch file	sequential access, binary
jidname.t90	90	multifrontal DDM scratch file	sequential access, binary
jidname.fld	91	Forming Limit input file	sequential access, binary
filename.apr	94	APT file - machining option	sequential access, file
filename.ccl	95	CL file - machining option	sequential access, file
EXITMSG	97	Exit messages	sequential access, formatted
USRDEF	98	User global default file (see <i>MSC.Marc Volume C: Program Input</i> , Appendix C: Default File)	sequential access, formatted
jidname.hmr	N/A	Hypermesh results file	sequential access, binary C file
*OOO denotes Out-Of-Core solution.			

Program Initiation

Procedures (shell script) are set up that facilitate the execution of MSC.Marc on most computers. These procedures invoke machine-dependent control or command statements. These statements control files associated with a job.

This shell script submits a job and automatically takes care of all file assignments. This command must be executed at the directory where all input and output files concerning this MSC.Marc job are available. To use this shell script, every MSC.Marc job should have a unique name qualifier and all MSC.Marc output files connected to that job uses this same qualifier. For restart, post, change state, all default MSC.Marc Fortran units should be used.

To actually submit a MSC.Marc job, the following command should be used:

```
run_marc -prog prog_name -jid job_name -rid rid_name -pid pid_name \
-sid sid_name -queue queue_name -user user_name -back back_value \
-ver verify_value -save save_value -vf view_name -def def_name \
-nproc number_of_processors -nthread number_of_threads \
-dir directory_where_job_is_processed -itree message_passing_type \
-host hostfile (for running over the network) -pq queue_priority \
-at date_time -comp compatible_machines_on_network -cpu time_limit \
-autorst autorestart_value
```

where the \ provides for continuation of the command line.

Table 2-3 Keyword Descriptions*

Keyword	Options	Description
-jid (-j)	job_name	Input file (and, therefore, job) name identification. Requires job_name.dat for all programs except the curve fit and neutral plot programs.
-prog (-pr)	progname	Run marc with or without user subroutine. Run the post file conversion program pldump. Run saved executable <i>progname.marc</i> from a previous job (see -user and -save).
-user (-u)	user_name	User subroutine user_name.f is used to generate a new executable program called user_name.marc (see -save and -prog).
-save (-sa)	no yes	Do <i>not</i> save the new executable program user_name.marc. Save the executable program user_name.marc for a next time (see -prog and -user).
-rid -(r)	restart_name	For marc or progname: identification of previous job_name that created restart file.

*Default options are shown in **bold**.

Table 2-3 Keyword Descriptions* (Continued)

Keyword	Options	Description
-pid (-pi)	post_name	For marc or progame: identification of previous job_name that created the post file containing temperature data.
-sid (-si)	substructure	Substructure jobs only: name of the substructuring file substructure.t31.
-queue (-q)	background foreground queue name	Run MSC.Marc in the background. Run MSC.Marc in the foreground. Submit to batch queue the queue name. Only available for machines with batch queue; for example, Convex, Cray. Queue names and submit command syntax can differ from site to site, adjust run_marc if necessary.
-back (-b)	yes no	Alternative for -queue: run MSC.Marc in the background. Run MSC.Marc in the foreground.
-ver (-v)	yes no	Ask for confirmation of these input options before starting the job. Start the job immediately.
-def (-de)	data_name	File name containing user defined default data.
-vf	viewfactor	Name of file containing viewfactors for radiation viewfactor.vfs.
-nprocd	number	Number of domains for parallel processing.
-nthread	number	Number of threads per task.
-itree		Message passing tree type for domain decomposition.(Normally for internal debugging purposes only.)
-dir	directory_name	Pathname to directory where the job I/O should take place. Defaults to current directory.
-sdir	directory_name	Directory where scratch files are placed.
-host (-ho)	hostfile	Specify the name of the host file for running over a network (default is execution on one machine only in which case this option is not needed).
-comp (-co)	yes no	When machines are compatible in a run over the network. Examples of compatible machines are: 1. Two or more SGI, SUN, IBM, HP, and DEC with exactly the same processor type and OS. 2. One SGI R8000/Irix 6.2 and one SGI R10000/Irix 6.5 machine. 3. One SUN Ultra/Solaris 2.5 and one SUN Ultra/Solaris 2.6. 4. One HP J Class/HPUX-10.20 and one HP C Class/HPUX-10.20. This option is only needed when user subroutines are used.
-ci	yes no	Copy input files automatically to remote hosts for a network run, if necessary.
-cr	yes no	Copy post files automatically from remote hosts used for a network run, if necessary.
-pq	0,1,2,etc	Batch queue only: queue priority.

*Default options are shown in **bold**.

Table 2-3 Keyword Descriptions* (Continued)

Keyword	Options	Description
-at (-a)	date/time	Batch queue only: delay time for start of job. Syntax: January,1,1994,12:30 or: today,5pm
-cpu	sec	Batch queue only: CPU time limit.
-autorst	0 or 1	If 0 when remeshing is required, the analysis program goes into a wait state until meshing is complete. If 1 when remeshing is required, the analysis program stops, the mesher begins, and the analysis program automatically restarts. Using the default procedure (0) uses more memory, but less I/O. Using the restart procedure (1), invokes the RESTART LAST option.
*Default options are shown in bold .		

Examples of Running MSC.Marc Jobs

Example 1:

```
run_marc -jid e2x1
```

This runs the job e2x1 in the background using a single processor.

The input file is e2x1.dat in the current working directory.

Example 2:

```
run_marc -jid e2x14 -user u2x14 -sav y -nproc 4
```

This runs the job e2x14 in the background with four processors. The user subroutine is linked with the MSC.Marc library and a new execu module is created as u2x14.marc and saved in the current working directory after completion of the job.

Example 3:

```
run_marc -jid e2x14a -prog u2x14 -nproc 4
```

Use the above saved module u2x14.marc to run the job e2x14a in the background with four processors.

Example 4:

```
run_marc -jid e3x2a -v no -b no -nproc 2
```

Run the job e3x2a in the foreground with two processors. The job runs immediately without verifying any arguments interactively. If there are any input errors in the arguments, the job does not run and the error message is sent to the screen.

Example 5:

```
run_marc -jid e3x2b -rid e3x2a
```

Run the job e3x2b in the background using a single processor. The job uses e3x2a.t08, which is created from Example 4, as restart file.

Example 6:

```
run_marc -jid e2x1 -nproc 2
```

Runs a two processor job on a single parallel machine.

Example 7:

```
run_marc -jid e2x1 -nproc 2 -host hostfile
```

Runs a two-processor job over a network. The hosts are specified in the file hostfile.

CHAPTER

3

Data Entry

- Input Conventions
- Parameters
- Model Definition Options
- History Definition Options
- REZONE Option

The input data structure is made up of three logically distinct sections:

1. Parameters describe the problem type and size.
2. Model definition options give a detailed problem description.
3. History definition options define the load history.

Input data is organized in (optional) blocks. Key words identify the data for each optional block. This form of input enables you to specify only the data for the optional blocks that you need to define your problem. The various blocks of input are “optional” in the sense that many have built-in default values which can be used by MSC.Marc in the absence of any explicit input from you.

Input Conventions

MSC.Marc performs all data conversion internally so that the system does not abort because of data errors made by you. The program reads all input data options alphanumerically and converts them to integer, floating point, or keywords, as necessary. MSC.Marc issues error messages and displays the illegal option image if it cannot interpret the option data field according to the specifications given in the manual. When such errors occur, the program attempts to scan the remainder of the data file and ends the run with an exit error message at the **END OPTION** option or at the end of the input file.

Two input format conventions can be used: fixed and free format. You can mix fixed and free format options within a file, but you can only enter one type of format on a single option.

The syntax rules for fixed fields are as follows:

- You must right-justify integers in their fields. (The right blanks are filled with zeroes).
- Give floating point numbers with or without an exponent. If you give an exponent, it must be preceded by the character E or D and must be right-justified.

The syntax rules for free fields are as follows:

- Check that each option contains the same number of data items that it would contain under standard fixed-format control. This syntax rule allows you to mix fixed-field and free-field options in the data file because the number of options you need to input any data list are the same in both cases.
- Separate data items on a option with a comma. The comma can be surrounded by any number of blanks. Within the data item itself, no embedded blanks can appear.
- Give floating point numbers with or without an exponent. If you use an exponent, it must be preceded by the character E or D and must immediately follow the mantissa (no embedded blanks).
- Give keywords exactly as they are written in the manual. Embedded blanks do not count as separators here (for example, BEAM SECT is one word only).
- If a option contains only one free-field data item, follow that item with a comma. For example, the number “1” must be entered as “1,” if it is the only data item on a option. If the comma is omitted, the entry is treated as fixed format and may not be properly right-justified.

- If the **EXTENDED** parameter is used, integer data is given using 10 fields as opposed to 5 fields. This allows very large models to be included in MSC.Marc. Additionally, real numbers are entered using 20 or 30 fields as opposed to 10 or 15. This allows increased accuracy when reading in data.
- All data can be entered as uppercase or lowercase text.

Input of List of Items

MSC.Marc often requests that you enter a list of items in association with certain program functions. As an example, these items can be a set of elements as in the **ISOTROPIC** option, or a set of nodes as in the **POINT LOAD** option. Twelve types of items can be requested:

Element numbers	Points ids
Node numbers	Curves ids
Degree of freedom numbers	Surfaces ids
Integration point numbers	Body numbers
Layer numbers	Edges pairs
Increment numbers	Faces pairs

This list can be entered using either the OLD format (compatible with the G, H, and J versions of MSC.Marc) or the NEW format (the K version).

Using the OLD format, you can specify the list of items in three different forms. You can specify:

1. A range of items as:

$m \ n \ p$

which implies items m through n by p . If p is not specified, the program assumes it is 1. Note that the range can either increase or decrease.

2. A list of items as:

$-n \ a_1 \ a_2 \ a_3 \ \dots \ a_n$

which implies that you should give n items, and they are $a_1, \dots, a_n$.

3. A set name as:

MYSET

which implies that all items previously specified in the set MYSET are used. Specify the items in a set using the **DEFINE** model definition option.

Using the NEW format, you can express the list of items as a combination of one or more sublists. These sublists can be specified in three different forms. The following operations can be performed between sublists:

AND
INTERSECT
EXCEPT

When you form a list, subsets are combined in binary operations (from left to right). The following lists are examples.

1. `SUBLIST1` and `SUBLIST2`

This list implies all items in subsets `SUBLIST1` and `SUBLIST2`. Duplicate items are eliminated and the resulting list is sorted.

2. `SUBLIST1 INTERSECT SUBLIST2`

This list implies only those items occurring both in subsets `SUBLIST1` and `SUBLIST2`; the resulting list is sorted.

3. `SUBLIST1 EXCEPT SUBLIST2`

This list implies all items in subset `SUBLIST1` except those which occur in subset `SUBLIST2`; the resulting list is sorted.

4. `SUBLIST1 AND SUBLIST2 EXCEPT SUBLIST3 INTERSECT SUBLIST4`

This list implies the items in subsets `SUBLIST1` and `SUBLIST2` minus those items that occur in subset `SUBLIST3`. Then, if the remaining items also occur in subset `SUBLIST4`, they are included in the list.

Sublists can have several forms. You can specify:

1. A range of items as:

`m TO n BY p`
or
`m THROUGH n BY p`

which implies items `m` through `n` by `p`. If “`BY p`” is not included, the program assumes “`BY 1`”. Note that the range can either increase or decrease.

2. A string of items as:

`a1 a2 a3 ... an`

which implies that `n` items are to be included. If continuation options are necessary, then either a `C` or **CONTINUE** should be the last item on the option.

3. A set name as:

`MYSET`

which implies that all items you previously specified to be in the set `MYSET` are used. You specify the items in a set using the **DEFINE** option.

Note: `INTERSECT` or `EXCEPT` cannot be used when defining lists of degrees of freedom.

In a list, edges and faces are entered as pairs (*i*:*j*) where *i* is the user element id and *j* is the edge id or face id. The edge id/face id for the different element classes is given in *MSC.Marc Volume C: Program Input*, [Chapter 1](#).

There are two types of edge and face sets; those expressed in MSC.Marc convention or the MSC.Marc Mentat convention. The edge/face id in MSC.Marc convention is one greater than the MSC.Marc Mentat convention.

For example, to specify edge 1 on elements 1 to 20, one would use:

1:1 TO 20:1

Examples of Lists

This section presents some examples of lists and entry formats.

Use the [DEFINE](#) model definition option to associate a list of items with a set name with items. Three sets are defined below: FLOOR, NWALL, and WWALL.

- DEFINE NODE SET FLOOR contains:
1 TO 15 (1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 11, 12, 13, 14, 15)
- DEFINE NODE SET NWALL contains:
5 TO 15 BY 5 AND 20 TO 22 (5, 10, 15, 20, 21, 22)
- DEFINE NODE SET WWALL contains:
11 TO 20 (11, 12, 13, 14, 15, 16, 17, 18, 19, 20)

Some possible lists are:

- NWALL AND WWALL, which would contain nodes:
5 10 11 12 13 14 15 16 17 18 19 20 21 22
- NWALL INTERSECT WWALL, which would contain nodes:
15 20
- NWALL AND WWALL EXCEPT FLOOR, which would contain nodes:
16 17 19 20 21 22

Parameters

This group of parameters allocates the necessary working space for the problem and sets up initial switches to control the flow of the program through the desired analysis. This set of input must be terminated with an **END** parameter. The input format for these parameters is described in *MSC.Marc Volume C: Program Input*.

Model Definition Options

This set of data options enters the initial loading, geometry, and material data of the model and provides nodal point data, such as boundary conditions. Model definition options are also used to govern the error control and restart capability. Model definition options can also specify print-out and postprocessing options. The data you enter on model definition options provides the program with the necessary information for determining an initial elastic solution (zero increment solution). This group of options must be terminated with the [END OPTION](#) option. The input format for these options is described in *MSC.Marc Volume C: Program Input*.

History Definition Options

This group of options provides the load incrementation and controls the program after the initial elastic analysis. History definition options also include blocks which allow changes in the initial model specifications. Each set of load sets must be terminated with a [CONTINUE](#) option. This option requests that the program perform another increment or series of increments if you request the auto-incrementation features. The input format for these options is described in [MSC.Marc Volume C: Program Input](#).

A typical input file setup for the MSC.Marc program is shown below.

- MSC.Marc Parameter
Terminated by an [END](#) parameter
- MSC.Marc Model Definition Options
(Zero Increment)
Terminated by an [END OPTION](#) option
- MSC.Marc History Definition
Data for the First Increment
Terminated by a [CONTINUE](#) option
- (Additional History Definition
Option for the second, third, ..., Increments)

[Figure 3-1](#) is a dimensional representation of the MSC.Marc input data file.

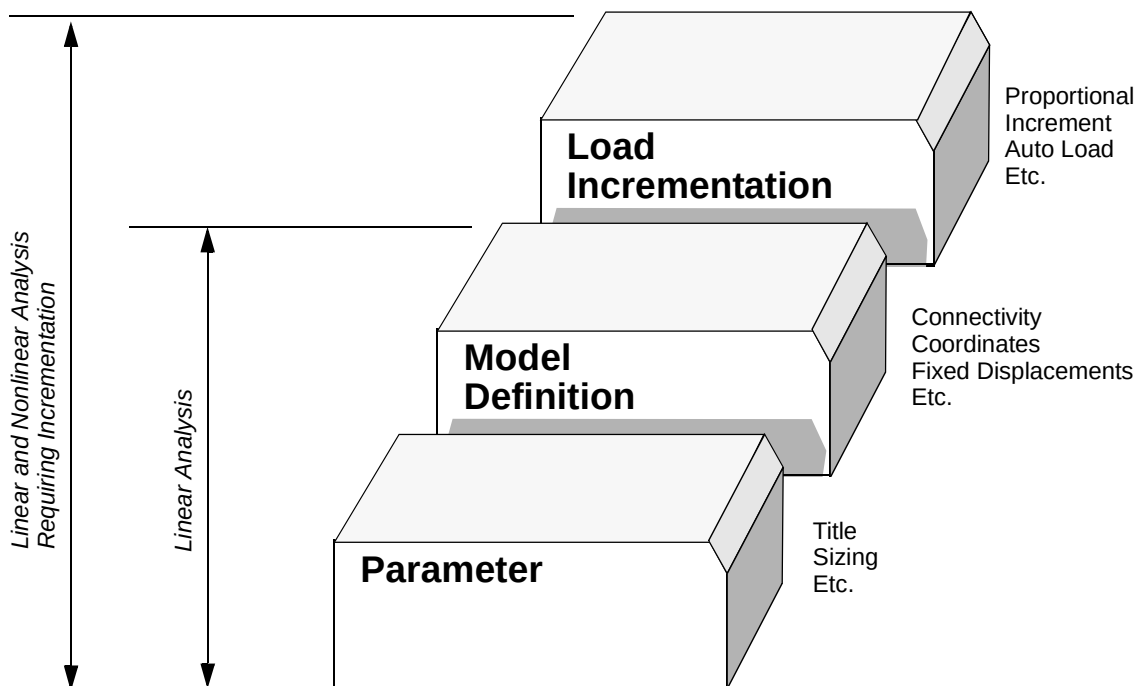


Figure 3-1 The MSC.Marc Input Data File

REZONE Option

When the [REZONE](#) option is inserted into the input file and the manual procedure is used, the program reads additional data options to control the rezoning steps. These options must immediately follow the [END OPTION](#) option or a [CONTINUE](#) history definition option.

You can select as many rezoning steps in one increment as you need. Every rezoning step is defined by the data, starting with the [REZONE](#) option and ending with the [CONTINUE](#) option. The [END REZONE](#) option terminates the complete set of rezoning steps that form a complete rezoning increment. Follow the rezoning input with normal history definition data, or again by rezoning data. The input format for these options is described in [MSC.Marc Volume C: Program Input](#).

CHAPTER

4

Introduction to Mesh Definition

- Direct Input
- User Subroutine Input
- MESH2D
- MSC.Marc Mentat
- FXORD Option
- Incremental Mesh Generators
- Bandwidth Optimization
- Rezoning
- Substructure
- BEAM SECT Parameter
- Error Analysis
- Local Adaptivity
- Global Remeshing

This chapter describes the techniques for mesh definition available internally in MSC.Marc. Mesh definition is the process of converting a physical problem into discrete geometric entities for the purpose of analysis. Before a body can undergo finite element analysis, it must be modeled into discrete physical elements. An example of mesh definition is shown in [Figure 4-1](#).

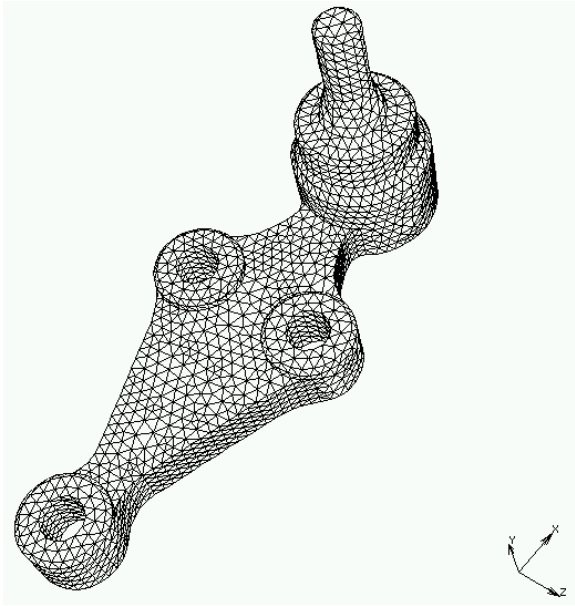


Figure 4-1 Structure with Finite Element Mesh

Mesh definition encompasses the placement of geometric coordinates and the grouping of nodes into elements. For MSC.Marc to have a valid mesh definition, the nodes must have geometric coordinates and must be connected to an element.

First, describe the element by entering the element number, the element type, and the node numbers that make up the element.

Next, enter the physical coordinates of the nodal points.

Note: You do not need to enter element numbers and node numbers sequentially or consecutively

Direct Input

You must enter two types of data into MSC.Marc for direct mesh definition: connectivity data, which describes the nodal points for each element, and coordinate data which gives the spatial coordinates of each nodal point. This section describes how to enter this data.

Element Connectivity Data

You can enter connectivity data from either the input option file (FORTRAN unit 5) or from an auxiliary file. Several blocks of connectivity can be input. For example, the program can read one block from tape and subsequently read a block from the input option file. Each block must begin with the word `CONNECTIVITY`. In the case of duplicate specification, MSC.Marc always uses the data that was input last for a particular element.

Enter the nodal points of two-dimensional elements in a counterclockwise order. [Figure 4-2](#) illustrates correct and incorrect numbering of element connectivity data.

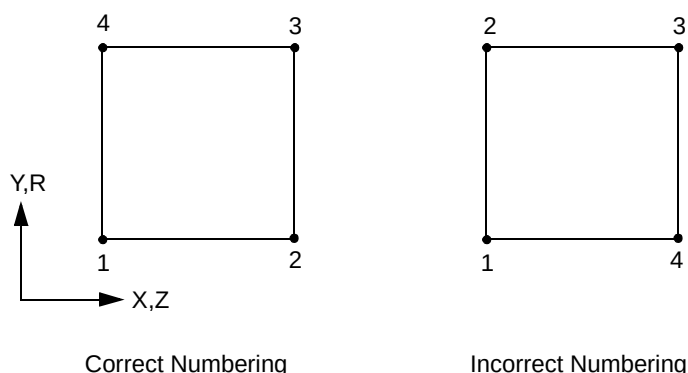


Figure 4-2 Correct/Incorrect Numbering of Two-Dimensional Element Connectivity of 4-Node Elements

When there are eight nodal points on a two-dimensional element, number the corner nodes 1 through 4 in counterclockwise order. The midside nodes 5 through 8 are subsequently numbered in counterclockwise order. [Figure 4-3](#) illustrates the correct numbering of element connectivity of 8-node elements.

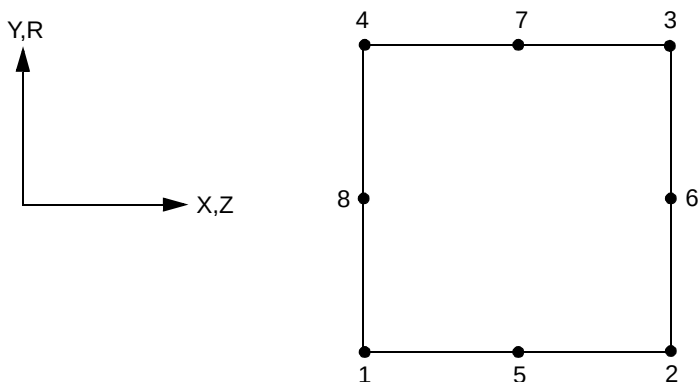


Figure 4-3 Numbering of Two-Dimensional Element Connectivity for 8-Node Quadrilateral Elements

Lower-order triangular elements are numbered using the counterclockwise rule.

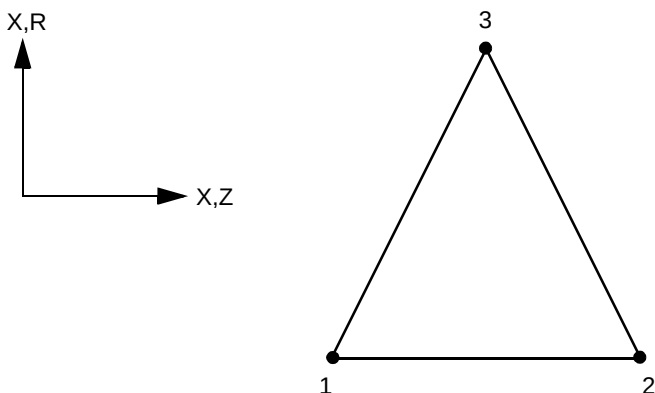


Figure 4-4 Numbering of 3-Node Triangular Element

Note that quadrilateral elements can be collapsed into triangular elements by repeating the last node.

The higher order triangular elements have six nodes, the corner nodes are numbered first in a counterclockwise direction. The midside nodes 4 through 6 are subsequently numbered as shown in [Figure 4-5](#).

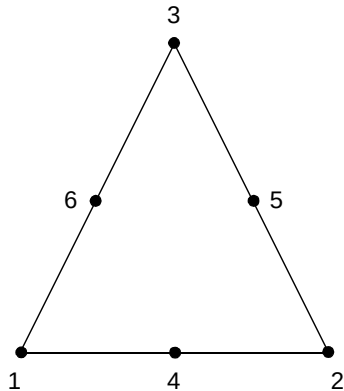


Figure 4-5 Numbering of 6-Node Triangular Element

Number three-dimensional elements in the same order as two-dimensional elements for each plane. Enter nodes for an 8-node brick in counterclockwise order as viewed from inside the element. First, enter nodes comprising the base; then enter ceiling nodes as shown in [Figure 4-6](#).

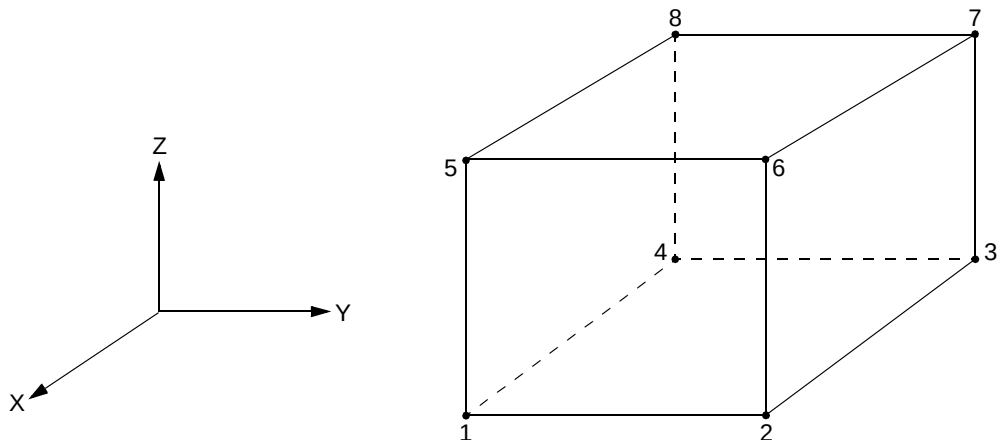


Figure 4-6 Numbering of Element Connectivity for 8-Node Brick

A 20-node brick contains two 8-node planes and four nodes at the midpoints between the two planes. Nodes 1 through 4 are the corner nodes of one face, given in counterclockwise order as viewed from within the element. Nodes 5 through 8 are on the opposing face; nodes 9 through 12 are midside nodes on the first face, while nodes

13 through 16 are their opposing midside nodes. Finally, nodes 17 through 20 lie between the faces with node 17 between 1 and 5. [Figure 4-7](#) illustrates the numbering of element connectivity for a 20-node brick.

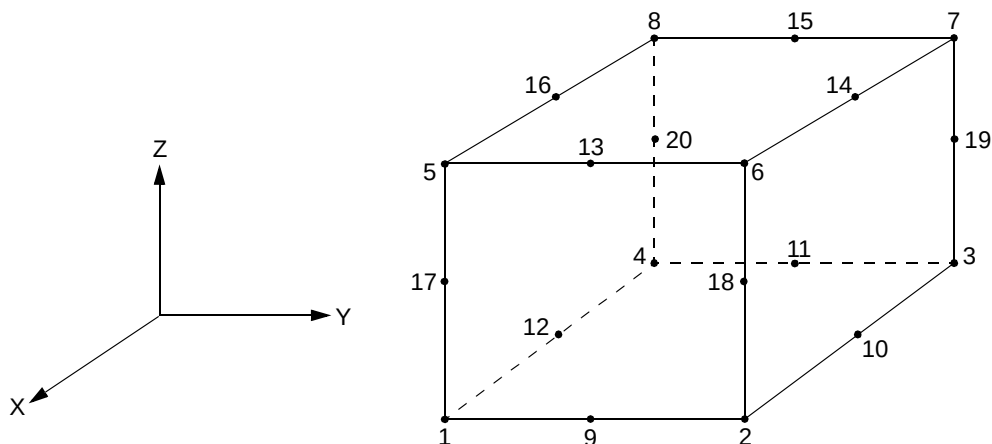


Figure 4-7 Numbering of Three-Dimensional Element Connectivity for 20-Node Brick

The four node tetrahedral is shown in [Figure 4-8](#).

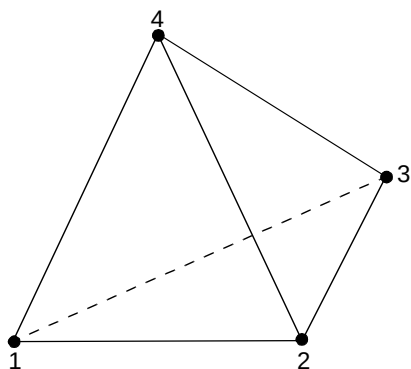


Figure 4-8 Numbering of Four-Node Tetrahedral

The ten-node tetrahedral is shown in [Figure 4-9](#). The corner nodes 1-4 are numbered first. The first three midside nodes occur on the first face. Nodes 8, 9, and 10 are between nodes 1 and 4, 2 and 4, and 3 and 4, respectively.

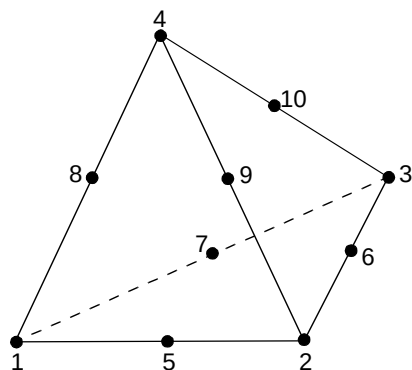


Figure 4-9 Numbering of Ten-Node Tetrahedral

Nodal Coordinate Data

You can enter nodal coordinates directly from the input option file (FORTRAN unit 5) or from an auxiliary file. You can enter several blocks of nodal coordinate data in a file. In the case of duplicate specifications, the program uses data entered last for a particular nodal point in the mesh definition.

Direct nodal input can be used to input local corrections to a previously generated set of coordinates. These options give the modified nodal coordinates.

The **CYLINDRICAL** option can be used to transform coordinates given in a cylindrical system to a Cartesian system.

Note: requires the final coordinate data in terms of a single Cartesian system. Refer to [MSC.Marc Volume B: Element Library](#) to determine the required coordinate data for a particular element type.

Activate/Deactivate

You have the ability to turn on and off elements using this option, which is useful when modeling ablation or excavation. When you enter the mesh connectivity, the program assumes that all elements are to be included in the analysis unless they are deactivated. This effectively removes this material from the model. These elements can be reinstated later by using the **ACTIVATE** option. If the element is activated, one can select if the level of stress is to be reinstated or set to zero. The use of these options results in nonlinear behavior and have an effect upon convergence.

User Subroutine Input

User subroutines can be used to generate or modify the data for mesh definition. User subroutine [UFCNN](#) generates or modifies element connectivity data. The [UFCNN](#) model definition option activates this subroutine. The user subroutine is called once for each element requested. Refer to *Volume D: User Subroutines and Special Routines* for a description of user subroutine [UFCNN](#) and instructions for its use.

User subroutine [UFORD](#) generates or modifies the nodal coordinates. The [UFORD](#) model definition option activates this subroutine. The user subroutine is called once for each node requested. Refer to *Volume D: User Subroutines and Special Routines* for a description of user subroutine [UFORD](#) and instructions for its use.

MESH2D

MESH2D generates a mesh of quadrilateral or triangular elements for a two-dimensional body of any shape. The generated mesh is written to a separate file and must be read with the **CONNECTIVITY**, **COORDINATES**, and **FIXED DISP**, etc., options.

Block Definition

In MESH2D, a physical object or domain is divided into quadrilateral and/or triangular parts, called “blocks”. Quadrilateral blocks are created by MSC.Marc by mapping with polynomials of the third order from a unit square. These blocks can, therefore, be used to approximate curved boundaries. The geometry of a quadrilateral block is defined by the coordinates on 12 nodes shown in **Figure 4-10**. If the interior nodes on an edge of the block are equal to zero or are not specified, the edge of the block is straight.

Triangular blocks have straight edges. The geometry of a triangular block is defined by the coordinates of the three vertices.

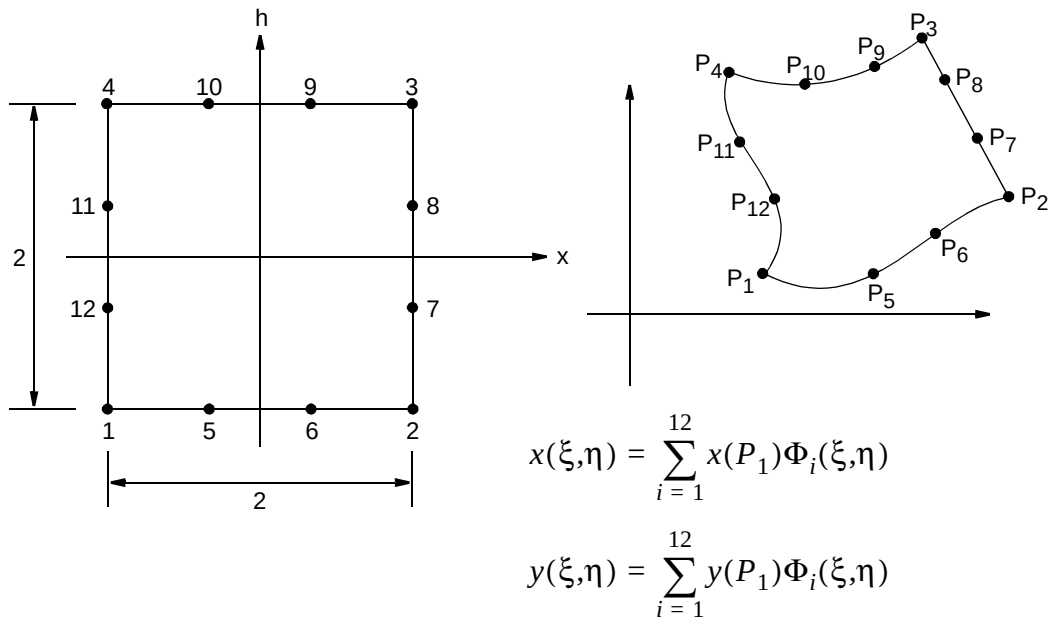


Figure 4-10 Typical Quadrilateral Block

Merging of Nodes

MSC.Marc creates each block with a unique numbering scheme. The **MERGE** option fuses all nodes that lie within a small circle, renumbers the nodes in sequence, and then removes all gaps in the numbering system. You can select which blocks are to be merged together, or you can request that all blocks be merged. You must give the closeness distance for which nodes will be merged.

Block Types

MESH2D generates two types of quadrilateral blocks. Block Type 1 is a quadrilateral block that is covered by a regular grid. The program obtains this grid by dividing the block edge into M by N intervals. [Figure 4-11](#) illustrates the division of block edges into intervals with M = 4, N = 3.

The $P_1 P_4$ face of the block is the 1-4 face of triangular elements and the $P_1 P_2$ face of the block becomes the 1-2 face of quadrilateral elements.

Block Type 2 is a quadrilateral block that allows the transition of a coarse mesh to a finer one. In one direction, the block is divided into M, 2M, 4M ...; while in the other direction, the block is divided into N intervals. [Figure 4-12](#) illustrates the division of Block Type 2 edges into intervals.

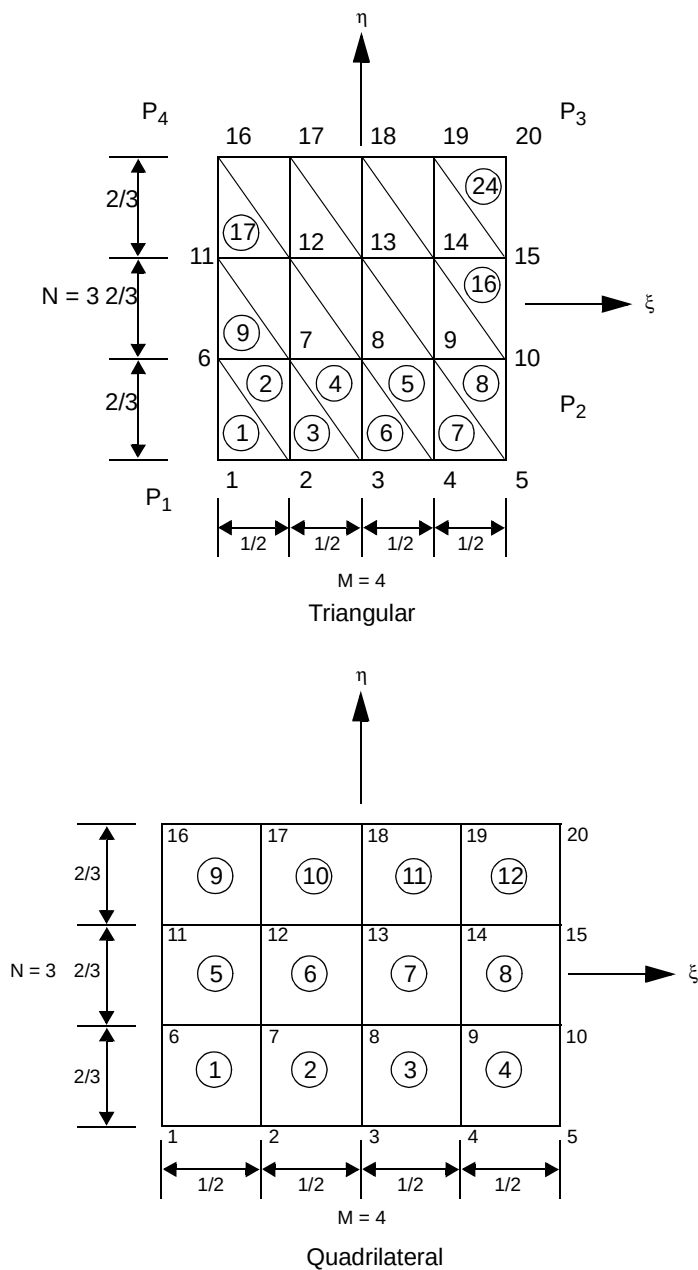


Figure 4-11 Block Type 1

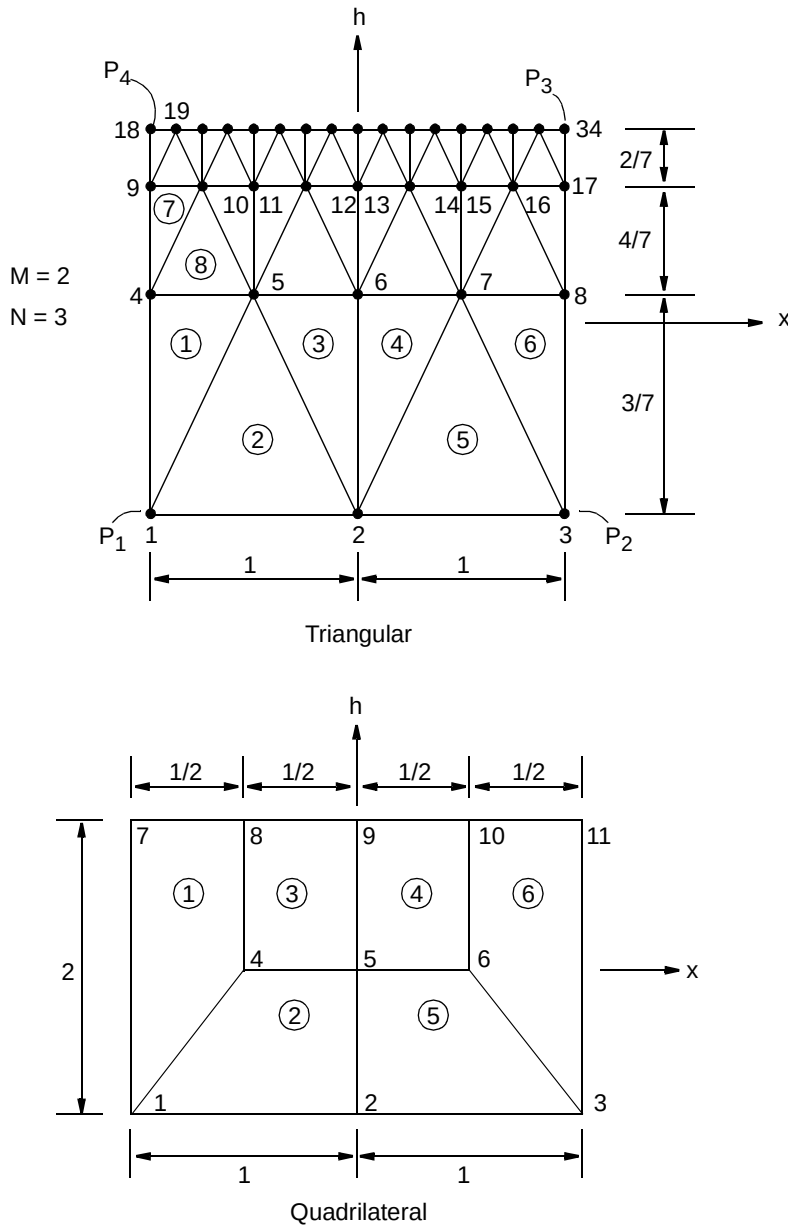


Figure 4-12 Block Type 2

The $P_1 P_2$ face of the block becomes the 1-2 face of quadrilateral elements, and the $P_2 P_3$ face of the block is the 2-3 face of triangular elements.

Block Type 3 is a triangular block. The program obtains the mesh for this block by dividing each side into N equal intervals. [Figure 4-13](#) illustrates Block Type 3 for triangular and quadrilateral elements.

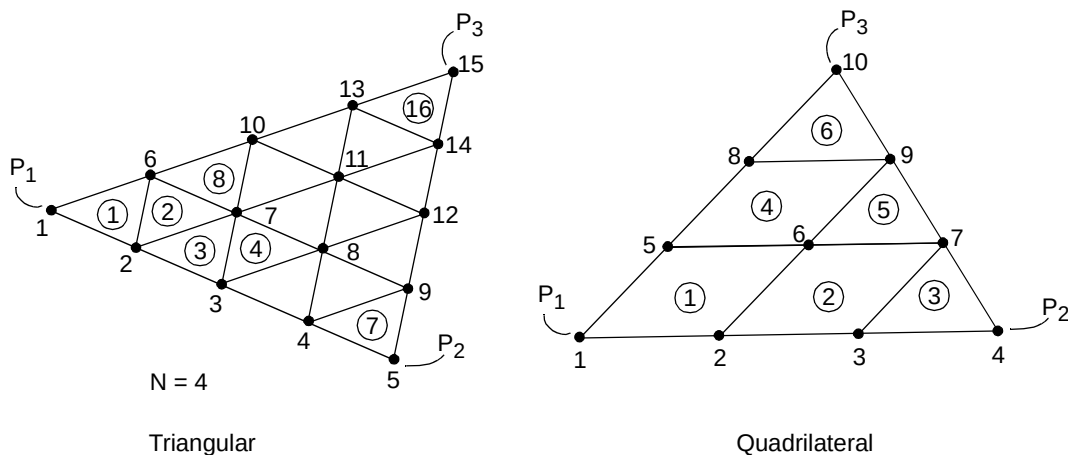


Figure 4-13 Block Type 3

Block Type 4 is a refine operation about a single node of a block. The values of N and M are not used.

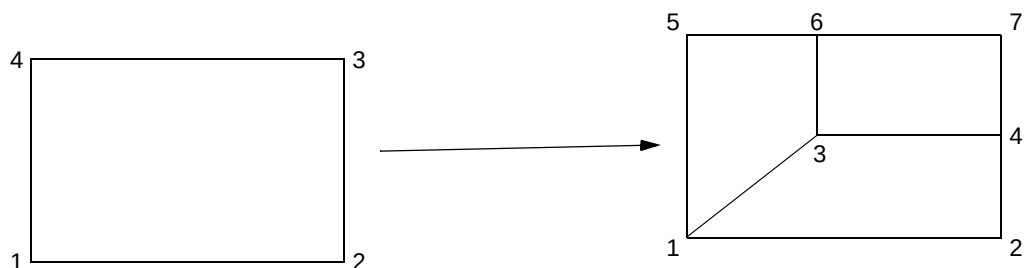


Figure 4-14 Block Type 4

If quadrilateral elements are used in a triangular block, the element near the $P_2 P_3$ face of the block is collapsed by MESH2D in every row. The $P_1 P_2$ face of the block is the 1-2 face of the generated elements.

Symmetry, Weighting, and Constraints

MESH2D contains several features that facilitate the generation of a mesh: use of symmetries, generation of weighted meshes, and constraints. These features are discussed below.

MESH2D can use symmetries in physical bodies during block generation. An axis of symmetry is defined by the coordinates of one nodal point, and the component of a vector on the axis. One block can be reflected across many axes to form the domain. [Figure 4-15](#) illustrates the symmetry features of MESH2D.

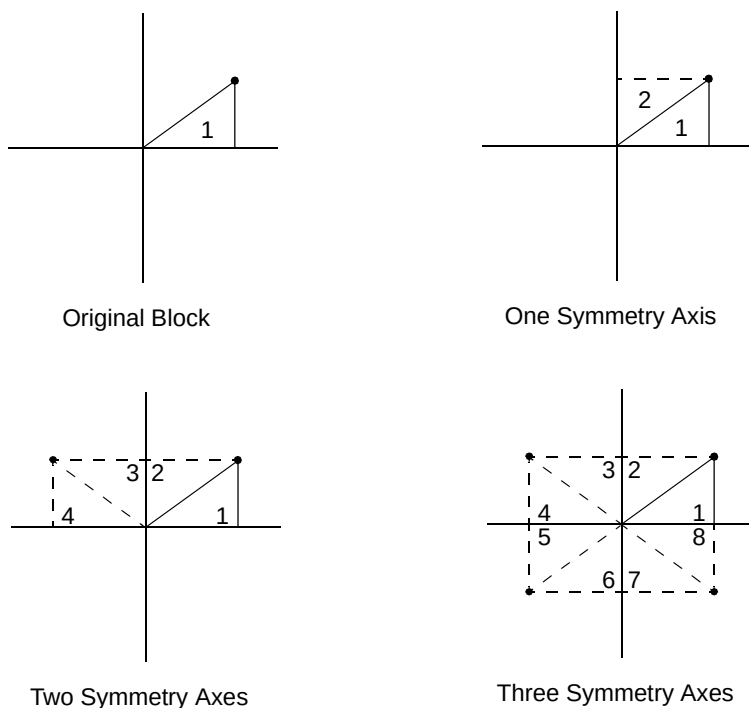


Figure 4-15 Symmetry Option Example

A weighted mesh is generated by the program by spacing the two intermediate points along the length of a boundary. This technique biases the mesh in a way that is similar to the weighting of the boundary points. This is performed according to the third order isoparametric mapping function.

Note: If a weighted mesh is to be generated, be cautious not to move the interior boundary points excessively. If the points are moved more than 1/6 of the block length from the 1/3 positions, the generated elements can turn inside-out.

The **CONSTRAINT** option generates boundary condition restraints for a particular degree of freedom for all nodes on one side of a block. The option then writes the constraints into the file after it writes the coordinate data. The **FIXED DISP**, etc., option must be used to read the boundary conditions generated from the file.

Additional Options

Occasionally, you might want to position nodes at specific locations. The coordinates of these nodes are entered explicitly and substituted for the coordinates calculated by the program. This is performed using the **SPECIFIED NODES** option. Some additional options in MESH2D are:

- **MESH2D** can be used several times within one input file.
- The **START NUMBER** option gives starting node and element numbers.
- The **CONNECT** option allows forced connections and/or disconnections with other blocks. This option is useful when the final mesh has cracks, tying, or gaps between two parts.
- The **MANY TYPES** option specifies different element types.

MSC.Marc Mentat

MSC.Marc Mentat is an interactive program which facilitates mesh definition by generating element connectivity and nodal coordinates. Some of the MSC.Marc Mentat capabilities relevant to mesh generation are listed below.

- Prompts you for connectivity information and nodal coordinates. Accepts input from a keyboard or mouse.
- Accepts coordinates in several coordinate systems (Cartesian, cylindrical, or spherical).
- Translates and rotates (partial) meshes.
- Combines several pre-formulated meshes.
- Duplicates a mesh to a different physical location.
- Generates a mirror image of a mesh.
- Subdivides a mesh into a finer mesh.
- Automatic mesh generation in two- and three-dimensions.
- Imports geometric and finite element data from CAD systems.
- Smooths nodal point coordinates to form a regular mesh
- Converts geometric surfaces to meshes.
- Refines a mesh about a point or line.
- Expands line mesh into a surface mesh, or a surface mesh into a solid mesh.
- Calculates the intersection of meshes.
- Maps nodal point coordinates onto prescribed surfaces.
- Writes input data file for connectivity and coordinates in MSC.Marc format for use in future analyses.
- Apply boundary conditions to nodes and elements.
- Define material properties.
- Submit MSC.Marc jobs.

FXORD Option

The **FXORD** model definition option (*Volume C: Program Input*) generates doubly curved shell elements of element type 4, 8, or 24 for the geometries most frequently found in shell analysis. Since the mathematical form of the surface is well-defined, the program can generate the 11 or 14 nodal coordinates needed by element type 8, 24, or 4 to fit a doubly curved surface from a reduced set of coordinates. For example, you can generate an axisymmetric shell by entering only four coordinates per node. The **FXORD** option automatically generates the complete set of coordinates required by the elements in the program from the mathematical form of the surface.

A rotation and translation option is available for all components of the surface to give complete generality to the surface generation. The input to **FXORD** consists of the reduced set of coordinates given in a local coordinate system and a set of coordinates which orient the local system with respect to the global system used in the analysis. The program uses these two sets of coordinates to generate a structure made up of several shell components for analysis.

The **FXORD** option allows for the generation of several types of geometries. Because you may need to analyze shells with well-defined surfaces not available in this option, you can use the **UFORD** user subroutine to perform your own coordinate generation (*Volume D: User Subroutines and Special Routines*). The **FXORD** option can also be used to convert cylindrical coordinates or spherical coordinates to Cartesian coordinates for continuum elements.

Major Classes of the FXORD Option

The following cases are considered:

- **Shallow Shell (Type 1)**
- **Axisymmetric Shell (Type 2)**
- **Cylindrical Shell Panel (Type 3)**
- **Circular Cylinder (Type 4)**
- **Plate (Type 5)**
- **Curved Circular Cylinder (Type 6)**
- **Convert Cylindrical to Cartesian (Type 7)**
- **Convert Spherical to Cartesian (Type 8)**

Shallow Shell (Type 1)

Type 1 is a shallow shell with

$$\theta_1 = x_1, \quad \theta_2 = x_2 \quad (4-1)$$

The middle surface of Figure 4-16 (Type 1) is defined by an equation of the form

$$x_3 = x_3(x_1, x_2) \quad (4-2)$$

and the surface is determined when the following information is given at each node.

$$x_1, x_2, x_3, \frac{\partial x_3}{\partial x_1}, \frac{\partial x_3}{\partial x_2}, \frac{\partial^2 x_3}{\partial x_1 \partial x_2} \quad (4-3)$$

The last coordinate is only necessary for Element Type 4.

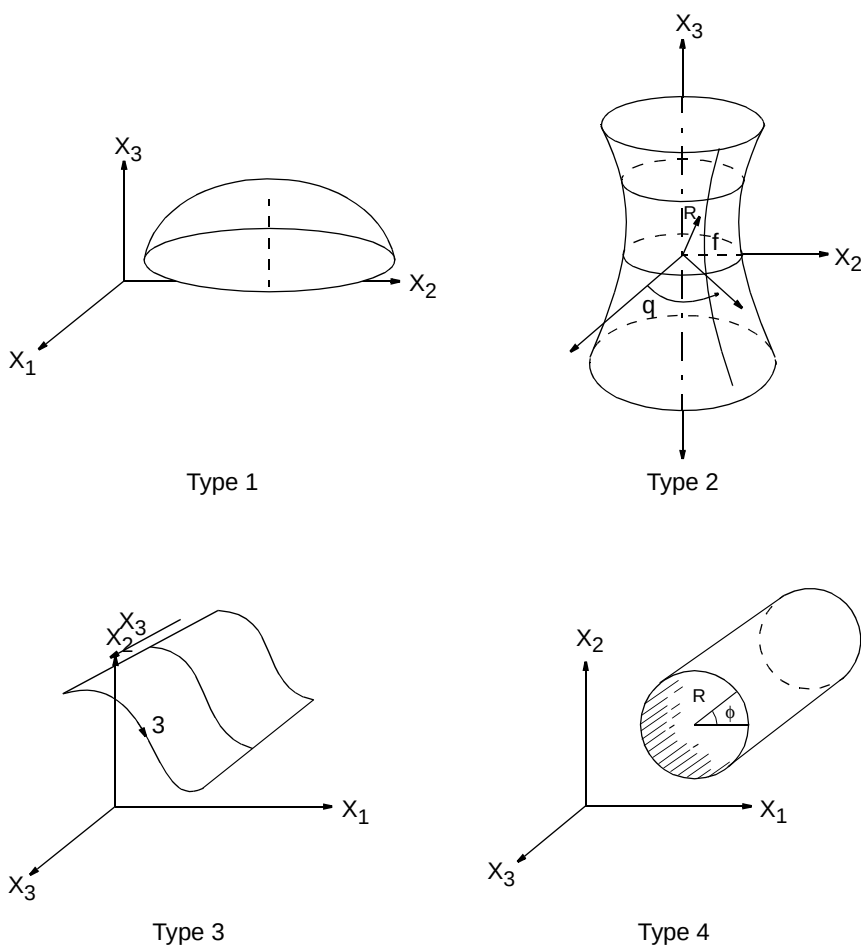


Figure 4-16 Classification of Shells

Axisymmetric Shell (Type 2)

The middle surface symmetric to the x_3 axis ([Figure 4-16](#), Type 2) is defined as:

$$\begin{aligned}x_1 &= R(\phi) \cos \phi \cos \theta \\x_2 &= R(\phi) \cos \phi \sin \theta \\x_3 &= R(\phi) \sin \phi\end{aligned}\tag{4-4}$$

where ϕ and θ are the angles shown in [Figure 4-16](#). In this case, the surface is defined by

$$\theta, \phi, R, \frac{dR}{d\phi}\tag{4-5}$$

The angles θ and ϕ are given in degrees.

Cylindrical Shell Panel (Type 3)

The middle surface is the cylinder defined by [Figure 4-16](#).

$$\begin{aligned}x_1 &= x_1(s) \\x_2 &= x_2(s) \\x_3 &= x_3\end{aligned}\tag{4-6}$$

The nodal geometric data required is

$$s, x_3, x_1, x_2, \frac{dx_1}{ds}, \frac{dx_2}{ds}\tag{4-7}$$

Circular Cylinder (Type 4)

This is the particular case of Type 3 where the curve

$$x_1(s), x_2(s)\tag{4-8}$$

is the circle given by [Figure 4-16](#) (Type 4).

$$\begin{aligned}x_1 &= R \cos \theta \\x_2 &= R \sin \theta\end{aligned}\tag{4-9}$$

The only nodal information is now

$$\theta, x_3, R \quad (4-10)$$

Note that θ is given in degrees and, because R is constant, it needs to be given for the first nodal point only.

Plate (Type 5)

The shell is degenerated into the plate

$$x_3 = 0 \quad (4-11)$$

The data is reduced to

$$x_1, x_2 \quad (4-12)$$

Curved Circular Cylinder (Type 6)

Figure 4-17 illustrates this type of geometry.

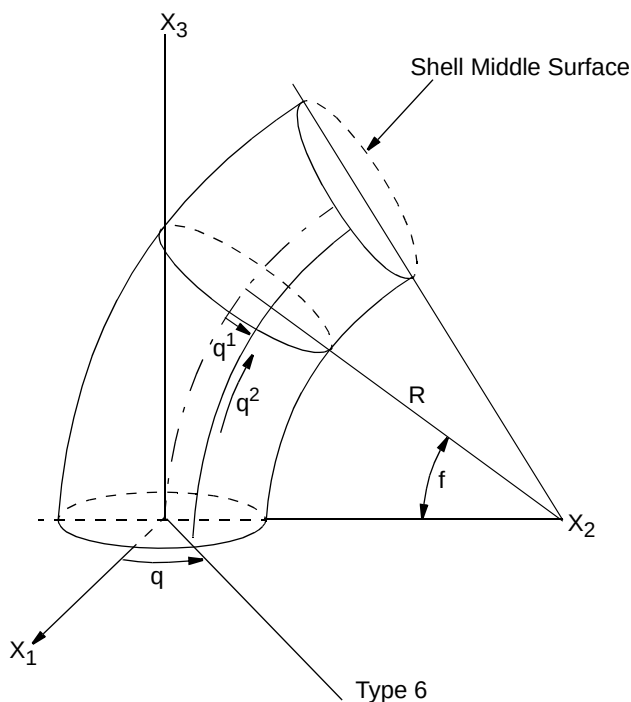


Figure 4-17 Curved Circular Cylinder

The middle surface of the shell is defined by the equations

$$\begin{aligned}x_1 &= r \cos \theta \\x_2 &= r \sin \theta \cos \phi + R(1 - \cos \phi) \\x_3 &= (R - r \sin \theta \sin \phi) \sin \phi\end{aligned}\tag{4-13}$$

The Gaussian coordinates on the surface are

$$\begin{aligned}\theta_1 &= r\theta \\ \theta_2 &= R\phi\end{aligned}\tag{4-14}$$

and form an orthonormal coordinate system. The nodal point information is

$$\theta, \phi, r, R\tag{4-15}$$

θ and ϕ in degrees. You need to specify the radii r and R only for the first nodal point.

Convert Cylindrical to Cartesian (Type 7)

Type 7 allows you to enter the coordinates for continuum elements in cylindrical coordinates, which are converted by MSC.Marc to Cartesian coordinates. In this way, you can enter R , θ , Z and obtain x , y , z where θ is given in degrees and

$$\begin{aligned}x &= R \cos \theta \\y &= R \sin \theta \\z &= Z\end{aligned}\tag{4-16}$$

Convert Spherical to Cartesian (Type 8)

Type 8 allows you to enter the coordinates for continuum elements in spherical coordinates, which are converted by MSC.Marc to Cartesian coordinates. In this way, you can enter R , θ , ϕ and obtain x , y , z where θ and ϕ are given in degrees and

$$\begin{aligned}x &= R \cos \theta \sin \phi \\y &= R \sin \theta \cos \phi \\z &= R \cos \phi\end{aligned}\tag{4-17}$$

Recommendations on Use of the FXORD Option

When a continuous surface has a line of discontinuity, for example, a complete cylinder at $\theta = 0^\circ = 360^\circ$, you must place two nodes at each nodal location on the line to allow the distinct coordinate to be input. You must use tying element type 100 to join the degrees of freedom. Generally, when different surfaces come together, you must use the intersecting shell tyings.

The **FXORD** option cannot precede the **COORDINATES** option, because it uses input from that option.

Incremental Mesh Generators

Incremental mesh generators are a collection of options available in MSC.Marc to assist you in generating the mesh. Incremental mesh generators generate connectivity lists by repeating patterns and generate nodal coordinates by interpolation. Use these options directly during the model definition phase of the input.

During the model definition phase, you can often divide the structure into regions, or blocks, for which a particular mesh pattern can be easily generated. This mesh pattern is established for each region and is associated with a single element connectivity list. Use the [CONNECTIVITY](#) option to input this element connectivity list. The incremental mesh generators then generate the remainder of the connectivity lists.

Critical nodes define the outline of the regions to be analyzed. Use the [COORDINATES](#) option to enter the critical nodes. The incremental mesh generators complete the rest and join the regions by merging nodes.

A special connectivity interpolator option generates midside nodes for elements where these nodes have not been specified in the original connectivity. A separate mesh generation run is sometimes required to determine the position of these nodes. This run can be followed by mesh display plotting.

The incremental mesh generators are listed below:

Element Connectivity Generator – The [CONN GENER](#) option repeats the pattern of the connectivity data for previously defined master elements. One element can be removed for each series of elements, allowing the program to generate a tapered mesh. Two elements can be removed for each series with triangular elements.

Element Connectivity Interpolator – The [CONN FILL](#) option completes the connectivity list by generating midside nodes. You first generate the simpler quadrilateral or brick elements without the midside nodes. You can then fill in the midside nodes with this option.

Coordinate Generator – The [NODE GENER](#) option creates a new set of nodes by copying the spacing of another specified set of nodes.

Coordinate Interpolator – The [NODE FILL](#) generates intermediate nodes on a line defined by two end nodes. The spaces between the nodes can be varied according to a geometric progression.

Coordinate Generation for Circular Arcs – The [NODE CIRCLE](#) option generates the coordinates for a series of nodes which lie on a circular arc.

Nodal Merge – The [NODE MERGE](#) option merges all nodes which are closer than a specified distance from one another and it eliminates all gaps in the nodal numbers.

Bandwidth Optimization

MSC.Marc can minimize the nodal bandwidth of a structure in several ways. The amount of storage is directly related to the size of the bandwidth, and the computation time increases in proportion to the square of the average bandwidth.

The **OPTIMIZE** option allows you to choose from several bandwidth optimization algorithms. The minimum degree algorithm should only be used if the direct sparse solver is used.

The four available **OPTIMIZE** options are listed in [Table 4-1](#).

Note: This option creates an internal node numbering that is different from your node numbering. Use your node numbering for all inputs. All output appears with your node numbering. The occurrence of gap or Herrmann elements can change the internal node numbers. On occasion, this change can result in a nonoptimal node numbering system, but this system is necessary for successful solutions.

The nodal correspondence obtained through this process can be saved and then used in subsequent analyses. This eliminates the need to go through the optimization step in later analyses. The correspondence is used to relate the user-defined node (external) numbers to the program-optimized (internal) node numbers and vice versa.

Table 4-1 Bandwidth Optimization Options

Option Number	Remarks	Solver
2	Cuthill-McKee algorithm	Profile (2)
9	Sloan	Profile (2)
10	Minimum Degree Algorithm	Sparse Direct (4) or Multifrontal (8)
11	Metis	Multifrontal (8)

Rezoning

The **REZONING** parameter defines a new mesh and transfers the state of the old mesh to the new mesh. Elements or nodes can be either added to or subtracted from the new mesh. This procedure requires a subincrement to perform the definition of the new mesh. The rezoning capability can be used for two- and three-dimensional continuum elements and for shell elements 22, 75, 138, 139, and 140. See Figure 4-18 for an example of rezoning.

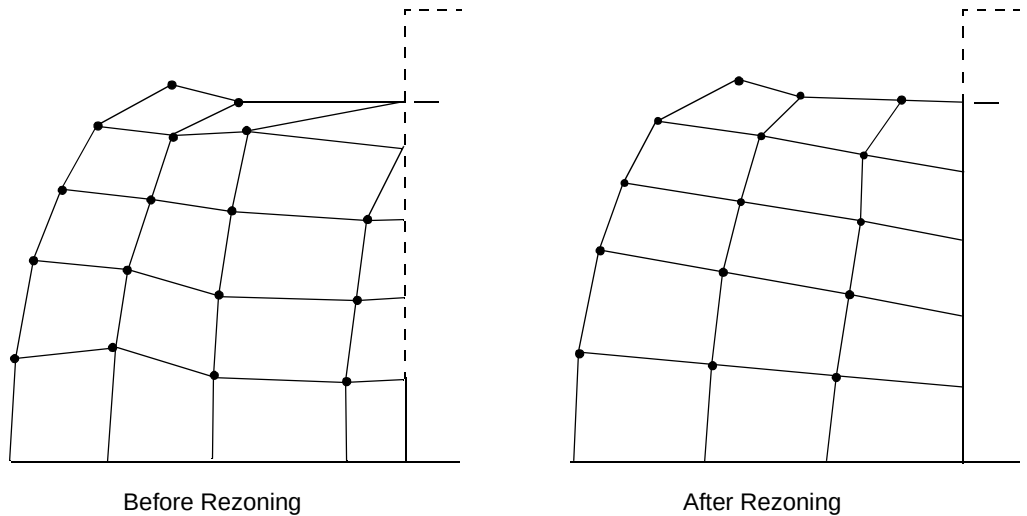


Figure 4-18 Mesh Rezoning

Substructure

MSC.Marc is capable of multilevel substructuring that includes:

- Generation of superelements
- Use of superelements in subsequent MSC.Marc analyses
- Recovery of solutions (displacements, stresses, and strains) in the individual substructures

The MSC.Marc multilevel procedure allows superelements to be used. One self-descriptive database stores all data needed during the complete analysis. You only have to ensure this database is saved after every step of the analysis.

The advantages of substructuring are the following:

- Separates linear and nonlinear parts of the model
- Allows repetition of symmetrical or identical parts of the model for linear elastic analysis
- Separates large models into multiple, moderate-size models
- Separates fixed model parts from parts of the model that may undergo design changes

A disadvantage of substructuring is the large amount of data that must be stored on the database.

Three steps are involved in a substructuring run.

- The superelement generation step is done for every superelement at a certain level.
- The use of superelements in subsequent MSC.Marc runs is done at the highest level, or is incorporated into Step 1 for the intermediate levels.
- Recovery of solutions within a certain superelement can or cannot be done for every superelement.

Substructuring in MSC.Marc is only possible for static analysis. Nonlinearities are not allowed with a superelement. You, as a user, must ensure that nonlinearities are not present.

The maximum number of levels in a complete analysis is 26. The maximum number of substructures in the complete analysis is 676.

Each step can be done in an individual run, or an unlimited number of steps can be combined into a single run.

During superelement generation in MSC.Marc, you can generate a complete new superelement or you can copy a previously defined superelement with identical or newly defined external load conditions. Any number of superelements can be formed in a generation run.

MSC.Marc offers flexibility in the use of superelements by allowing rotation or mirroring of a superelement. If a run is nonlinear, superelements are treated as linear elastic parts. At every increment, you can perform a detailed analysis of certain substructures by descending down to the desired superelements. Use the normal MSC.Marc control algorithm ([AUTO INCREMENT](#), [AUTO LOAD](#), [PROPORTIONAL INCREMENT](#)) to control the load on the superelements.

Use parameter [SUBSTRUC](#) to declare the formation of a substructure. Use the [SUBSTRUCTURE](#) model definition option to define the nodes and degrees of freedom that belong to the substructure. Use the [SUPER](#) parameter combined with the [SUPERINPUT](#) model definition data to use the substructure in a later run. Use the user subroutine [SSTRAN](#) to rotate or mirror a substructure.

Technical Background

The system of equations for a linear static structure is

$$Ku = P \quad (4-18)$$

When local degrees of freedom (subscripted $_l$) and external degrees of freedom (subscripted $_e$) are considered, this can be rewritten as

$$\begin{bmatrix} K_{ll} & K_{el} \\ K_{le} & K_{ee} \end{bmatrix} \begin{pmatrix} u_l \\ u_e \end{pmatrix} = \begin{pmatrix} P_l \\ P_e \end{pmatrix} \quad (4-19)$$

To obtain both the stiffness matrix and the load vector of the substructure, it is necessary to eliminate u_l and rewrite the above system with u_e as the only unknown splitting the above equation.

$$K_{ll}u_l + K_{el}u_e = P_l$$

and

$$K_{le}u_l + K_{ee}u_e = P_e \quad (4-20)$$

The first equation can be written as

$$u_l = -K_{ll}^{-1} \bullet K_{el}u_e + K_{ll}^{-1}P_l \quad (4-21)$$

Substituting this equation into the second

$$-K_{le}K_{ll}^{-1}K_{el}u_e + K_{le}K_{ll}^{-1}P_l + K_{ee}u_e = P_e \quad (4-22)$$

This can be rewritten as

$$K_{ee}^*u_e = P_e^* \quad (4-23)$$

where

$$K_{ee}^* = K_{ee} - K_{le}K_{ll}^{-1}K_{el} \quad (4-24)$$

and

$$P_e^* = P_e - K_{le}K_{ll}^{-1}P_l \quad (4-25)$$

K_{ee}^* and P_e^* are solved by the triangularization of K_{ll} , the forward and backward substitution of K_{el} and P_l , respectively, and premultiplication with K_{le} , K_{ee}^* , and P_e^* are used in the next part of the analysis with other substructures or with another element mesh. That analysis results in the calculation of u_e .

You can now calculate the local degrees of freedom and/or the stresses using the following procedure:

$$K_{ll} \bullet u_l = P_l - K_{el}u_e \quad (4-26)$$

which can be written as

$$u_l = K_{ll}^{-1} \bullet P_l^* \quad (4-27)$$

where

$$P_l^* = P_l - K_{el}u_e \quad (4-28)$$

The displacement of the substructure is, therefore, known, and stresses and strains can be calculated in the normal way.

Scaling Element Stiffness

Occasionally, it is desirable to perform a scalar multiplication of the stiffness, mass, and load matrix to represent a selective duplication of the finite element mesh.

The **STIFSCALE** option can be used to enter the scaling factor for each element. In this case, the global stiffness, mass, and load matrices are formed as follows:

$$K^g = \sum s_i K_i^{el}, \quad M^g = \sum s_i M_i^{el}, \quad \text{and} \quad F^g = \sum s_i f_i^{el} \quad (4-29)$$

Note that no transformation of the stiffness matrix occurs and that point loads are not scaled.

BEAM SECT Parameter

The **BEAM SECT** parameter inputs data to define the sectional properties for three-dimensional beam elements. Include this option if you are using element types 13, 77, or 79 or element types 14, 25, 76, or 78 with a noncircular section, or if you are using element types 52 or 98 and torsional and shear stiffness must be defined independently.

The convention adopted for the local (beam) coordinate system is: the first and second director (local X and Y) at a point are normal to the beam axis; the third director (local Z) is tangent to the beam axis and is in the direction of increasing distances along the beam. The director set forms a right-handed system.

Orientation of the Section in Space

The beam axis in an element is interpolated from the two nodes of the element.

$$x, y, z, \frac{dx}{ds}, \frac{dy}{ds}, \frac{dz}{ds} \quad (4-30)$$

where the last three coordinates are only used for element 13.

The beam section orientation in an element is defined by the direction of the first director (local X) at a point, and this direction is specified via the coordinates of an additional node or through the **GEOMETRY** option (see *MSC.Marc Volume C: Program Input*).

Definition of the Section

You can include any number of different beam sections in any problem. Data options following the **BEAM SECT** parameter of the MSC.Marc input (see *MSC.Marc Volume C: Program Input*) define each section. The program numbers the sections in the order they are entered. To use a particular section for a beam element, set EGEOM2 (**GEOMETRY** option, Option 2, Columns 11-20) to the floating-point value of the section number, for example, 1, 2, or 3. The program uses the default circular section for the closed section beam elements (14, 25, 76, 78) if EGEOM1 is nonzero. The program uses the default solid rectangular cross section for elements 52 or 98 if EGEOM1 is nonzero. [Figure 4-19](#) shows how the thin-walled section is defined using input data.

The rules and conventions for defining a section are listed below:

1. An $x^1 - y^1$ coordinate system defines the section, with x^1 the first director at a point of the beam. The origin of the $x^1 - y^1$ system represents the location of the node with respect to the section.
2. Enter the section as a series of branches. Branches can have different geometries, but they must form a complete traverse of the section in the input sequence so that the endpoint of one branch is the start of the next branch. It is often necessary for the traverse of the section to double back on itself. To cause the traverse to do this, specify a branch with zero thickness.
3. You must divide each branch into segments. The stress points of the section are the branch division points. The stress points are the points used for numerical integration of a section's stiffness and for output for stress results. Branch endpoints are always stress points. There must always be an even number of divisions (nonzero) in any branch. Not counting branches of zero thickness, you can use a maximum of 31 stress points (30 divisions) in a complete section.
4. Branch thickness varies linearly between the values given for branch endpoint thickness. The thickness can be discontinuous between branches. A branch is assumed to be of constant thickness equal to the thickness given at the beginning of the branch if the thickness at the end of the branch is given as an exact zero.
5. The shape of a branch is interpolated as a cubic based on the values of x^1 and y^1 and their directions, in relation to distance along the branch. The data is input at the two ends of the branch. If both dx^1/ds and dy^1/ds are given as exact zeros at both ends of the branch, the branch is assumed to be straight. The section can have a discontinuous slope at the branch ends. The beginning point of one branch must coincide with the endpoint of the previous branch. As a result, x^1 and y^1 for the beginning of a branch need to be given only for the first branch of a section.
6. Stress points are merged into one point if they are separated by a distance less than $t/10$, where t is a thickness at one of these points.

Figure 4-19 shows three sections of a beam. Notice the use of zero thickness branches in the traverse of the I section. The program provides the following data: the location of each stress point in the section, the thickness at that point, the weight associated with each point (for numerical integration of the section stiffness), and the warping function at each section.

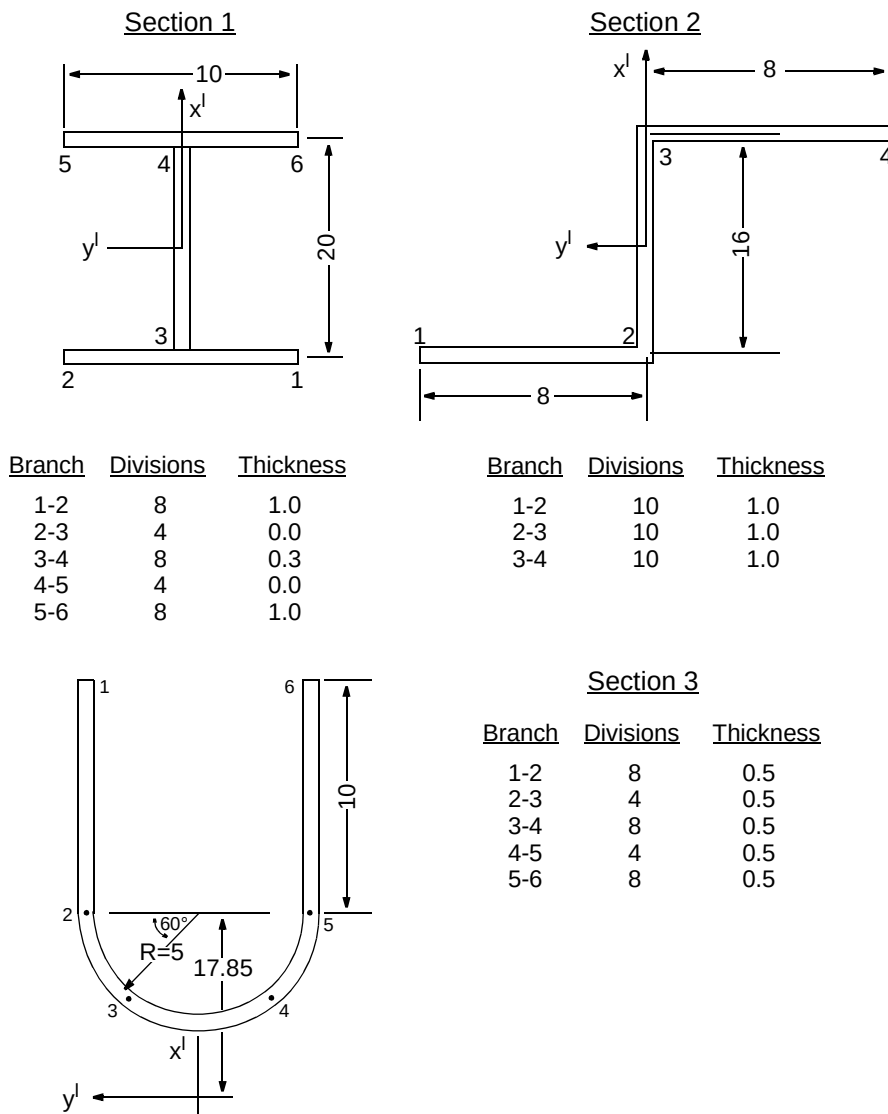


Figure 4-19 Beam Section Definition Examples

Error Analysis

You can determine the quality of the analysis by using the **ERROR ESTIMATE** option.

The **ERROR ESTIMATE** option can be used to determine the mesh quality (aspect ratio, and skewness), and how they change with deformation. While all of the MSC.Marc elements satisfy the patch test, the accuracy of the solution often depends on having regular elements. In analyses where the updated Lagrangian method is used, the mesh often becomes highly distorted during the deformation process. This option tells you when it would be beneficial to perform a rezoning step.

This option can also be used to examine the stress discontinuity in the analysis. This is a measure of the meshes ability to represent the stress gradients in the problem. Large stress discontinuities are an indication that the mesh is not of sufficient quality. This can be resolved by increasing the number of elements or choosing a higher order element.

Local Adaptivity

The adaptive mesh generation capability increases the number of elements and nodes to improve the accuracy of the solution. The capability is applicable for both linear elastic analysis and for nonlinear analysis. The capability can be used for lower-order elements, 3-node triangles, 4-node quadrilaterals, 4-node tetrahedrals, and 8-node hexahedral elements.

When used in conjunction with the [ELASTIC](#) parameter for linear analysis, the mesh is adapted and the analysis repeated until the adaptive criteria is satisfied. When used in a nonlinear analysis, an increment is performed. If necessary, this increment is followed by a mesh adjustment which is followed by the analysis of the next increment in time. While this can result in some error, as long as the mesh is not overly coarse, it should be adequate.

Number of Elements Created

The adaptive meshing procedure works by dividing an element and internally tying nodes to insure compatibility. [Figure 4-20](#) shows the process for a single quadrilateral element.

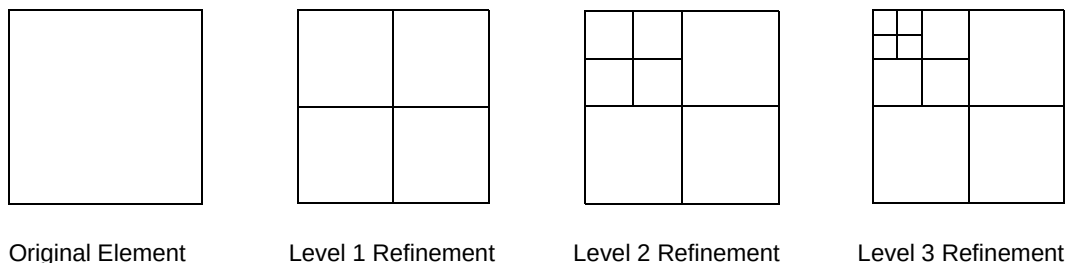


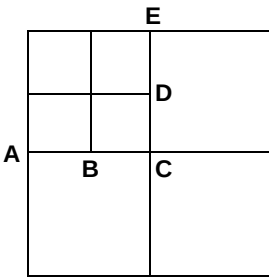
Figure 4-20 Single Quadrilateral Element Process

A similar process occurs for the triangles, tetrahedrons, and hexahedrons elements. You can observe that for quadrilaterals the number of elements expands by four with each subdivision; similarly, the number of elements increases by eight for hexahedrals. If full refinement occurs, you observe that the number of elements is $2^{(\text{level} \times 2)}$ for quadrilaterals and $2^{(\text{level} \times 3)}$ for hexahedrons elements. The number of levels also limits the amount of subdivisions that may occur.

Level	Number of Elements	
	Quadrilaterals	Hexahedrals
0	1	1
1	4	8
2	16	64
3	64	512
4	256	4096

For this reason, it is felt that the number of levels should, in general, be limited to three.

When adaptive meshing occurs, you can observe that discontinuities are created in the mesh as shown below:



To ensure compatibility node B is effectively tied to nodes A and C, and node D is effectively tied to nodes C and E. All of this occurs internally and does not conflict with other user-defined ties or contact.

Boundary Conditions

When mesh refinement occurs, boundary conditions are automatically adjusted to reflect the change in mesh. The rules listed below are followed:

1. Fixed Displacement

For both 2-D and 3-D, if both corner nodes on an edge have identical boundary conditions, the new node created on that edge has the same boundary conditions. For 3-D, if all four nodes on a face have identical boundary conditions, the new node created in the center of the face has the same boundary conditions. Note that identical here means the same in the first degree of freedom, second degree of freedom, etc. independently of one another.

2. Point Loads

The point loads remain unchanged on the original node number.

3. Distributed Loads

Distributed loads are automatically placed on the new elements. Caution should be used when using user subroutine **FORCEM** as the element numbers can be changed due to the new mesh process.

4. Contact

The new nodes generated on the exterior of a body are automatically treated as potential contact nodes. The elements in a deformable body are expanded to include the new elements created. After the new mesh is created, the new nodes are checked to determine if they are in contact. It should be noted that new nodes on the bounding surface that are completely tied to the corresponding edge nodes are not checked for contact.

CAUTION: None of the nodes of an element being subdivided should have a local coordinate system defined through the **TRANSFORMATION** option.

Location of New Nodes

When an element is refined, the default is that the new node on an edge is midside to the two corner nodes. As an alternative, the **POINTS**, **CURVES**, **SURFACES**, **ATTACH EDGE** and **ATTACH FACE** options or user subroutine **UCOORD** can be used. The **CURVES** and **SURFACES** options can be used to describe the mathematical form of the curve or surface. If the corner nodes of an edge are attached to the surface, the new node is placed upon the actual surface.

This is illustrated in **Figure 4-21** and **Figure 4-22**, where initially a single element is used to represent a circle. The circle is defined with the **CURVES** option and the original four nodes are placed on it using the **ATTACH EDGE** option. Notice that the new nodes are placed on the circle.

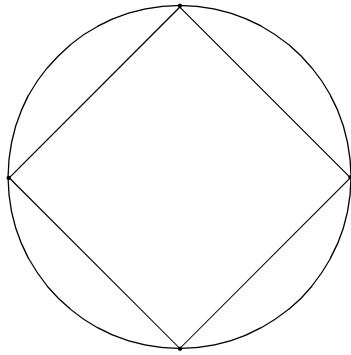
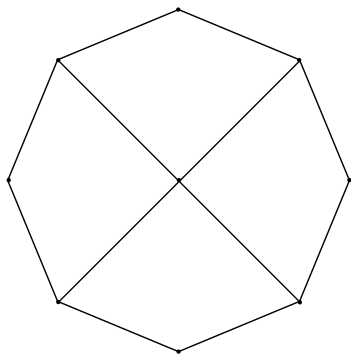
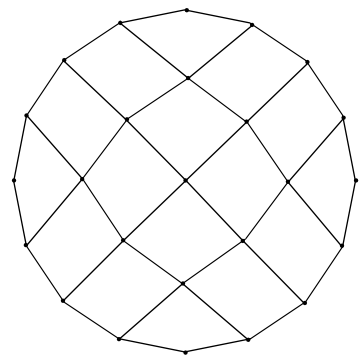


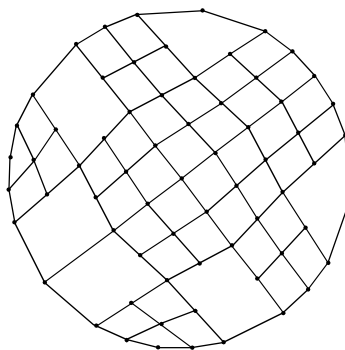
Figure 4-21 Original Mesh and Surface



Level 1 Refinement



Level 2 Refinement



Level 2 and 3 Refinement

Figure 4-22 Levels of Refinement

Adaptive Criteria

The adaptive meshing subdivision occurs when a particular adaptive criteria is satisfied. Multiple adaptive criteria can be selected using the **ADAPTIVE** model definition option. These include:

Mean Strain Energy Criterion

The element is refined if the strain energy of the element is greater than the average strain energy in a chosen set of elements times a given factor, f_1 .

$$\text{element strain energy} > \frac{\text{total strain energy}}{\text{number of elements}} * f_1 \quad (4-31)$$

Zienkiewicz-Zhu Criterion

The error norm is defined as either

$$\pi^2 = \frac{\int (\sigma^* - \sigma)^2 dV}{\int \sigma^2 dV + \int (\sigma^* - \sigma)^2 dV} \quad \gamma^2 = \frac{\int (E^* - E)^2 dV}{\int E^2 dV + \int (E^* - E)^2 dV} \quad (4-32)$$

The stress error and strain energy errors are

$$X = \int (\sigma^* - \sigma)^2 dV \quad \text{and} \quad Y = \int (E^* - E)^2 dV \quad (4-33)$$

where σ^* is the smoothed stress and σ is the calculated stress. Similarly, E is for energy.

An element is subdivided if

$$\pi > f_1 \quad \text{and}$$

$$X_{el} > f_2 * X / \text{NUMEL} + f_3 * X * f_1 / \pi / \text{NUMEL}$$

or

$$\gamma > f_1 \quad \text{and}$$

$$Y_{el} > f_4 * Y / \text{NUMEL} + f_5 * Y * f_1 / \gamma / \text{NUMEL}$$

where NUMEL is the number of elements in the mesh. If f_2 , f_3 , f_4 , and f_5 are input as zero, then $f_2 = 1.0$.

Zienkiewicz – Zhu Plastic Strain Criterion

The plastic strain error norm is defined as $\alpha^2 = \frac{\int (\epsilon^{p*} - \epsilon^p)^2 dV}{\int \epsilon^{p2} dV + \int (\epsilon^{p*} - \epsilon^p)^2 dV}$.

The plastic strain error is $A = \int (\epsilon^{p*} - \epsilon^p)^2 dV$. The allowable element plastic strain error is $AEPS = f_2 * A / \text{NUMEL} + f_3 * A * f_1 / \alpha / \text{NUMEL}$. The element will be subdivided when $\alpha > f_1$ and $A_{el} > AEPS$. NUMEL is the number of elements in the mesh.

Zienkiewicz-Zhu Creep Strain Criterion

Zienkiewicz-Zhu creep strain error norm is defined as

$\beta^2 = \frac{\int (\epsilon^{c*} - \epsilon^c)^2 dV}{\int \epsilon^{c2} dV + \int (\epsilon^{c*} - \epsilon^c)^2 dV}$. The creep strain error is $B = \int (\epsilon^{c*} - \epsilon^c)^2 dV$. The

allowable element creep strain error is

$$AECS = f_2 * B / \text{NUMEL} + f_3 * B * f_1 / \beta / \text{NUMEL}.$$

The element will be subdivided when $\beta > f_1$ and $B_{el} > AECS$. NUMEL is the number of elements in the mesh.

Equivalent Values Criterion

This method is based upon either relative or absolute testing using either the equivalent von Mises stress, the equivalent strain, equivalent plastic strain or equivalent creep strain. An element is subdivided if the current element value is a given fraction of the maximum (relative) or above a given absolute value.

$$\sigma_{vm} > f_1 \sigma_{vm}^{max} \text{ or } \sigma_{vm} > f_2$$

$$\epsilon_{vm} > f_3 \sigma_{vm}^{max} \text{ or } \epsilon_{vm} > f_4$$

Node Within A Box Criterion

An element is subdivided if it falls within the specified box. If all of the nodes of the subdivided elements move outside the box, the elements are merged back together.

It is also possible to specify that if all the children elements leave the box, they coalesce back into a single element.

The location of this box can be repositioned using user subroutine [UADAPBOX](#).

Nodes In Contact Criterion

An element is subdivided if one of its nodes is associated with a new contact condition. In the case of a deformable-to-rigid contact, this implies that the node has touched a rigid surface. For deformable-to-deformable contact, the node can be either a tied or retained node. Note that if chattering occurs, there can be an excessive number of elements generated. Use the level option to reduce this problem.

Temperature Gradient Criterion

An element is subdivided if the gradient in the element is greater than a given fraction of the maximum gradient in the solution. This is the recommended method for heat transfer.

User-defined Criterion

User subroutine [UADAP](#) can be used to prescribe a user-defined adaptive criteria.

Previously Refined Mesh Criterion

Use the refined mesh from a previous analysis as the starting point to this analysis. The information from the previous adapted analysis is read in.

Global Remeshing

In the analysis of metal or rubber, the materials may be deformed from some initial (maybe simple) shape to a final, very often, complex shape. During the process, the deformation can be so large that the mesh used to model the materials may become highly distorted, and the analysis cannot go any further without using some special techniques. Remeshing/rezoning in MSC.Marc is a useful feature to overcome the difficulties.

In a contact analysis, the basic steps of global remeshing/rezoning are:

1. When the mesh becomes too distorted because of the large deformation to continue the analysis, the analysis is stopped.
2. A new mesh is created based on the deformed shape of the contact body to be rezoned.
3. A data mapping is performed to transfer necessary data from the old, deformed mesh to the new mesh.
4. The contact tolerance is recalculated (if not specified by you) and the contact conditions are redefined.
5. The analysis continues.

With MSC.Marc 2003, the global remeshing/rezoning can be done in two- (see [Figure 4-23](#)) or three-dimensions (see [Figure 4-24](#)). Based on the different remeshing criteria specified, the program determines when remeshing/rezoning is required. The automatic remeshing can be controlled through the [ADAPT GLOBAL](#) option.

When remeshing/rezoning in 2-D, MSC.Marc finds the outline of the body to be rezoned and repairs the outline to remove possible penetration. MSC.Marc then calls the mesher to create a new mesh based on the clean outline.

When remeshing in 3-D with tetrahedral elements, MSC.Marc extracts and outputs the surface information together with other contact surface information and the meshing control parameters. A standalone program is then called to recreate 3-D surface mesh and the volume mesh.

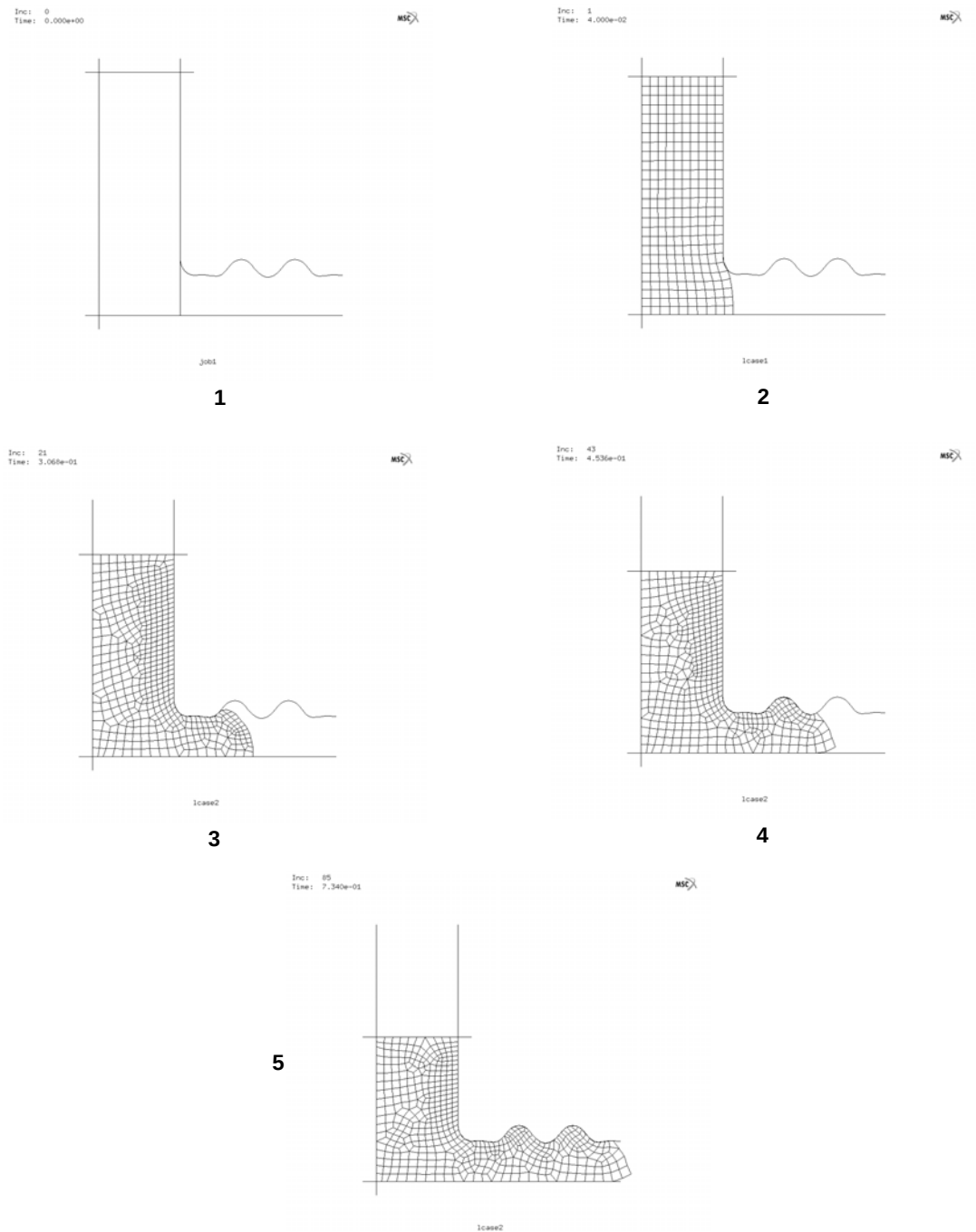


Figure 4-23 2-D Automatic Remeshing and Rezoning of a Rubber Seal

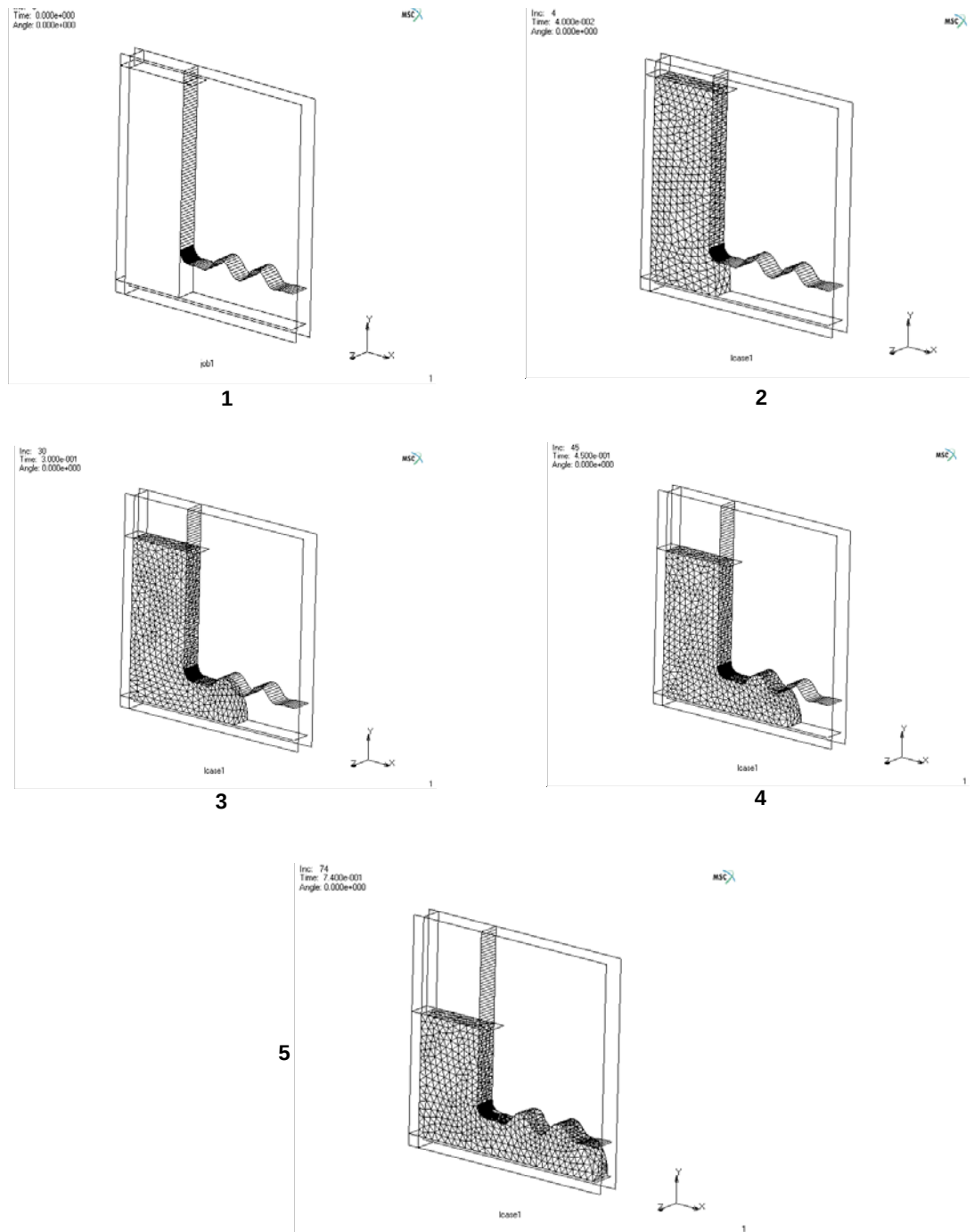


Figure 4-24 3-D Automatic Remeshing and Rezoning of a Rubber Seal

The automatic remeshing/rezoning feature uses the Updated Lagrange formulation by default. In 2-D, only the lower-order triangular and quadrilateral elements are supported. The automatic remeshing/rezoning feature can be activated using **REZONING** parameter. For 3-D remeshing with tetrahedral elements, element types **139** and **157** are supported. However, for problems involving incompressibility as encountered in plasticity or rubber, only element **157** can be used. Remeshing/rezoning can be carried out for one or more contact bodies at one increment. Different bodies can use different remeshing/rezoning criteria. The remeshing/rezoning criteria, the bodies to be rezoned, the element target length and other remeshing control parameters for the new meshes are specified via the **ADAPT GLOBAL** model definition option

Note: Only mechanical and thermal boundary conditions transferred through contact are preserved by the automatic remeshing/rezoning feature in the current release of MSC.Marc.

Remeshing Criteria

It is possible to choose one, two, three, or four remeshing criteria simultaneously.

Note: In general, frequent remeshing should be avoided for an effective and computationally efficient analysis. Also, since each remeshing and subsequent rezoning step involves interpolation and extrapolation of element variables, a possibility of error accumulation exists as the analysis progresses when remeshing occurs too frequently.

Increment

Remeshing occurs at specified increment frequency.

Element Distortion

The identified body is remeshed when the distortion in the elements becomes large.

For 2-D analysis, the distortion check is based on corner angles. Remeshing is performed if the following conditions are met:

- Any inner angle is greater than 175° or less than 5°
- Any inner angle change is greater than the user input data

For 3-D analysis, a volume ratio is measured to determine if remeshing is required. A volume ratio is calculated based on each corner node and its connecting nodes. If v is the volume of a tetrahedron formed by nodes 1, 2, 3, and 4 and s is the triangle area of nodes 2, 3, and 4, then the ratio:

$$r = \frac{h}{l}$$

$$\text{or if } h < l, r = \frac{l}{h}$$

Where h is the height and l is the average length of the triangle. They can be calculated respectively by

$$h = \frac{3v}{s}$$

$$l = \sqrt{s}$$

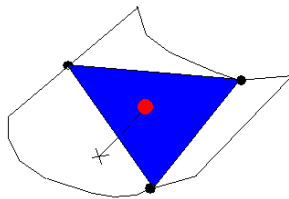
The default control ratio is 0.01. Any volume ratio of each corresponding corner node smaller than this value forces the analysis to perform remeshing. Users can change this number to control the remeshing.

Contact Penetration

The identified body is remeshed when the curvature of the contact body is such that the current mesh cannot accurately detect penetration.

For 2-D analysis, the penetration remeshing criteria is based upon examining the distance between the edge of an element and the contacted body.

For 3-D analysis, the penetration is measured from the center of each boundary element face to the contacting surface.



Remeshing is carried out if b is greater than twice the contact tolerance and less than the target element size, where the contact tolerance is 0.05 of the smallest element length and the target element size is the element size for remeshing. This check does not apply to the self-contact situation.

Remeshing is activated when the penetration distance reaches or exceeds the given penetration tolerance.

Immediate

The identified body is remeshed before performing any analysis. This control can also be used to change element types from quadrilateral elements to triangular elements or from hexahedral elements to tetrahedral elements. For example, it is possible to use this control to change an initially defined mesh using element type 7 (an 8 node-hexahedral element) to one using element type 157 (a 5 node-tetrahedral element) in order to use the tetrahedral remeshing capability.

Remeshing Techniques

For 2-D analysis, the remeshing techniques include outline extraction and repair and the mesh generation. After the outline is extracted and repaired, the mesh generator is called to create a mesh. For 2-D remeshing, the new mesh is created either through the built-in mesh generator using the overlay method or through a standalone mesh generator. When a standalone mesh generator is called, the program pauses while waiting for the mesh generator to create the new mesh. This can be memory intensive as both program and mesher are using memory. However, the program can be stopped automatically while the mesh is being created, freeing the memory for the mesh generation with the program automatically resuming after the meshing is complete. In MSC.Marc, this is accomplished by using the AUTO RESTART option, -autorst, through the command line parameter. In MSC.Marc Mentat, this is instructed through JOB→JOB PARAMETERS→REMESHING CONTROL→STOP AND RESTART.

For 3-D analysis, the outer surface of the contact bodies are extracted to a data file. This data file also contains remeshing control information. For tetrahedral meshing, a standalone 3-D mesher is called to create the surface mesh with triangles and then mesh the body with the tetrahedral elements. MSC.GS-mesher is used for the meshing. The new mesh nodes are adjusted to ensure that there is no penetration on the contact surfaces.

Mesh Generation

There are various 2-D and 3-D mesh generators available in MSC.Marc based on Advancing front, Overlay, and Delauney Triangulation techniques.

Advancing front mesher: This 2-D mesher creates either triangular, quadrilateral, or mixed triangular and quadrilateral mesh. For a given outline boundary, it starts by creating the elements along the boundary. The new boundary front is then formed when the layer of elements is created. This front advances inward until the complete region is meshed. Some smoothing technique is used to improve the quality of the elements. In general, this mesher works with any enclosed geometry and for geometry that has holes inside. The element size can be changed gradually from the boundary to the interior allowing smaller elements near the boundary with no tying constraints necessary (see [Figure 4-25](#)).

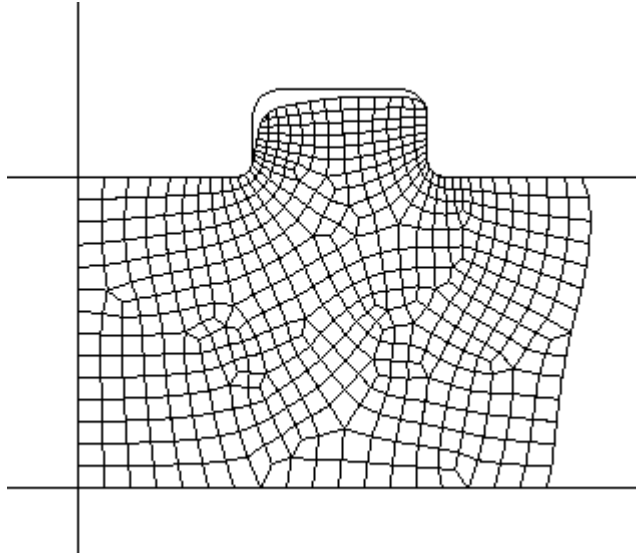


Figure 4-25 Advanced Front Meshing

Overlay meshing:

This is a quadrilateral mesh generator. The 2-D overlay mesher is included within MSC.Marc. It creates a quadrilateral mesh by forming a regular grid covering the center area of a body. A projection is then used to project all boundary nodes onto the real surface and form the outer layer elements. For the surface that is not in contact with other bodies, a cubic spline line is used to make the outline points smoother. This mesher also allows up to two level refinements on boundary where finer edges are needed to capture the geometry detail, and one level of coarsening in the interior where small elements are not necessary. This refinement and coarsening are performed using the tying constraints (see [Figure 4-26](#)).

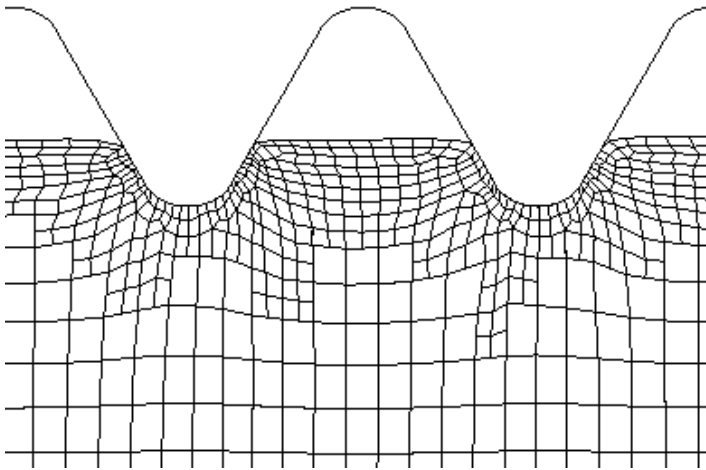


Figure 4-26 Local Refinement and Coarsening

In general, overlay meshing produces good quality elements. However, it does not take geometry with holes inside. It may not create a good mesh with geometry that has a very thin region or very irregular shape. Also, because the regular grid is created based on the global coordinate system, it may create a poor mesh if the geometry is not aligned with the global coordinate system.

Delauney Triangulation: This mesher creates only the triangular mesh. All the triangles satisfy the Delauney triangulation property. It takes all the seed points on the improved outlines as initial triangulation points. The triangulation is implemented by sequential insertion of new points into the triangulation until all the triangles satisfy the local density and quality requirement. Delauney triangulation algorithm assures the triangular mesh created has the best quality possible for the given set of points. The mesher also allows geometry to have holes inside the body and a variation of the elements with different sizes (Figure 4-27).

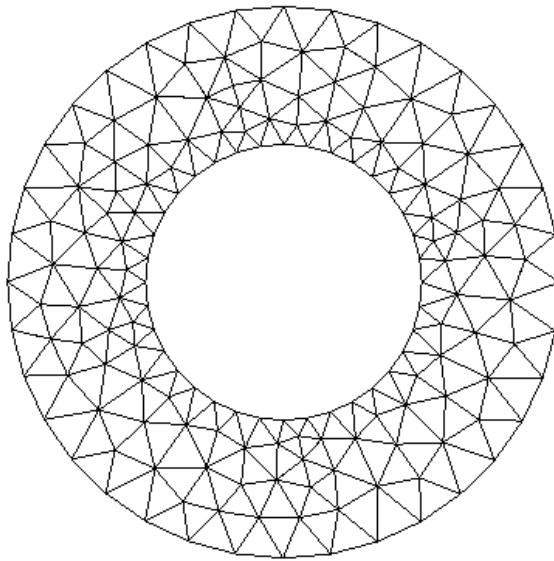


Figure 4-27 Meshing with Delauney Triangulation

MSC.Patran Tetrahedral Mesher

MOM (Meshing-On-Mesh) Surface Mesher: This is a 3-D surface mesher that creates a surface mesh based on an input mesh. The Meshing-on-mesh technology allows users to input a mesh, which is not good for the analysis, such as a distorted mesh or a mesh from the STL file.

Tetrahedral Hybrid Mesher:

This 3-D mesher is based on Delaunay triangulation and advancing front technology. It generates tetrahedral elements based on an enclosed triangular surface mesh.

2-D Outline Extraction and Repair

The outline consists of all the boundary edges of the contact body. Once the outline is extracted from the mesh, it is checked against other contacting bodies. The penetration from other contacting bodies is marked and the outline is corrected according to the penetration. If the outline is to be used for the advancing front or Delaunay mesher,

some refinement and corrections are required to obtain better outline points. These outline points become boundary nodes in the new mesh and cannot be altered during the meshing process. The following procedures are taken by the program to prepare the new outline for the remeshing:

- Step 1: Marking the hard points:** The hard points are those points that represent important features of the original outline. Hard points are points that mark the beginning or end of the boundary portion of the mesh that is in contact, and the points that represent a sharp corner (such as 90° angle).
- Step 2: Marking the points with target element size and minimum element size:** The outline points are placed based on the target element size. The refinements and the user-defined outline points are allowed to change this control. However, the minimum element size is used to make sure the outline segment is not too small for the mesh generation.
- Step 3: Marking the points with curvature consideration:** The curvature control is used to allow small outline segments to be used on the boundary where curvature radii are small. Three neighboring outline points are used to calculate the curvature radius. The associate curvature circle can then be formed by a number of line segments. With the same number of line segments used to approximate a circle, the curve with a smaller curvature circle gets a smaller line segment ([Figure 4-27](#)). Thus, we have a good variation of mesh size based on the curvature.

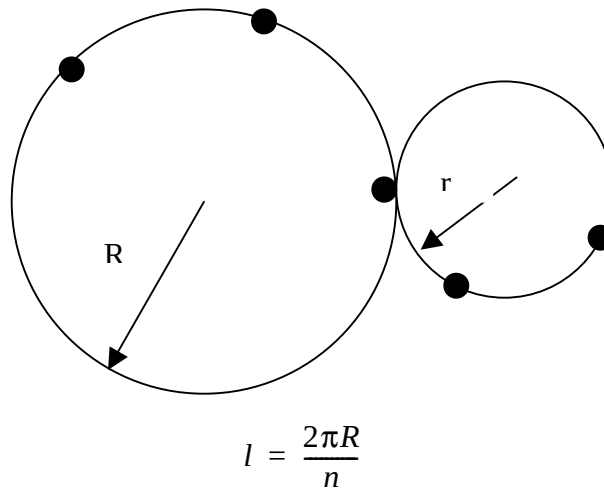


Figure 4-28 Curvature Consideration

- Step 4: Marking the points with thin region consideration:** In the area where a thin region is formed, small elements are preferred. This can be done by detecting the thin region and using smaller outline segments in the area. The segment length used for the thin area is to allow at least three elements to be presented across the thin area.
- Step 5: Smoothing the outline points:** Smoothing is required on the outline so that the segment length can gradually vary.
- Step 6: Interpolations:** Interpolation is the actual process to create a new outline based on the extracted outline and the marking of the outline points. Linear interpolation is used on the contact area to prevent penetration into the other contact bodies. Cubic spline line interpolation is used for the free surfaces.

3-D Surface Extraction and Meshing

Similar to the 2-D outline extraction, 3-D surface faces are extracted before calling the mesh generator. If the original mesh is a hexahedral mesh, the boundary element faces are converted into triangles. The contact information on each element face is also extracted and output to a data file. A standalone mesher reads the information and perform the following steps:

- Step 1:** Construct surface information for contact projection and volume check.
- Step 2:** Generate surface mesh using MSC.Patran surface mesher (MOM). Surface elements are created observing the basic geometry features and contact conditions.
- Step 3:** Check contact penetration of new nodes and adjust coordinates if penetration is found.
- Step 4:** Generate volume mesh with MSC.Patran Hybrid mesher.
- Step 5:** Output the mesh to a data file.

Remeshing Based on the Target Number of Elements

Instead of giving the element size, users can give the target number of elements for the remeshing. A uniform mesh assumption is used to compute the element size with the target number of elements. If the target number of elements is not provided, the number of elements in the current mesh is used. For 2-D analysis, the number of elements in the new mesh can also be controlled by using a percentage tolerance to ensure that the new mesh does not have too many or too few elements. However, this tolerance control requires remeshing trials and it cannot be used with the AUTO RESTART option.

CHAPTER
5

Structural Procedure Library

- Linear Analysis
- Nonlinear Analysis
- Fracture Mechanics
- Dynamics
- Rigid-Plastic Flow
- Superplasticity
- Soil Analysis
- Design Sensitivity Analysis
- Design Optimization
- Transfer Axisymmetric Analysis Data to 3-D Analysis
- Steady State Rolling Analysis
- References

This chapter describes the analysis procedures in MSC.Marc applicable to structural problems. These procedures range from simple linear elastic analysis to complex nonlinear analysis. A large number of options are available, but you need to consider only those capabilities that are applicable to your physical problem. This chapter provides technical background information as well as usage information about these capabilities.

Linear Analysis

Linear analysis is the type of stress analysis performed on linear elastic structures. Because linear analysis is simple and inexpensive to perform and generally gives satisfactory results, it is the most commonly used structural analysis. Nonlinearities due to material, geometry, or boundary conditions are not included in this type of analysis. The behavior of an isotropic, linear, elastic material can be defined by two material constants: Young’s modulus E , and Poisson’s ratio ν .

MSC.Marc allows you to perform linear elastic analysis using any element type in the program. Various kinematic constraints and loadings can be prescribed to the structure being analyzed; the problem can include both isotropic and anisotropic elastic materials.

The principle of superposition holds under conditions of linearity. Therefore, several individual solutions can be superimposed (summed) to obtain a total solution to a problem.

Linear analysis does not require storing as many quantities as does nonlinear analysis; therefore, it uses the core memory more sparingly. The **ELASTIC** parameter uses the assembled and decomposed stiffness matrices to arrive at repeated solutions for different loads.

Note: Linear analysis is always the default analysis type in the MSC.Marc program.

Linear analysis in MSC.Marc requires only the basic input. [Table 5-1](#) shows a subset of the MSC.Marc options and parameters which are often used for linear analysis.

Table 5-1 Basic Input

Type	Name
Parameter	TITLE SIZING ELEMENTS ELASTIC ALL POINTS CENTROID ADAPTIVE FOURIER END

Table 5-1 Basic Input

Type	Name
Model Definition	CONNECTIVITY COORDINATES GEOMETRY ISOTROPIC FIXED DISP DIST LOADS POINT LOAD CASE COMBIN END OPTION

More complex linear analyses require additional data blocks.

1. Parameter **ELASTIC** allows solutions for the same structural system with different loadings (multiple loading analysis). When using the **ELASTIC** parameter, you must apply total loads, rather than incremental quantities (for example., total force, total moment, total temperature) in subsequent increments.
2. The **RESTART** option, used with the **ELASTIC** parameter and/or **CASE COMBIN** option, stores individual load cases in a restart file. You can also store the decomposed stiffness matrix for later analyses.
3. Model definition option **CASE COMBIN** combines the results obtained from different loading cases previously stored in a restart file.
4. The **ADAPTIVE** option can be used to improve the accuracy of the analysis.
5. The **J-INTEGRAL** option allows the study of problems of linear fracture mechanics.
6. The **FOURIER** option allows the analysis of axisymmetric structures subjected to arbitrary loadings.
7. Model definition option **ORTHOTROPIC** or **ANISOTROPIC** activates the anisotropic behavior option. In addition, user subroutines **ANELAS**, **HOOKLW**, **ANEXP**, and **ORIENT** define the mechanical and thermal anisotropy and the preferred orientations.
8. You can use both the linear **SPRINGS** and **FOUNDATION** options in a linear stress analysis.

Accuracy

It is difficult to predict the accuracy of linear elastic analysis without employing special error estimation techniques. An inaccurate solution usually exhibits itself through one or more of the following phenomena:

- Strong discontinuities in stresses between elements
- Strong variation in stresses within an element
- Stresses that oscillate from element to element

Error Estimates

The **ERROR ESTIMATE** option can also be used to obtain an indication of the quality of the results. You can have the program evaluate the geometric quality of the mesh by reporting the aspect ratios and skewness of the elements. In a large deformation updated Lagrange analysis, you can also observe how these change during the analysis, which indicates mesh distortion. When the mesh distortion is large, it is a good idea to do a rezoning step.

The **ERROR ESTIMATE** option can also be used to evaluate the stress discontinuity between elements. MSC.Marc first calculates a nodal stress based upon the extrapolated integration point values. These nodal values are compared between adjacent elements and reported. Large discrepancies indicate an inability of the mesh to capture high stress gradients, in which case you should refine the mesh and rerun the analysis.

The **ERROR ESTIMATE** option can be used for either linear or nonlinear analysis.

Adaptive Meshing

The **ADAPTIVE** option can be used to insure that a certain level of accuracy is achieved. The elastic analysis is repeated with a new mesh until the level of accuracy requested is achieved.

Fourier Analysis

Through Fourier expansion, MSC.Marc analyzes axisymmetric structures that are subjected to arbitrary loading. The **FOURIER** option is available only for linear analysis.

During Fourier analysis, a three-dimensional analysis decouples into a series of independent two-dimensional analyses, where the circumferential distribution of displacements and forces are expressed in terms of the Fourier series. Both mechanical and thermal loads can vary arbitrarily in the circumferential direction. You can determine the structure's total response from the sum of the Fourier components.

The Fourier formulation is restricted to axisymmetric structures with linear elastic material behavior and small strains and displacements. Therefore, conditions of linearity are essential and material properties must remain constant in the circumferential direction.

To use Fourier expansion analysis in MSC.Marc, the input must include the following information:

- The **FOURIER** parameter allocates storage for the series expansion.
- Fourier model definition blocks for as many series as are needed to describe tractions, thermal loading, and boundary conditions. Number the series sequentially in the order they occur during the **FOURIER** model definition input. Three ways to describe the series are listed below:
 - Specify coefficients $a_0, a_1, b_1 \dots$ on the Fourier model definition blocks.
 - Describe $F(\theta)$ (where θ is the angle in degrees about the circumference) in point-wise fashion with an arbitrary number of pairs $[\theta, F(\theta)]$ given on the blocks. MSC.Marc forms the corresponding series coefficients.
 - Generate an arbitrary number of $[\theta, F(\theta)]$ pairs using the user subroutine **UFOUR** and let the program calculate the series coefficients.

You can obtain the total solution at any position around the circumference by superposing the components already calculated after completion of all increments required by the analysis. The **CASE COMBIN** option calculates this total solution by summing the individual harmonics which are stored in the restart file.

The number of steps or increments needed for analysis depends on the number of harmonics that are chosen. For a full analysis with symmetric and antisymmetric load cases, the total number of increments equals twice the number of harmonics.

Table 5-2 shows which Fourier coefficients are used for a given increment.

Table 5-2 Fourier Coefficients – Increment Number

LOAD TERMS			
INC.	1st DOF,Z	2nd DOF,R	3rd DOF,θ
0	a_0	a_0	0
1	0	0	a_0
2	a_1	a_1	b_1
3	b_1	b_1	a_1
†	.	.	.
†	.	.	.
†	.	.	.
2n	a_n	a_n	b_n
2n + 1	b_n	b_n	a_n

The magnitude of concentrated forces should correspond to the value of the ring load integrated around the circumference. Therefore, if the Fourier coefficients for a varying ring load $p(\theta)$ are found from the $[\theta, p(\theta)]$ distribution, where $p(\theta)$ has the units of force per unit length, the force magnitude given in the point load block should equal the circumference of the loaded ring. If $p(\theta)$ is in units of force per radian, the point load magnitude should be 2π .

The Fourier series can be found for varying pressure loading from $[\theta, p(\theta)]$ input with p expressed in force per unit area. MSC.Marc calculates the equivalent nodal forces and integrates them around the circumference. The distributed load magnitude in the distributed loads block should be 1.0.

Table 5-3 shows the elements in the program that can be used for Fourier analysis.

Table 5-3 Elements Used for Fourier Analysis

Element Type	Description
62	8-node
73	8-node with reduced integration
63	8-node for incompressible behavior
74	8-node for incompressible behavior with reduced integration
90	3-node shell

Technical Background

The general form of the Fourier series expansion of the function $F(\theta)$ is shown in the equation below.

$$F(\theta) = a_0 + \sum_{n=1}^{\infty} (a_n \cos n\theta + b_n \sin n\theta) \quad (5-1)$$

This expression expands the displacement function in terms of sine and cosine terms. A symmetric and an antisymmetric problem are formulated for each value of n .

The displacements for the symmetric case, expressed in terms of their nodal values, are

$$\begin{aligned} u^n &= [N_1, N_2, \dots] \cos n\theta \{u^n\}^e \\ v^n &= [N_1, N_2, \dots] \cos n\theta \{v^n\}^e \\ w^n &= [N_1, N_2, \dots] \sin n\theta \{w^n\}^e \end{aligned} \quad (5-2)$$

Nodal forces are

$$\begin{aligned} Z &= Z_0 + \sum_{n=1}^{\infty} Z^n \cos n\theta \\ R &= R_0 + \sum_{n=1}^{\infty} R^n \cos n\theta \\ T &= T_0 + \sum_{n=1}^{\infty} T^n \sin n\theta \end{aligned} \quad (5-3)$$

The value $n = 0$ is a special case in Fourier analysis. If only the symmetric expansion terms are used, the formulation defaults to the fully axisymmetric two-dimensional analysis. The antisymmetric case for $n = 0$ yields a solution for the variable θ that corresponds to loading in the tangential direction. Analyze axisymmetric solids under pure torsion in this way.

Modal Shapes and Buckling Load Estimations During a Fourier Analysis

During a Fourier analysis, MSC.Marc can be asked to estimate both the modal shapes and buckling loads for each harmonic in the analysis. In either case, the program performs a Fourier analysis first and then estimates the modal shapes/buckling load at prescribed harmonic numbers. In addition to the input data required for a Fourier analysis (**FOURIER** parameter and **FOURIER** model definition option), the following must also be added: **DYNAMIC** parameter and **MODAL INCREMENT** model definition option; **BUCKLE** parameter and **BUCKLE INCREMENT** model definition option, for Fourier modal shape and Fourier buckling load estimations, respectively.

In the Fourier modal analysis, the mass matrix in the eigenvalue equation is a constant matrix. The stiffness matrix in the eigenvalue equation is the one associated with a prescribed harmonic of the Fourier analysis. The expression of the eigenvalue equation is:

$$[K^m]\phi - \omega^2[M^0]\phi = 0 \quad (5-4)$$

where $[K^m]$ is the stiffness matrix associated with the m^{th} harmonic of the Fourier analysis and $[M^0]$ is a constant matrix. Multiple modes for each harmonic can be extracted.

Similarly, in a Fourier buckling analysis, the stiffness matrices in the eigenvalue equation are (respectively); the linear elastic stiffness matrix and the geometric stiffness matrix associated with the prescribed harmonic of the Fourier analysis.

The eigenvalue equation is expressed

$$[K^m]\phi - \lambda[K_g^m]\phi = 0 \quad (5-5)$$

where $[K^m]$ is the linear elastic stiffness matrix and $[K_g^m]$ is the geometric stiffness matrix, associated with the m^{th} harmonic of the Fourier analysis. The stresses used in the calculation of the geometric stiffness matrix are those associated with the symmetric load case, $m = 0$. Multiple buckling load estimations for each harmonic are also available.

Nonlinear Analysis

The finite element method can be used for nonlinear, as well as linear, problems. Early development of nonlinear finite element technology was mostly influenced by the nuclear and aerospace industries. In the nuclear industry, nonlinearities are mainly due to the nonlinear, high-temperature behavior of materials. Nonlinearities in the aerospace industry are mainly geometric in nature and range from simple linear buckling to complicated post-bifurcation behavior. Nonlinear finite element techniques have become popular in metal forming manufacturing processes, fluid-solid interaction, and fluid flow. In recent years, the areas of biomechanics and electromagnetics have seen an increasing use of finite elements.

A problem is nonlinear if the force-displacement relationship depends on the current state (that is, current displacement, force, and stress-strain relations). Let u be a generalized displacement vector, P a generalized force vector, and K the stiffness matrix. The expression of the force-displacement relation for a nonlinear problem is

$$P = K(P, u)u \quad (5-6)$$

Linear problems form a subset of nonlinear problems. For example, in classical linear elastostatics, this relation can be written in the form

$$P = Ku \quad (5-7)$$

where the stiffness matrix K is independent of both u and P . If the matrix K depends on other state variables that do not depend on displacement or loads (such as temperature, radiation, moisture content, etc.), the problem is still linear.

Similarly, if the mass matrix is a constant matrix, the following undamped dynamic problem is also linear:

$$P = M\ddot{u} + Ku \quad (5-8)$$

There are three sources of nonlinearity: material, geometric, and nonlinear boundary conditions. Material nonlinearity results from the nonlinear relationship between stresses and strains. Considerable progress has been made in attempts to derive the continuum or macroscopic behavior of materials from microscopic backgrounds, but, up to now, commonly accepted constitutive laws are phenomenological. Difficulty in obtaining experimental data is usually a stumbling block in mathematical modeling of material behavior. A plethora of models exist for more commonly available materials like elastomers and metals. Other material model of considerable practical importance are: composites, viscoplastics, creep, soils, concrete, powder, and foams. [Figure 5-1](#)

shows the elastoplastic, elasto-viscoplasticity, and creep. Although the situation of strain hardening is more commonly encountered, strain softening and localization has gained considerable importance in recent times.

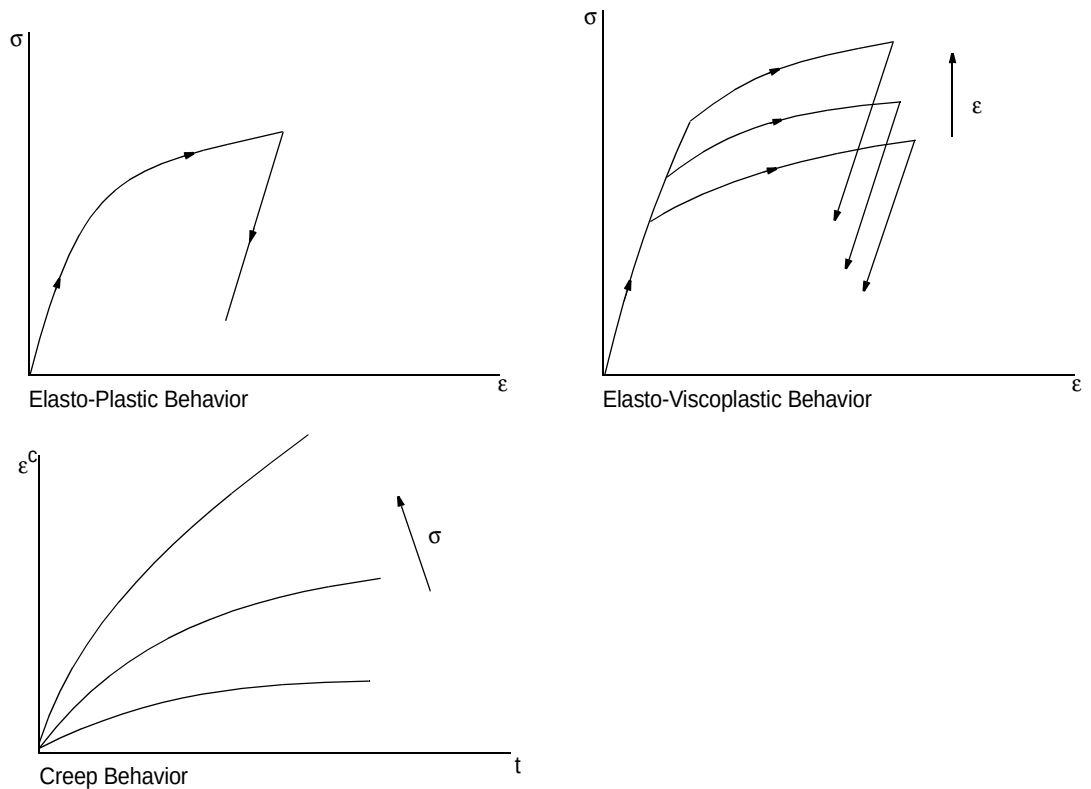


Figure 5-1 Material Nonlinearity

Geometric nonlinearity results from the nonlinear relationship between strains and displacements on the one hand and the nonlinear relation between stresses and forces on the other hand. If the stress measure is conjugate to the strain measure, both sources of nonlinearity have the same form. This type of nonlinearity is mathematically well defined, but often difficult to treat numerically. Two important types of geometric nonlinearity occur:

- a. The analysis of buckling and snap-through problems (see [Figure 5-2](#) and [Figure 5-3](#)).

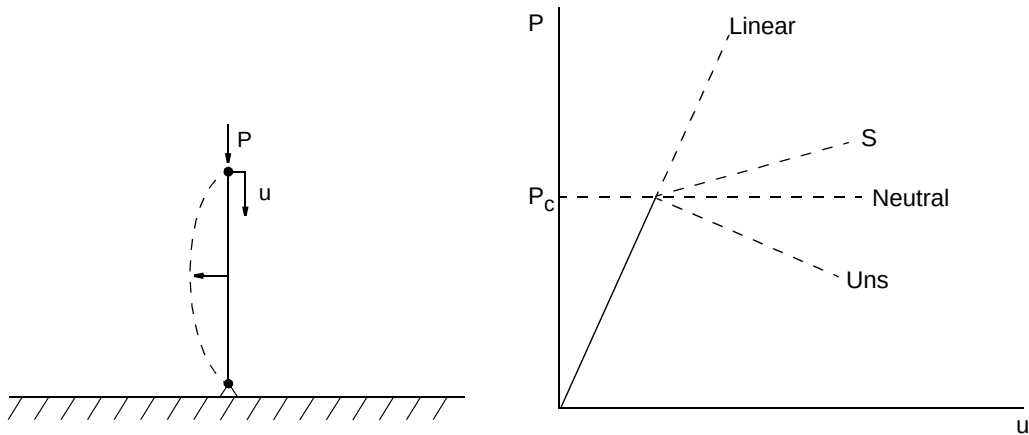


Figure 5-2 Buckling

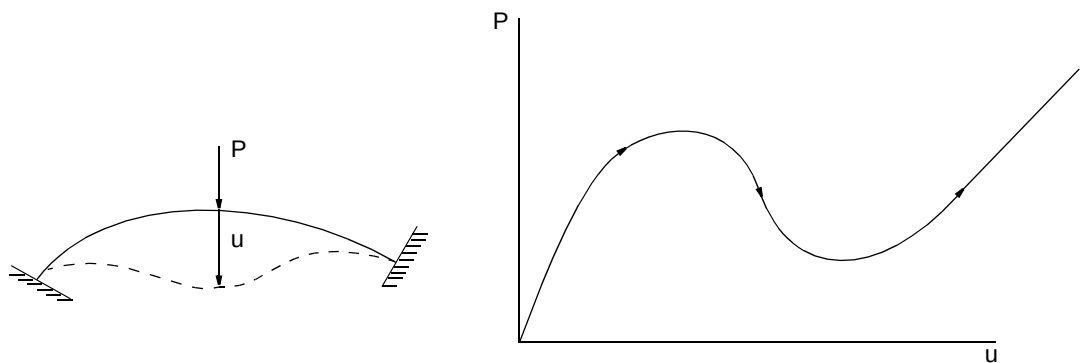


Figure 5-3 Snap-Through

- b.** Large strain problems such as manufacturing, crash, and impact problems. In such problems, due to large strain kinematics, the mathematical separation into geometric and material nonlinearity is nonunique.

Boundary conditions and/or loads can also cause nonlinearity. Contact and friction problems lead to nonlinear boundary conditions. This type of nonlinearity manifests itself in several real life situations; for example, metal forming, gears, interference of mechanical components, pneumatic tire contact, and crash (see [Figure 5-4](#)). Loads on a structure cause nonlinearity if they vary with the displacements of the structure. These loads can be conservative, as in the case of a centrifugal force field (see [Figure 5-5](#)); they can also be nonconservative, as in the case of a follower force on a

cantilever beam (see [Figure 5-6](#)). Also, such a follower force can be locally nonconservative, but represent a conservative loading system when integrated over the structure. A pressurized cylinder (see [Figure 5-7](#)) is an example of this.

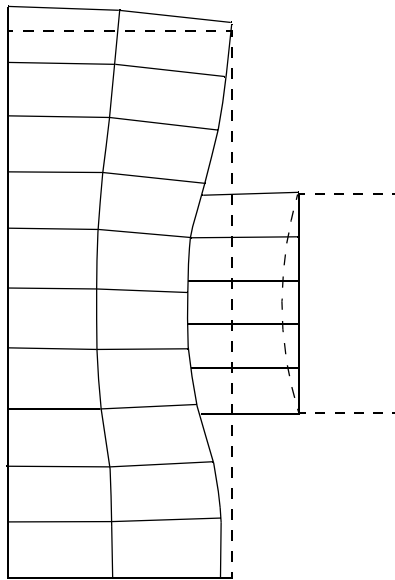


Figure 5-4 Contact and Friction Problem

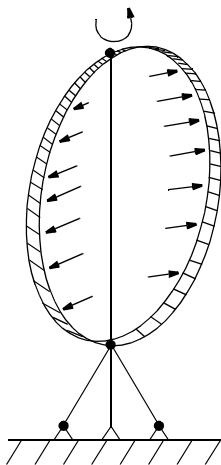


Figure 5-5 Centrifugal Load Problem (Conservative)

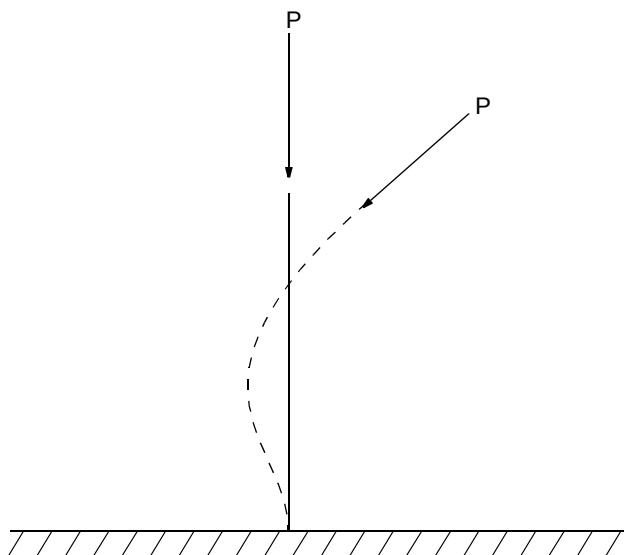


Figure 5-6 Follower Force Problem (Nonconservative)

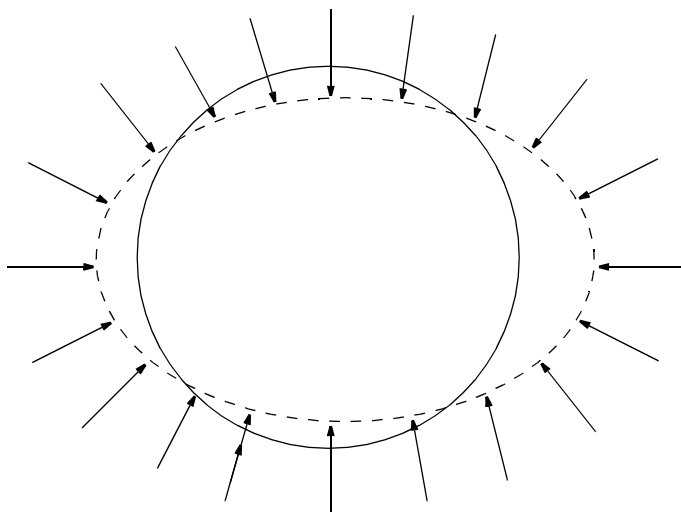


Figure 5-7 Pressurized Cylinder (Globally Conservative)

The three types of nonlinearities are described in detail in the following sections.

Geometric Nonlinearities

Geometric nonlinearity leads to two types of phenomena: change in structural behavior and loss of structural stability.

There are two natural classes of large deformation problems: the large displacement, small strain problem and the large displacement, large strain problem. For the large displacement, small strain problem, changes in the stress-strain law can be neglected, but the contributions from the nonlinear terms in the strain displacement relations cannot be neglected. For the large displacement, large strain problem, the constitutive relation must be defined in the correct frame of reference and is transformed from this frame of reference to the one in which the equilibrium equations are written.

The collapse load of a structure can be predicted by performing an eigenvalue analysis. If performed after the linear solution (increment zero), the Euler buckling estimate is obtained. An eigenvalue problem can be formulated after each increment of load; this procedure can be considered a nonlinear buckling analysis even though a linearized eigenvalue analysis is used at each stage.

The kinematics of deformation can be described by the following approaches:

A. Lagrangian Formulation

B. Eulerian Formulation

C. Arbitrary Eulerian-Lagrangian (AEL) Formulation

The choice of one over another can be dictated by the convenience of modeling physics of the problem, rezoning requirements, and integration of constitutive equations.

A. Lagrangian Formulation

In the Lagrangian method, the finite element mesh is attached to the material and moves through space along with the material. In this case, there is no difficulty in establishing stress or strain histories at a particular material point and the treatment of free surfaces is natural and straightforward.

The Lagrangian approach also naturally describes the deformation of structural elements; that is, shells and beams, and transient problems, such as the indentation problem shown in [Figure 5-8](#).

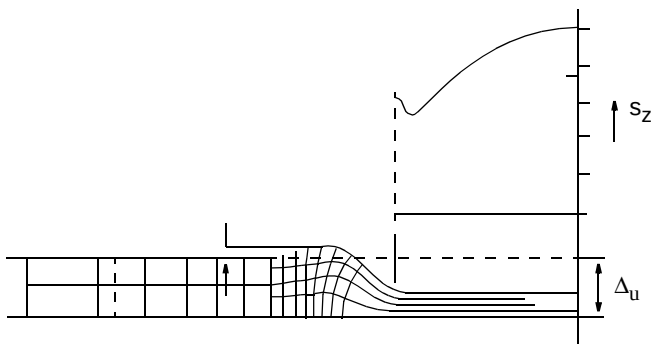


Figure 5-8 Indentation Problem with Pressure Distribution on Tool

This method can also analyze steady-state processes such as extrusion and rolling. Shortcomings of the Lagrangian method are that flow problems are difficult to model and that the mesh distortion is as severe as the deformation of the object. Severe mesh degeneration is shown in [Figure 5-9b](#). However, recent advances in adaptive meshing and rezoning have alleviated the problems of premature termination of the analysis due to mesh distortions as shown in [Figure 5-9c](#).

The Lagrangian approach can be classified in two categories: the total Lagrangian method and the updated Lagrangian method. In the total Lagrangian approach, the equilibrium is expressed with the original undeformed state as the reference; in the updated Lagrangian approach, the current configuration acts as the reference state. The kinematics of deformation and the description of motion is given in [Figure 5-10](#) and [Table 5-4](#).

Depending on which option you use, the stress and strain results are given in different form as discussed below. If none of the following parameters, [LARGE DISP](#), [UPDATE](#), or [FINITE](#), are used, the program uses and prints “engineering” stress and strain measures. These measures are suitable only for analyses without large incremental or total rotation or large incremental or total strains.

Using only the [LARGE DISP](#) parameter, MSC.Marc uses the total Lagrangian method. The program uses and prints second Piola-Kirchhoff stress and Green-Lagrange strain. These measures are suitable for analysis with large incremental rotations and large incremental strains.

With the combination of [LARGE DISP](#) and [UPDATE](#), MSC.Marc uses Cauchy stresses and true strains. This combination of parameters is suitable for analyses with small incremental rotations and small incremental strains. Stress and strain components are printed with respect to the current state.

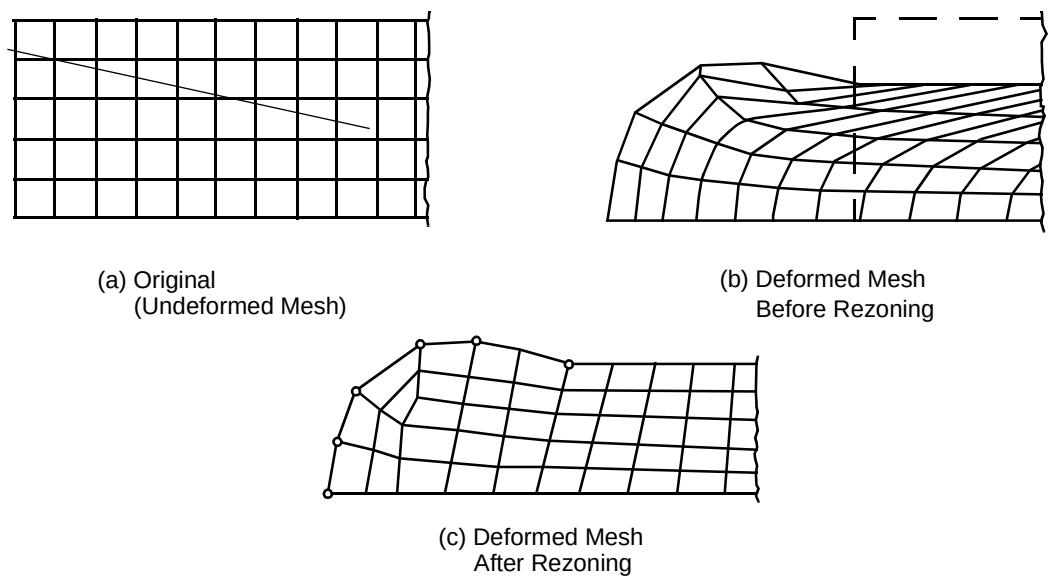


Figure 5-9 Rezoning Example

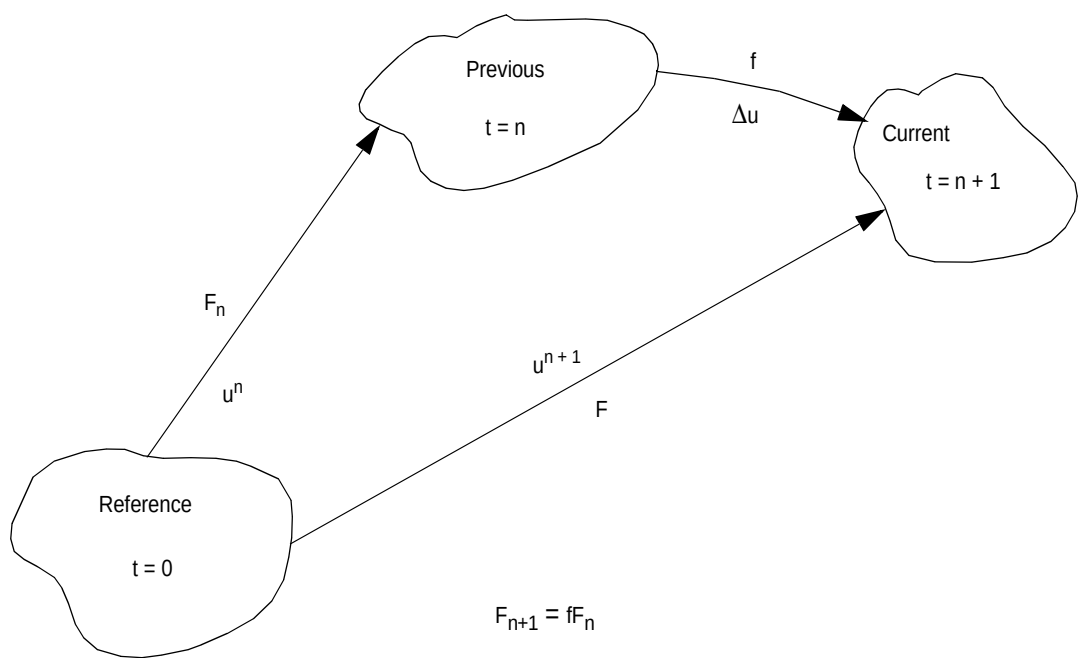


Figure 5-10 Description of Motion

Table 5-4 Kinematics and Stress-Strain Measures in Large Deformation

Measures	Configuration	Reference (t = 0 or n)	Current (t = n + 1)
Coordinates		X	x
Deformation Tensor		C (Right Cauchy-Green)	b (Left Cauchy-Green)
Strain Measure		E (Green-Lagrange) F (Deformation Gradient)	e (Logarithmic)
Stress Measure		S (second Piola-Kirchhoff) P (first Piola-Kirchhoff)	σ (Cauchy)

The combination of **LARGE DISP**, **UPDATE**, and **FINITE** (with constant dilatation used on the **GEOMETRY** option) or **PLASTICITY** (option 3 and 4) results in a complete large strain plasticity formulation with B-Bar method to satisfy incompatibility using the updated Lagrange procedure. The use of **PLASTICITY** (option 3 and 4) parameter obviates the need of the **CONSTANT DILATATION** parameter. The program internally uses true (Cauchy) stress and rotation neutralized strains. In the case of proportional straining, this method leads to logarithmic strains.

Note: For materials exhibiting large strain plasticity with volumetric changes (for example, soils, powder, snow, wood) only **LARGE DISP**, **FINITE**, and **UPDATE** should be used. Use of **CONSTANT DILATATION** or **PLASTICITY** parameters will enforce the incompressibility condition and, in such materials, yield incorrect and nonphysical behavior.

The use of **PLASTICITY**,5 results in a complete large strain plasticity formulation in mixed framework using the updated Lagrange. The program uses the multiplicative decomposition of deformation gradient. The results are given in true (Cauchy) stress and logarithmic strains.

PLASTICITY,3 and **PLASTICITY**,5 are best suited for isotropic materials in continuum elements, while **PLASTICITY**,4 is valid for isotropic and anisotropic materials in continuum and shell elements. In this sense, **PLASTICITY**,4 is the most general formulation. MSC.Marc automatically switches to the correct parameter if a certain limitation is encountered with the choice of the initially chosen **PLASTICITY**.

Large strain rubber elasticity can be modeled in either total Lagrange (**ELASTICITY**,1 : second Piola-Kirchhoff stress, Green Lagrange strain) or updated Lagrange (**ELASTICITY**,2 : Cauchy stress, Logarithmic strain) framework.

PLASTICITY,5 and **ELASTICITY**,2 parameters can be used with either displacement or Herrmann elements.

Parameter	Kinematics	Formulation
PLASTICITY,1	Total Lagrange	Small strain elasto-plasticity, mean normal return mapping, additive decomposition of strain rates.
PLASTICITY,3	Updated Lagrange	Large strain elasto-plasticity, mean normal return mapping, additive decomposition of strain rates. Best suited for isotropic, continuum elements. Equivalent to: LARGE DISP, FINITE, UPDATE, and CONSTANT DILATATION.
PLASTICITY,4	Updated Lagrange	Large strain elasto-plasticity, multistage return mapping, additive decomposition of strain rates. Best suited for isotropic and Anisotropy materials for continuum and shell elements.
PLASTICITY,5	Updated Lagrange	Large strain, radial return multiplicative decomposition of deformation gradient. Available for isotropic, continuum elements (except plane strain).
ELASTICITY,1	Total Lagrange	Large strain formulation for elastomeric materials.
ELASTICITY,2	Updated Lagrange	Large strain formulation for elastomeric materials (except plane stress).

Theoretically and numerically, if formulated mathematically correct, the two formulations yield exactly the same results. However, integration of constitutive equations for certain types of material behavior (for example, plasticity) make the implementation of the total Lagrange formulation inconvenient. If the constitutive equations are convected back to the original configuration and proper transformations are applied, then both formulations are equivalent. However, for deformations involving excess distortions, ease of rezoning favors the updated Lagrangian formulation. This is reflected in the fact that a rezoned mesh in the current state is mapped back to excessively distorted mesh leading to negative Jacobian in the total Lagrangian formulation.

The terminology total and updated Lagrangian has been used with some vagueness [1, 2]. In this document, for a sequence of incremental motions at $t = 0, 1, 2, \dots, n$ and $n + 1$, the total Lagrangian formulation entails the use of $t = 0$ configuration as reference; while in the updated Lagrangian configuration, the $t = n + 1$ (unequilibrated) configuration is the reference.

Total Lagrangian Procedure

The total Lagrangian procedure can be used for linear or nonlinear materials, in conjunction with static or dynamic analysis. Although this formulation is based on the initial element geometry, the incremental stiffness matrices are formed to account for previously developed stress and changes in geometry.

This method is particularly suitable for the analysis of nonlinear elastic problems (for instance, with the Mooney or Ogden material behavior or the user subroutine [HYPELA](#)). The total Lagrangian approach is also useful for problems in plasticity and creep, where moderately large rotations but small strains occur. A case typical in problems of beam or shell bending. However, this is only due to the approximations involved.

To activate the large displacement (total Lagrangian approach) option in MSC.Marc, use the [LARGE DISP](#) parameter. Include the [FOLLOW FOR](#) parameter for follower force (for example, centrifugal or pressure load) problems. This parameter forms all distributed loads on the basis of the current geometry. Do not use the [CENTROID](#) parameter with this parameter. Always use residual load corrections with this parameter. To input control tolerances for large displacement analysis, use model definition option [CONTROL](#).

In the total Lagrangian approach, the equilibrium can be expressed by the principle of virtual work as:

$$\int_{V_0} S_{ij} \delta E_{ij} dV = \int_{V_0} b_i^0 \delta \eta_i dV + \int_{A_0} t_i^0 \delta \eta_i dA \quad (5-9)$$

Here S_{ij} is the symmetric second Piola-Kirchhoff stress tensor, E_{ij} is the Green-Lagrange strain, b_i^0 is the body force in the reference configuration, t_i^0 is the traction vector in the reference configuration, and η_i is the virtual displacements.

Integrations are carried out in the original configuration at $t = 0$. The strains are decomposed in total strains for equilibrated configurations and the incremental strains between $t = n$ and $t = n + 1$ as:

$$E_{ij}^{n+1} = E_{ij}^n + \Delta E_{ij} \quad (5-10)$$

while the incremental strains are further decomposed into linear, ΔE_{ij}^{ℓ} and nonlinear, $\Delta E_{ij}^{n\ell}$ parts as:

$$\Delta E_{ij} = \Delta E_{ij}^{\ell} + \Delta E_{ij}^{n\ell}$$

where $\Delta E'$ is the linear part of the incremental strain expressed as:

$$\Delta E' = \frac{1}{2} \left[\frac{\partial \Delta u_i}{\partial X_j} + \frac{\partial \Delta u_j}{\partial X_i} \right] + \frac{1}{2} \left[\left(\frac{\partial u_k^n}{\partial X_i} \right) \left(\frac{\partial \Delta u_k}{\partial X_j} \right) + \left(\frac{\partial u_k^n}{\partial X_j} \right) \left(\frac{\partial \Delta u_k}{\partial X_i} \right) \right] \quad (5-11)$$

The second term in the bracket in Equation 5-11 is the initial displacement effect.

$\Delta E^{n'}$ is the nonlinear part of the incremental strain expressed as:

$$\Delta E^{n'} = \frac{1}{2} \left[\left(\frac{\partial \Delta u_k}{\partial X_i} \right) \left(\frac{\partial \Delta u_k}{\partial X_j} \right) \right] \quad (5-12)$$

Linearization of equilibrium of Equation 5-9 yields:

$$\{K_0 + K_1 + K_2\} \delta u = F - R \quad (5-13)$$

where K_0 is the small displacement stiffness matrix defined as

$$(K_0)_{ij} = \int_{V_0} \beta_{imn}^0 D_{mnpq} \beta_{pqj}^0 dV$$

K_1 is the initial displacement stiffness matrix defined as

$$(K_1)_{ij} = \int_{V_0} \{ \beta_{imn}^u D_{mnpq} \beta_{pqj}^u + \beta_{imn}^u D_{mnpq} \beta_{pqj}^0 + \beta_{imn}^u D_{mnpq} \beta_{pqj}^u \} dv$$

in the above equations, β_{imn}^0 and β_{imn}^u are the constant and displacement dependent symmetric shape function gradient matrices, respectively, and D_{mnpq} is the material tangent,

and K_2 is the initial stress stiffness matrix

$$(K_2)_{ij} = \int_{V_0} N_{i,k} N_{j,l} S_{kl} dV$$

in which S_{kl} is the second Piola-Kirchhoff stresses and $N_{i,k}$ is the shape function gradient matrix.

Also, δu is the correction displacement vector. Refer to Chapter 11 for more details on the solution procedures. F and R are the external and internal forces, respectively.

This Lagrangian formulation can be applied to problems if the undeformed configuration is known so that integrals can be evaluated, and if the second Piola-Kirchhoff stress is a known function of the strain. The first condition is not usually met for fluids, because the deformation history is usually unknown. For solids, however, each analysis usually starts in the stress-free undeformed state, and the integrations can be carried out without any difficulty.

For viscoelastic fluids and elastic-plastic and viscoplastic solids, the constitutive equations usually supply an expression for the rate of stress in terms of deformation rate, stress, deformation, and sometimes other (internal) material parameters. The relevant quantity for the constitutive equations is the rate of stress at a given material point.

It, therefore, seems most obvious to differentiate the Lagrangian virtual work equation with respect to time. The rate of virtual work is readily found as

$$\int_{V_0} \left[\dot{S}_{ij} \delta E_{ij} + S_{ij} \frac{\partial v_k}{\partial X_i} \frac{\partial \delta \eta_k}{\partial X_j} \right] dV = \int_{V_0} \dot{b}_i \delta \eta_i dV + \int_{A_0} \dot{t}_i \delta \eta_i dA \quad (5-14)$$

This formulation is adequate for most materials, because the rate of the second Piola-Kirchhoff stress can be written as

$$\dot{S}_{ij} = \dot{S}_{ij}(\dot{E}_{kl}, S_{mn}, E_{pq}) \quad (5-15)$$

For many materials, the stress rate is even a linear function of the strain rate

$$\dot{S}_{ij} = D_{ijkl}(S_{mn}, E_{pq}) \dot{E}_{kl} \quad (5-16)$$

Equation 5-14 supplies a set of linear relations in terms of the velocity field. The velocity field can be solved noniteratively and the displacement can be obtained by time integration of the velocities.

The second Piola-Kirchhoff stress for elastic and hyperelastic materials is a function of the Green-Lagrange strain defined below:

$$S_{ij} = S_{ij}(E_{kl}) \quad (5-17)$$

If the stress is a linear function of the strain (linear elasticity)

$$S_{ij} = D_{ijkl} E_{kl} \quad (5-18)$$

the resulting set of equations is still nonlinear because the strain is a nonlinear function of displacement.

Updated Lagrangian Procedure

The Updated Lagrange formulation takes the reference configuration at $t = n + 1$. True or Cauchy stress and an energetically conjugate strain measure, namely the true strain, are used in the constitutive relationship.

The updated Lagrangian approach is useful in:

- a. analysis of shell and beam structures in which rotations are large so that the nonlinear terms in the curvature expressions may no longer be neglected, and
- b. large strain plasticity analysis, for calculations which the plastic deformations cannot be assumed to be infinitesimal.

In general, this approach can be used to analyze structures where inelastic behavior (for example, plasticity, viscoplasticity, or creep) causes the large deformations. The (initial) Lagrangian coordinate frame has little physical significance in these analyses since the inelastic deformations are, by definition, permanent. For these analyses, the Lagrangian frame of reference is redefined at the last completed iteration of the current increment.

You can use the updated procedure with or without the **LARGE DISP** parameter. When you use the **LARGE DISP** parameter, MSC.Marc takes into account the effect of the internal stresses by forming the initial stress stiffness. MSC.Marc also calculates the strain increment to second order accuracy to allow large rotation increments. It is recommended that the **LARGE DISP** parameter be used in conjunction with the **UPDATE** parameter.

The **UPDATE** parameter in MSC.Marc (used with or without the **LARGE DISP** parameter) defines a new (Lagrangian) frame of reference at the beginning of each increment. This option is suitable for analysis of problems of large total rotation but small strain. If analysis of large plastic strain is required, use the **FINITE** parameter in addition to the **UPDATE** parameter. MSC.Marc uses true stress (σ) and logarithmic strains. Hence, you must input your stress-strain curves as true stress versus logarithmic strain.

The **PLASTICITY** parameter with options 3, 4, and 5 utilize the updated Lagrange procedure for elastic-plastic analysis. The **ELASTICITY** parameter with option 2 utilizes the updated Lagrange procedure for large strain elasticity (Mooney, Ogden, Arruda-Boyce, or Gent).

It is instructive to derive the stiffness matrices for the updated Lagrangian formulation starting from the virtual work principle in [Equation 5-9](#).

Direct linearization of the left-hand side of Equation 5-9 yields:

$$\int_{V_0} S_{ij}(d(\delta E_{ij}))dV = \int_{V_{n+1}} \nabla \eta_{ik} \sigma_{kj} \nabla \Delta u_{ij} dv \quad (5-19)$$

where Δu and η are actual incremental and virtual displacements respectively, and σ_{kj} is Cauchy stress tensor.

$$\int_{V_0} dS_{ij} \delta E_{ij} dV = \int_{V_{n+1}} \nabla^s \eta_{ij} L_{ijkl} \nabla^s (\Delta u_{kl}) dv \quad (5-20)$$

∇^s denotes the symmetric part of ∇ , which represents the gradient operator in the current configuration. Also, in Equation 5-19 and Equation 5-20, three identities are used:

$$\sigma_{ij} = \frac{1}{J} F_{im} S_{mn} F_{jn}$$

$$\delta E_{ij} = F_{mi} \nabla^s \eta_{mn} F_{nj} \quad (5-21)$$

and

$$L_{ijkl} = \frac{1}{J} F_{im} F_{jn} F_{kp} F_{lq} D_{mnpq}$$

in which D_{mnpq} represents the material moduli tensor in the reference configuration which is convected to the current configuration, L_{ijkl} . This yields:

$$\{K_1 + K_2\} \delta u = F - R \quad (5-22)$$

where K_1 is the material stiffness matrix written as

$$(K_1)_{ij} = \int_{V_{n+1}} \beta_{imn} L_{mnpq} \beta_{pqj}$$

in which β_{imn} is the symmetric gradient operator-evaluated in the current configuration and σ_{kl} is the Cauchy stresses

and K_2 is the geometric stiffness matrix written as

$$(K_2)_{ij} = \int_{V_{n+1}} \sigma_{kl} N_{i,k} N_{j,l} dv$$

while F and R are the external and internal forces, respectively.

Keeping in view that the reference state is the current state, a rate formulation analogous to Equation 5-14 can be obtained by setting:

$$F_{ij} = \delta_{ij}, \quad \delta E_{ij} = \delta d_{ij}, \quad \frac{\partial}{\partial X_i} = \frac{\partial}{\partial x_i}, \quad S_{ij} = \sigma_{ij} \quad (5-23)$$

where F is the deformation tensor, and d is the rate of deformation. Hence,

$$\int_{V_{n+1}} \left[\overset{\nabla}{\sigma}_{ij} \delta d_{ij} + \sigma_{ij} \frac{\partial v_k}{\partial x_i} \frac{\partial \delta \eta_k}{\partial x_j} \right] dv = \int_{V_{n+1}} \dot{b}_i \delta \eta_i dv + \int_{A_{n+1}} \dot{t}_i \delta \eta_i da \quad (5-24)$$

in which b_i and t_i is the body force and surface traction, respectively, in the current configuration

In this equation, $\overset{\nabla}{\sigma}_{ij}$ is the Truesdell rate of Cauchy stress which is essentially a Lie derivative of Cauchy stress obtained as:

$$\overset{\nabla}{\sigma}_{ij} = \overline{F_{in} (J F_{nk}^{-1} \sigma_{kl} F_{ml}^{-1}) F_{mj}} \quad (5-25)$$

The Truesdell rate of Cauchy stress is materially objective implying that if a rigid rotation is imposed on the material, the Truesdell rate vanishes, whereas the usual material rate does not vanish. This fact has important consequences in the large deformation problems where large rotations are involved. The constitutive equations can be formulated in terms of the Truesdell rate of Cauchy stress as:

$$\overset{\nabla}{\sigma}_{ij} = L_{ijk\ell} d_{k\ell} \quad (5-26)$$

B. Eulerian Formulation

In analysis of fluid flow processes, the Lagrangian approach results in highly distorted meshes since the mesh convects with the material. Hence, an alternative formulation, namely Eulerian, is used to describe the motion of the body. In this method, the finite

element mesh is fixed in space and the material flows through the mesh. This approach is particularly suitable for the analysis of steady-state processes, such as the steady-state extrusion or rolling processes shown in [Figure 5-11](#)

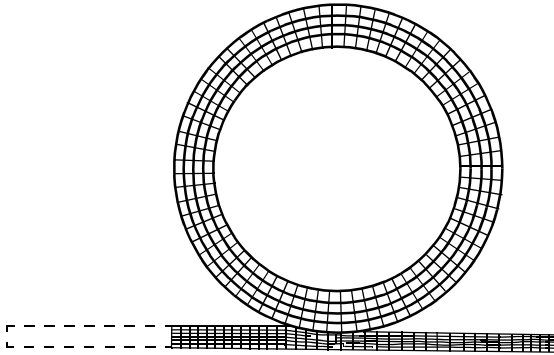


Figure 5-11 Rolling Analysis

The governing differential equations of equilibrium for fluid flow through an enclosed volume are now written as:

$$\frac{D(\rho v_i)}{Dt} = \rho b_i + \frac{\partial \sigma_{ij}}{\partial x_j} \quad (5-27)$$

where, $\frac{D}{Dt}$ is the material time derivative of a quantity. For an incompressible fluid, [Equation 5-27](#) along with continuity equation (mass conservation) yields:

$$\rho \frac{\partial v_i}{\partial t} + \rho v_j \frac{\partial v_i}{\partial x_j} = \rho b_i + \frac{\partial \sigma_{ij}}{\partial x_j} \quad (5-28)$$

The left-hand side of [Equation 5-28](#) represents the local rate of change augmented by the convection effects. The same principle can be called to physically explain the material time derivative of Cauchy stress; that is, Truesdell rate of Cauchy stress. It can be seen from [Equation 5-25](#) that:

$$\dot{\sigma}_{ij} = \dot{\sigma}_{ij} - \frac{\partial v_i}{\partial x_k} \sigma_{kj} - \sigma_{ik} \frac{\partial v_j}{\partial x_k} + \sigma_{ij} \frac{\partial v_k}{\partial x_k} \quad (5-29)$$

The second and third terms on the right-hand side represent the convection effects. The last term vanishes for a completely incompressible material; a condition enforced in the rigid-plastic flow of solids.

C. Arbitrary Eularian-Lagrangian (AEL) Formulation

In the AEL formulation referential system, the grid moves independently from the material, yet in a way that spans the material at any time. Hence, a relationship between derivative with respect to the material and grid derivative is expressed as:

$$\left(\frac{\partial}{\partial t} \right) = \left(\frac{\partial}{\partial t} \right)^* + c_i \left(\frac{\partial}{\partial x_i} \right), i \quad (5-30)$$

where c_i is the relative velocity between the material particle, v_i^p and the mesh velocity, v_i^m ; for example,

$$c_i = v_i^p - v_i^m \quad (5-31)$$

The second or latter term represents the convective effect between the grid and the material. Note that for $v_i^p = v_i^m$, a purely Eularian formulation is obtained. The equation of momentum, for instance, can be represented as:

$$\rho \dot{v}_i^* + \rho (v_j^p - v_j^m) \frac{\partial v_i}{\partial x_j} = \frac{\partial \sigma_{ij}}{\partial x_j} + b_i \quad (5-32)$$

Due to its strong resemblance to the pure Eularian formulation, AEL is also called quasi-Eularian formulation.

Nonlinear Boundary Conditions

There are three types of problems associated with nonlinear boundary conditions: contact, nonlinear support, and nonlinear loading. The contact problem might be solved through the use of special gap elements of the **CONTACT** option. Nonlinear support might involve nonlinear springs and/or foundations. Sometimes nonlinearities due to rigid links that become activated or deactivated during an analysis can be modeled through adaptive linear constraints. Nonlinear loading is present if the loading system is nonconservative, as is the case with follower forces or frictional slip effects.

Discontinuities are inherent in the nature of many of these nonlinearities, making the solution by means of incremental linear approximations difficult. Some of the most severe nonlinearities in mechanics are introduced by nonlinear boundary conditions. It is, therefore, very important to be aware of potential problem areas and to have a good understanding of the underlying principles. This awareness and understanding enables you to validate numerical answers and to take alternative approaches if an initial attempt fails.

Contact Problems

Contact problems are commonly encountered in physical systems. Some examples of contact problems are the interface between the metal workpiece and the die in metal forming processes, pipe whip in piping systems, and crash simulation in automobile designs.

Contact problems are characterized by two important phenomena: gap opening and closing and friction. As shown in [Figure 5-12](#), the gap describes the contact (gap closed) and separation (gap open) conditions of two objects (structures). Friction influences the interface relations of the objects after they are in contact. The gap condition is dependent on the movement (displacement) of the objects, and friction is dependent on the contact force as well as the coefficient of (Coulomb) friction at contact surfaces. The analysis involving gap and friction must be carried out incrementally. Iterations can also be required in each (load/time) increment to stabilize the gap-friction behavior.

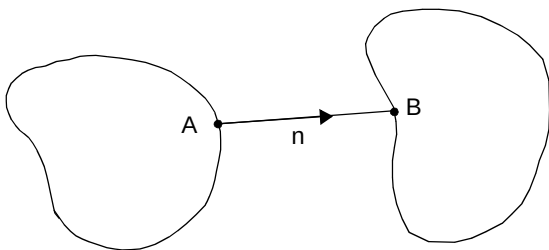


Figure 5-12 Normal Gap Between Potentially Contacting Bodies

Two options are available in MSC.Marc for the simulation of a contact problem. A detailed description of these options (gap-friction element and the [CONTACT](#) option) is given in [Chapter 8, Contact](#).

Nonlinear Support

MSC.Marc provides two options for the modeling of support conditions: springs and elastic foundations. Both linear and nonlinear springs can be specified in the input. In a nonlinear problem, the spring stiffness and the equivalent spring stiffness of the elastic foundation can also be modified through a user subroutine. In the nonlinear spring option, the incremental force in the spring is

$$\Delta F = K(\Delta u_2 - \Delta u_1) \quad (5-33)$$

where K is the spring stiffness, Δu_2 is the displacement increment of the degree of freedom at the second end of the spring, and Δu_1 is the displacement increment of the degree of freedom at the first end of the spring.

Use the **SPRINGS** model definition option for the input of linear and nonlinear spring data. User subroutine **USPRNG** may also be used to specify the value of K based on the amount of previous deformation for nonlinear springs. In dynamic analysis, the **SPRINGS** option can also be used to define a dashpot. In thermal analysis or electrical analysis (heat transfer, Joule heating, heat transfer pass of a coupled analysis), the **SPRINGS** option can be used to define a thermal or electrical link.

In the elastic nonlinear **FOUNDATION** option, the elements in MSC.Marc can be specified as being supported on a frictionless (nonlinear) foundation. The foundation supports the structure with an increment force per unit area given by

$$\Delta P_n = K(u_n)\Delta u_n \quad (5-34)$$

where K is the equivalent spring stiffness of the foundation (per unit surface area), and Δu_n is the incremental displacement of the surface at a point in the same direction as ΔP_n .

To input nonlinear foundation data, use the **FOUNDATION** model definition option.

To specify the value of K for the nonlinear equivalent spring stiffness based on the amount of previous deformation of the foundation, use the user subroutine **USPRNG**.

Nonlinear Loading

When the structure is deformed, the directions and the areas of the surface loads are changed. For most deformed structures, such changes are so small that the effect on the equilibrium equation can be ignored. But for some structures such as flexible shell structure with large pressure loads, the effects on the results can be quite significant so that the surface load effects have to be included in the finite element equations.

MSC.Marc forms both pressure stiffness and pressure terms based on current deformed configuration with the **FOLLOW FOR** parameter. The **FOLLOW FOR** parameter should be used with the **LARGE DISP** parameter and the **CENTROID** parameter should not be included due to the use of the residual load correction.

A special case of nonlinear loading is the resultant pressure due to a gas in an enclosed cavity. In such problems, the pressure changes as the volume changes based upon the ideal gas law. This is discussed in Chapter 9, in the **Cavity Pressure Loading** section.

Buckling Analysis

Buckling analysis allows you to determine at what load the structure will collapse. You can detect the buckling of a structure when the structure's stiffness matrix approaches a singular value. You can extract the eigenvalue in a linear analyses to obtain the linear buckling load. You can also perform eigenvalue analysis for buckling load in a nonlinear problem based on the incremental stiffness matrices.

The buckling option estimates the maximum load that can be applied to a geometrically nonlinear structure before instability sets in. To activate the buckling option in the program, use the parameter **BUCKLE**. If a nonlinear buckling analysis is performed, also use the parameter **LARGE DISP**.

Use the history definition option **BUCKLE** to input control tolerances for buckling load estimation (eigenvalue extraction by a power sweep or Lanczos method). You can estimate the buckling load after every load increment. The **BUCKLE INCREMENT** option can be used if a collapse load calculation is required at multiple increments.

The linear buckling load analysis is correct when you take a very small load step in increment zero, or make sure the solution has converged before buckling load analysis (if multiple increments are taken). Linear buckling (after increment zero) can be done without using the **LARGE DISP** parameter, in which case the restriction on the load step size no longer applies. This value should be used with caution, as it is not conservative in predicting the actual collapse of structures.

In a buckling problem that involves material nonlinearity (for example, plasticity), the nonlinear problem must be solved incrementally. During the analysis, a failure to converge in the iteration process or nonpositive definite stiffness signals the plastic collapse.

For extremely nonlinear problems, the **BUCKLE** option cannot produce accurate results. In that case, the history definition option **AUTO INCREMENT** allows automatic load stepping in a quasi-static fashion for both geometric large displacement and material (elastic-plastic) nonlinear problems. The option can handle elastic-plastic snap-through phenomena. Therefore, the post-buckling behavior of structures can be analyzed.

The buckling option solves the following eigenvalue problem by the inverse power sweep method:

$$[K + \lambda \Delta K_G(\Delta u, u, \Delta \sigma)] \phi = 0 \quad (5-35)$$

where ΔK_G is assumed to be a linear function of the load increment ΔP to cause buckling.

The geometric stiffness ΔK_G used for the buckling load calculation is based on the stress and displacement state change at the start of the last increment. However, the stress and strain states are not updated during the buckling analysis. The buckling load is therefore estimated by:

$$P(\text{beginning}) + \lambda \Delta P \quad (5-36)$$

where for increments greater than 1, $P(\text{beginning})$ is the load applied at the beginning of the increment prior to the buckling analyses, and λ is the value obtained by the power sweep or Lanczos method.

The control tolerances for the inverse power sweep method are the maximum number of iterations in the power sweep and the convergence tolerance. The power sweep terminates when the difference between the eigenvalues in two consecutive sweeps divided by the eigenvalue is less than the tolerance. The Lanczos method concludes when the normalized difference between all eigenvalues satisfies the tolerance. The maximum number of iterations and the tolerance are specified through the **BUCKLE** history definition option.

Perturbation Analysis

The buckling mode can be used to perform a perturbation analysis of the structure. In the manual mode, a buckling increment is performed upon request and the coordinates are perturbed by a fraction of the buckling mode or eigenvector. You can enter an individual eigenvector number and the fraction or can request that a combination of modes be used. In the subsequent increments, the coordinates are:

$$X = X + f \frac{\phi}{|\phi|} \text{ or } X = X + \sum f_i \frac{\phi_i}{|\phi_i|} \quad (5-37)$$

The manual mode can be activated by using the **BUCKLE INCREMENT** model definition, or **BUCKLE** load incrementation option. In the automatic mode, the program checks for a nonpositive definite system during the solution phase. When this occurs, it automatically performs a buckle analysis during the next increment and updates the coordinates. The automatic mode can be activated by using the **BUCKLE INCREMENT** option. Also, be sure to force the solution of the nonpositive definite system through the **CONTROL** option or **PRINT** parameter.

Material Nonlinearities

In a large strain analysis, it is usually difficult to separate the kinematics from the material description. [Table 5-5](#) lists the characteristics of some common materials.

Table 5-5 Common Material Characteristics

Material	Characteristics	Examples	Models
Composites	Anisotropic: 1) layered, $ds_{ij} = C_{ijk\ell} d\epsilon_{k\ell}$ 21 constants 2) Fiber reinforced, $S = \frac{E}{2}(T^t C T - 1)$ one-dimensional strain in fibers	Bearings, aircraft panels Tires, glass/epoxy	Composite continuum elements Rebars
Creep	Strains increasing with time under constant load. Stresses decreasing with time under constant deformations. Creep strains are non-instantaneous.	Metals at high temperatures, polymide films, semiconductor materials	ORNL Norton Maxwell
Elastic	Stress functions of instantaneous strain only. Linear load-displacement relation.	Small deformation (below yield) for most materials: metals, glass, wood	Hookes Law
Elasto-plasticity	Yield condition flow rule and hardening rule necessary to calculate stress, plastic strain. Permanent deformation upon unloading.	Metals Soils	von Mises Isotropic Cam -Clay Hill's Anisotropic
Hyperelastic	Stress function of instantaneous strain. Nonlinear load-displacement relation. Unloading path same as loading.	Rubber	Mooney Ogden Arruda-Boyce Gent
Hypoelastic	Rate form of stress-strain law	Concrete	Buyukozturk
Viscoelastic	Time dependence of stresses in elastic material under loads. Full recovery after unloading.	Rubber, Glass, industrial plastics	Simo Model Narayanaswamy
Viscoplastic	Combined plasticity and creep phenomenon	Metals Powder	Power law Shima Model
Shape Memory	Superelastic and shape memory effect with phase transformations.	Biomedical stents, Satellite antennae	Aurichio, Thermo-mechanical

A complete description of the material types mentioned in the is given in [Chapter 7](#). However, some no characteristics and procedural considerations of some commonly encountered materials behavior are listed next.

Inaccuracies in experimental data, misinterpretation of material model parameters and errors in user-defined material law are some common sources of error in the analysis from the materials viewpoint. It is useful to check the material behavior by running a single element test with prescribed displacement and load boundary conditions in uniaxial tension and shear.

Large Strain Elasticity

Structures composed of elastomers, such as tires and bushings, are typically subjected to large deformation and large strain. An elastomer is a polymer, such as rubber, which shows a nonlinear elastic stress-strain behavior. The large strain elasticity capability in MSC.Marc deals primarily with elastomeric materials. These materials are characterized by the form of their elastic strain energy function. For a more detailed description of elastomeric material, see [Elastomer](#) in Chapter 7.

For the finite element analysis of elastomers, there are some special considerations which do not apply for linear elastic analysis. These considerations, discussed below, include:

- [Large Deformations](#)
- [Incompressible Behavior](#)
- [Instabilities](#)
- [Existence of Multiple Solutions](#)

Large Deformations

The formulation is complete for arbitrarily large displacements and strains.

When extremely large deformations occur, the element mesh should be designed so that it can follow these deformations without complete degeneration of elements. For problems involving extreme distortions, rezoning must be done. Rezoning can be used with the formulation in the updated Lagrangian framework using conventional displacement based elements.

Incompressible Behavior

One of the most frequent causes of problems analyzing elastomers is the incompressible material behavior. Lagrangian multipliers (pressure variables) are used to apply the incompressibility constraint. The result is that the volume is kept constant in a generalized sense, over an element.

Both the total, as well as updated Lagrange formulations, are implemented with appropriate constraint ratios for lower- and higher-order elements in 2-D and 3-D. For many practical analysis, the LBB (Ladyszchenskaya-Babuska-Brezzi) condition does not have to be satisfied in the strictest sense; for example, four node quadrilateral based on Herrmann principle.

For elements that satisfy the LBB condition, error estimates of the following form can be established

$$\|u^h - u\|_1 + \|p^h - p\|_0 = O(h^{\min\{k, \ell+1\}}) \quad (5-38)$$

where k and ℓ are the orders of displacements and pressure interpolations, respectively. If $K = \min\{k, \ell+1\}$, the rate of convergence is said to be optimal, and elements satisfying the LBB condition would not lock.

The large strain elasticity formulation may also be used with conventional plane stress, membrane, and shell elements. Because of the plane stress conditions, the incompressibility constraint can be satisfied without the use of Lagrange multipliers.

Instabilities

Under some circumstances, materials can become unstable. This instability can be real or can be due to the mathematical formulation used in calculation.

Instability can also result from the approximate satisfaction of incompressibility constraints. If the number of Lagrangian multipliers is insufficient, local volume changes can occur. Under some circumstances, these volume changes can be associated with a decrease in total energy. This type of instability usually occurs only if there is a large tensile hydrostatic stress. Similarly, overconstraints give rise to mesh locking and inordinate increase in total energy under large compressive stresses.

Existence of Multiple Solutions

It is possible that more than one stable solution exists (due to nonlinearity) for a given set of boundary conditions. An example of such multiple solutions is a hollow hemisphere with zero prescribed loads. Two equilibrium solutions exist: the undeformed stress-free state and the inverted self-equilibrating state. An example of these solutions is shown in [Figure 5-13](#) and [Figure 5-14](#). If the equilibrium solution remains stable, no problems should occur; however, if the equilibrium becomes unstable at some point in the analysis, problems can occur.

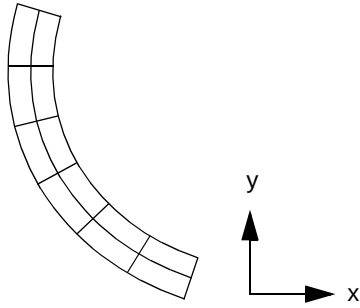


Figure 5-13 Rubber Hemisphere

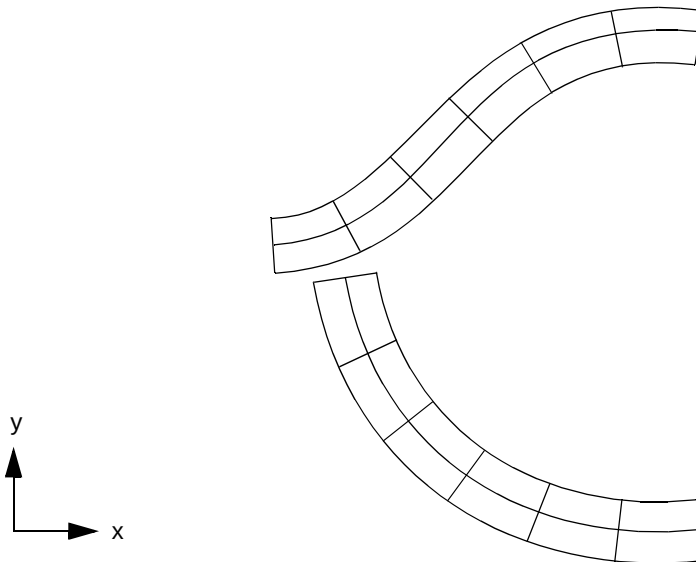


Figure 5-14 Inverted Rubber Hemisphere

When incompressible material is being modeled, the basic linearized incremental procedure is used in conjunction with mixed variational principles similar in form to the Herrmann incompressible elastic formulation. These formulations are incorporated in plane strain, axisymmetric, generalized plane strain, and three-dimensional elements. These mixed elements may be used in combination with other elements in the library (suitable tying may be necessary) and with each other. Where different materials are joined, the pressure variable at the corner nodes must be uncoupled to allow for mean pressure discontinuity. Tying must be used to couple the displacements only.

Large Strain Plasticity

In recent years there has been a tremendous growth in the analysis of metal forming problems by the finite element method. Although an Eulerian flow-type approach has been used for steady-state and transient problems, the updated Lagrangian procedure, pioneered by McMeeking and Rice, is most suitable for analysis of large strain plasticity problems. The main reasons for this are: (a) its ability to trace free boundaries, and (b) the flexibility of taking elasticity and history effects into account. Also, residual stresses can be accurately calculated.

The large strain plasticity capability in MSC.Marc allows you to analyze problems of large-strain, elastic-plastic material behavior. These problems can include manufacturing processes such as forging, upsetting, extension or deep drawing, and/or large deformation of structures that occur during plastic collapse. The analysis involves both material, geometric and boundary nonlinearities.

In addition to the options required for plasticity analysis, the **LARGE DISP**, **UPDATE**, and **FINITE** parameters are needed for large strain plasticity analysis.

In performing finite deformation elastic-plastic analysis, there are some special considerations which do not apply for linear elastic analysis include:

- **Choice of Finite Element Types**
- **Nearly Incompressible Behavior**
- **Treatment of Boundary Conditions**
- **Severe Mesh Distortion**
- **Instabilities**

Choice of Finite Element Types

Accurate calculation of large strain plasticity problems depends on the selection of adequate finite element types. In addition to the usual criteria for selection, two aspects need to be given special consideration: the element types selected need to be insensitive to (strong) distortion; for plane strain, axisymmetric, and three-dimensional problems, the element mesh must be able to represent nondilatational (incompressible) deformation modes.

Nearly Incompressible Behavior

Most finite element types tend to lock during fully plastic (incompressible) material behavior. A remedy is to introduce a modified variational principle which effectively reduces the number of independent dilatational modes (constraints) in the mesh. This procedure is successful for plasticity problems in the conventional “small” strain formulation. Zienkiewicz pointed out the positive effect of reduced integration for this type of problem and demonstrates the similarity between modified variational procedures and reduced integration. MSC.Software Corporation recommends the use

of lower-order elements, invoking the constant dilatation option. The lower-order elements, which use reduced integration and hourglass control, also behave well for nearly incompressible materials.

Treatment of Boundary Conditions

In many large strain plasticity problems, specifically in the analysis of manufacturing processes, the material slides with or without friction over curved surfaces. This results in a severely nonlinear boundary condition. The MSC.Marc gap-friction element and **CONTACT** option can model such sliding boundary conditions.

Severe Mesh Distortion

Because the mesh is attached to the deforming material, severe distortion of the element mesh often occurs, which leads to a degeneration of the results in many problems. The **ERROR ESTIMATE** option can be used to monitor this distortion. To avoid this degeneration, generate a new finite element mesh for the problem and then transfer the current deformation state to the new finite element mesh. The **global adaptive** and **rezoning** procedure in the program is specifically designed for this purpose.

Instabilities

Elastic-plastic structures are often unstable due to necking phenomena. Consider a rod of a rigid-plastic incompressible workhardening material. With $\dot{\epsilon}$ the current true uniaxial strain rate and h the current workhardening, the rate of true uniaxial stress σ is equal to

$$\dot{\sigma} = H\dot{\epsilon} \quad (5-39)$$

The applied force is equal to $F = \sigma A$, where A is the current area of the rod. The rate of the force is therefore equal to

$$\dot{F} = \dot{\sigma}A + \sigma\dot{A} \quad (5-40)$$

On the other hand, conservation of volume requires that

$$A\dot{\epsilon} + \dot{A} = 0 \quad (5-41)$$

Hence, the force rate can be calculated as

$$\dot{F} = (H - \sigma)A\dot{\epsilon} \quad (5-42)$$

Instability clearly occurs if $\sigma > H$. For applied loads (as opposed to applied boundary conditions), the stiffness matrix becomes singular (nonpositive definite).

For the large strain plasticity option, the workhardening slope for plasticity is the rate of true stress versus the true plastic strain rate. The workhardening curve must, therefore, be entered as the true stress versus the logarithmic plastic strain in a uniaxial tension test.

Computational Procedures for Elastic-Plastic Analysis

Three basic procedures for plasticity exist in MSC.Marc. In this section, the variational form of equilibrium equations and constitutive relations, and incompressibility are summarized. Issues regarding return mapping procedures for stress calculation in three-dimensional and plane-stress conditions are also discussed.

For notational purpose, three configurations are considered at any point, original ($t = 0$), previous ($t = n$) and current ($t = n + 1$). An iterative procedure, full or modified Newton-Raphson, secant, or arc-length is used to solve for the equilibrium at $t = n + 1$.

1. Small Strain Plasticity (reference configuration: $t = 0$):

In this approach, the basis of variational formulation are 2nd Piola-Kirchhoff stress, S and Green-Lagrange strain, E . Equilibrium of the current state can thus be represented by the following virtual work principle:

$$\int_{V_n} S : \delta E \, dV = \int_{A_n} t : \delta \eta \, dA + \int_{V_n} b : \delta \eta \, dV \quad (5-43)$$

During the increment, all state variables are defined with respect to the state at $t = n$ and are updated at the end of increment upon convergence.

The linearized form of constitutive equations is given as:

$$dS = L^{ep} : dE - \sigma \cdot dE - dE \cdot \sigma \quad (5-44)$$

in which S is the second Piola-Kirchhoff stress, σ is the Cauchy stress, and L^{ep} is the elasto-plastic moduli. This linearization has the advantage that it is fully independent of the rotation increment, but the disadvantage is that the linearization causes errors equal to the square of the strain increment. Moreover, imposing the incompressibility condition in terms of the trace of Green-Lagrange strains leads to errors in the form of fictitious volume changes in fully developed plastic flow.

The above procedure works well in the context of small strain plasticity. However, in many large deformation problems including metal forming processes, the plastic strain increments can be very large and the above procedure can lead to large errors in the results. The two finite strain plasticity formulations to model the large inelastic strains are: rate based (hypoelastic) and total (hyperelastic).

2. Finite Strain Plasticity with additive decomposition of strain rates (reference configuration: $t = n + 1$).

This formulation is based on the integration of the constitutive equations in the current configuration. To maintain objectivity, the notion of rotation neutralized stress and strain measures is introduced. All objective stress rates, are manifestation of Lie derivative:

$$L_v(\Sigma) = \Phi \left(\frac{d}{dt} (\Phi^{-1} \Sigma \Phi^{-T}) \right) \Phi^T \quad (5-45)$$

where Φ is a deformation measure; for example, deformation gradient or rotation tensor, R while Σ is a stress measure in the current configuration. The general form for an objective stress rate is:

$$\overset{\vee}{\sigma} = \dot{\sigma} - \sigma \Omega^T - \Omega \sigma + \alpha \sigma \text{tr}(d) \quad (5-46)$$

From [Table 5-6](#), it can be seen that there is a possibility of a number of stress rates. It can be observed that while all the above stress rates are objective, the Truesdell and Durban-Baruch rates would not yield symmetric matrices.

Table 5-6 Objective Stress Rates

Stress Rates	Ω	α
Truesdell	L	1
Cotter-Rivlin	L	0
Oldroyd	$-L^T$	0
Jaumann-Zaremba-Noll	W	0
Green-McInnis-Nagdhi	$\dot{R} R^{-1}$	0
Durban Baruch	$\frac{1}{2}(D + W)$	1

MSC.Marc's implementation of rate formulation involves the use of Jaumann rate of Cauchy stress which is obtained as an average of the Oldroyd and Cotter-Rivlin stress rates. Thus, the Jaumann rate can be written as:

$$\overset{\nabla}{\sigma} = \dot{\sigma} - W\sigma + \sigma W \quad (5-47)$$

$$\overset{\nabla}{\sigma} = \dot{\sigma} + \Omega^T \cdot \sigma + \sigma \cdot \Omega + tr(d\epsilon)\sigma \quad (5-48)$$

where $(\dot{})$ is the ordinary rate and $(\overset{\nabla}{})$ is the objective rate with W is the spin or the antisymmetric part of the velocity gradient, L . The last term is neglected because of the incompressible nature of plasticity.

The equilibrium in the current state is given by the virtual work principle at $t = n + 1$:

$$\int_{V_{n+1}} \sigma : \delta\epsilon \, dv = \int_{A_{n+1}} t \cdot \delta\eta \, da + \int_{V_{n+1}} b \cdot \delta\eta \, dv \quad (5-49)$$

Linearization of the above form leads to the variational statement:

$$\int_{V_{n+1}} [d\sigma : \delta\epsilon - 2(d\epsilon \cdot \sigma) : \delta\epsilon + tr(d\epsilon)\sigma : \delta\epsilon + \sigma : \{(\nabla u) \cdot (\nabla \eta)^T\}] \, dv = d \left(\int_{V_{n+1}} b \cdot \delta\eta \, dv + \int_{A_{n+1}} t \cdot \delta\eta \, da \right) - \int_{V_{n+1}} \sigma : \delta\epsilon \, dv \quad (5-50)$$

where,

$$d\sigma = R \cdot d\sigma^{RN} \cdot R^T \quad (5-51)$$

Within the context of rotation neutralized form of constitutive relations,

$d\sigma^{RN}$ is defined as:

$$d\sigma^{RN} = L^{RN} : d\epsilon^{RN} \quad (5-52)$$

where, $d\epsilon^{RN} = R^T \cdot d\epsilon \cdot R$. Also, the rotation neutralized strain can be calculated by:

$$\epsilon^{RN} = (U + I)^{-T} (F + I) \epsilon^{MID} (F + I) (U + I)^{-1} \quad (5-53)$$

with ϵ^{MID} being the mid-increment strain, a good approximation for incremental logarithmic strain measure.

Where $e = U - I$ is the engineering strain and U is the stretch tensor obtained by the polar decomposition of F . Admitting an error of the order of e^2 in the approximation of the logarithmic strain, one obtains:

$$d\sigma = L^{ep} : d\epsilon \quad (5-54)$$

$$\text{where } L^{ep} = R \cdot (R \cdot L^{RN} \cdot R^T) \cdot R^T \quad (5-55)$$

During computations, the third term of Equation 5-50, $tr(d\epsilon) \cdot \sigma : \delta\epsilon$, is neglected due to its nonsymmetric nature.

In a fully developed plastic flow, the volumetric part of the energy can become extremely large and lead to volumetric locking. Hence, a special treatment of incompressibility is done to relax the volumetric constraint in an assumed strain format. The volumetric part of deformation gradient is modified such that, the assumed deformation gradient is:

$$\bar{F} = \bar{J}^{\frac{1}{3}} J^{-\frac{1}{3}} F \quad (5-56)$$

Linearization of $\bar{F}$ in the original state relates it to the displacement gradients in the current state as:

$$D\bar{F} = \bar{\nabla}(\Delta u) \bar{F} \quad (5-57)$$

where,

$$\bar{\nabla}(\Delta u) = (\Delta u)_{i,j} - \frac{1}{3} \text{div}(\Delta u) \delta_{ij} + \frac{1}{3} \overline{\text{div}(\Delta u)} \delta_{ij} \quad (5-58)$$

where the first two terms are evaluated at each integration point and the last term is averaged over the element.

For a lower order element, the procedure leads to mean or constant dilatation approach. Considering Equation 5-50 to Equation 5-58 the resulting system can be expressed as:

$$\begin{aligned} & \int_{V_{n+1}} [\bar{\nabla}^s \eta : L^{ep} : \bar{\nabla}^s(\Delta u) - 2\bar{\nabla}^s \eta \cdot \sigma : \bar{\nabla}^s(\Delta u) + \sigma : \bar{\nabla}(\Delta u) \cdot (\bar{\nabla} \eta)^T] dv = \\ & d \left(\int_{V_{n+1}} b \cdot \delta \eta dv + \int_{A_{n+1}} t \cdot \delta \eta da \right) - \int_{V_{n+1}} [\bar{\nabla}^s \eta]^T : \sigma dv \end{aligned} \quad (5-59)$$

In the event, the elastic strains become large in an elastic-plastic analysis the rate based constitutive equations do not accurately model the material response. This results from the fact that the elasticity matrix are assumed to have constant coefficients in the current deformed configuration.

In the next formulation, the rate of deformation tensor is decomposed multiplicatively into the elastic and plastic parts to resolve these problems.

3. Finite strain plasticity with multiplicative decomposition of deformation gradient.

An alternative formulation, based on the multiplicative decomposition of the deformation gradient has been implemented in MSC.Marc, namely:

$$F = F^e F^\theta F^p \quad (5-60)$$

where F^e , F^θ , and F^p are (elastic, thermal, and plastic) deformation gradients, respectively. The thermo-mechanical coupling is implemented using the staggered approach.

The above decomposition has a physical basis to it as the stresses are derived from quadratic - logarithmic strain energy density function:

$$W = \frac{1}{2} \lambda (\ln J^e)^2 + \mu \operatorname{tr} \left(\frac{1}{2} \ln b^e \right)^2 \quad (5-61)$$

This function has been chosen due to the availability of the material coefficients from material testing in a small strain case.

In the metal forming applications, the elastic strains are negligible and the rate-based (or hypoelastic) as well as the total (or hyperelastic) form of constitutive equations give virtually the same results. However, many polymers, metals subjected to large hydrostatic pressures, high velocity impact loading of metals and processes, where shape changes after deformation need to be evaluated precisely, the hyperelastic formulation yields physically more meaningful results.

The return mapping procedure for the calculation of stresses is based on the radial return procedure. With the use of exponential mapping algorithm, the incompressibility condition is imposed exactly:

$$\det F^p = 1 \quad (5-62)$$

The strain energy is separated into deviatoric and volumetric parts in the framework of mixed formulation. The general form of three-field variational principle is:

$$\pi(u, p, J) = \int_{V_0} [W(\bar{b}) + U(\bar{J}^e) + p(J^e - \bar{J}^e)] dV \quad (5-63)$$

$$\text{where, } \bar{b} = J^{-\frac{2}{3}} b \quad (5-64)$$

$W(\bar{b})$ and $U(\bar{J}^e)$ are the deviatoric and elastic volumetric parts of the free energy, p is the pressure, J^e and $\bar{J}^e$ are elastic pointwise and average elastic Jacobian of the element, respectively. The volumetric free energy can be of any form in [Equation 5-62](#). However, if the incompressibility is enforced in a pointwise fashion and the volumetric free energy is assumed to be of the form:

$$U(J^e) = \frac{9}{2} K \left[(J^e)^{\frac{1}{3}} - 1 \right]^2 \quad (5-65)$$

the perturbed Lagrangian form of the variational principle can be cast in a two-field framework as:

$$\pi(u, P) = \int_{V_0} \left[W(\bar{b}) + 3P \left\{ (J^e)^{\frac{1}{3}} - 1 \right\} - \frac{P^2}{2K} \right] dV \quad (5-66)$$

Note that $P = p(J^e)^{\frac{2}{3}}$, hence, P is not the true pressure in [Equation 5-66](#). Choice of the volumetric free energy in [Equation 5-65](#) is not arbitrary and is described in more detail in Chapter 7 in the [Elastomer](#) section. Linearization of the equilibrium condition arising from the stationary of the variational principle [Equation 5-66](#), yields:

$$\begin{aligned}
& \int_{V_{n+1}} \left[\nabla^s \eta : \left\{ \frac{1}{J} \left(\mathbb{C}_{dev} + P(J^e)^{\frac{1}{3}} \left(\frac{1}{3} \mathbf{1} \otimes \mathbf{1} - 2I \right) \right) \right\} : \nabla^s u + \right. \\
& \quad \left. \sigma : \{ (\nabla(\Delta u)) \cdot (\nabla \eta)^T \} + \frac{dP}{J} (J^e)^{\frac{1}{3}} \mathbf{1} : \nabla \eta \right] dv = \\
& \quad d \left(\int_{V_{n+1}} b \cdot \delta \eta \, dv + \int_{A_{n+1}} t \cdot \delta \eta \, da \right) - \int_{V_{n+1}} (\nabla^s \eta)^T : \sigma \, dv
\end{aligned} \tag{5-67}$$

The linearization of the constraint equation defines the term dP in [Equation 5-67](#).

The derivation of the spatial tangent $\mathbb{C}_{dev}$ follows by a push forward of the material tangent into the current configuration:

$$\text{Hence, } S = F^{-1} \tau^{dev} F^{-T} \tag{5-68}$$

where S is the pull back of the deviatoric Kirchhoff stress tensor.

Assuming the entire incremental deformation to be elastic, a symmetric stress tensor, $\hat{S}$ can be defined with respect to a fixed plastic intermediate configuration F_n^p as:

$$\hat{S} = (F_e^{tr})^{-1} \tau^{dev} (F_e^{tr})^{-T} \tag{5-69}$$

also noting that,

$$\hat{S} = \sum_{A=1}^3 S_A N_A \otimes N_A \text{ where } \hat{S}_A = \frac{\beta_A}{(\lambda_{Ae}^{tr})^2} \tag{5-70}$$

$$d\hat{S} = \sum_{A=1}^3 d\hat{S}_A N_A \otimes N_A + \sum_{A=1}^3 \sum_{B \neq A}^3 (\hat{S}_B - \hat{S}_A) \Omega_{BA} N_A \otimes N_B \tag{5-71}$$

$$\text{where } d\hat{S}_A = \frac{1}{(\lambda_{Ae}^{tr})^2 (\lambda_{Be}^{tr})^2} \left(\frac{\partial \beta_A}{\partial \epsilon_{Be}^{tr}} - 2\beta_A \delta_{AB} \right) \lambda_{Be}^{tr} d\lambda_{Be}^{tr} \tag{5-72}$$

$$\text{and } dN_A = \Omega_{AB} N_B \tag{5-73}$$

$$\text{similarly, } \hat{C} = (F_e^{tr})^T F_e^{tr} \text{ with } F_e^{tr} = \sum_{A=1}^3 \lambda_{Ae}^{tr} n_A \otimes N_A \quad (5-74)$$

$$\begin{aligned} d\hat{C} = & \sum_{B=1}^3 2\lambda_{Be}^{tr} d\lambda_{Be}^{tr} N_B \otimes N_B + \\ & \sum_{A=1}^3 \sum_{B \neq A}^3 \left\{ (\lambda_{Be}^{tr})^2 - (\lambda_{Be}^{tr})^2 \right\} \Omega_{BA} N_A \otimes N_B \end{aligned} \quad (5-75)$$

The algorithmic elasto-plastic tangent can be obtained as:

$$\mathbb{C}_{dev} = 2F \left(F \frac{\partial S}{\partial C} F^T \right) F^T = 2(F_e^{tr})(F_e^{tr}) \frac{\partial \hat{S}}{\partial \hat{C}} (F_e^{tr})^T (F_e^{tr})^T \quad (5-76)$$

Combining Equation 5-71, Equation 5-76, and Equation 5-77, the spatial form of deviatoric part of the tangent as:

$$\begin{aligned} \mathbb{C}_{dev} = & \sum_{A=1}^3 \sum_{B \neq A}^3 \left(\frac{\partial \beta_A}{\partial \epsilon_{Be}^{tr}} - 2\beta_A \delta_{AB} \right) n_A \otimes n_A \otimes n_B \otimes n_B \\ & + \sum_{A=1}^3 \sum_{B \neq A}^3 \left[\frac{(\lambda_{Ae}^{tr})^2 \beta_B - (\lambda_{Be}^{tr})^2 \beta_A}{(\lambda_{Be}^{tr})^2 - (\lambda_{Ae}^{tr})^2} \right] (n_A \otimes n_B \otimes n_A \otimes n_B + \\ & n_A \otimes n_B \otimes n_B \otimes n_A) \end{aligned} \quad (5-77)$$

The singularity for two- or three-equal stretch ratios is removed by repeated application of L'Hospital's rule. Due to separate interpolation of the pressure, a larger system of equations needs to be solved. A static condensation of the element variables is carried out before going in the solver. This step renders the singularity of the system to be inversely proportional to the bulk modulus of the material. Due to the multiplicative split of the deformation gradient and an additive split of the free energy into deviatoric and volumetric parts, this framework is eminently suitable for implementation of general nonlinear elastic as well as inelastic material models. Besides, compressible and nearly incompressible material response can be naturally handled.

Finally, the stress calculation is done via return mapping to the yield surface. There are several methods to return the stress. Among them, the popular ones include:

Closest point projection (which reduces to radial return for von Mises), mean normal approach, midpoint rule, trapezoidal rule, tangent cutting plane algorithm, and multistage method.

Krieg and Krieg (1977) investigated the radial return algorithm which has been proposed by Wilkins (1964). The third algorithm analyzed was the so-called secant stiffness method proposed by Rice and Tracy (1973) which represented a special case of the generalized Trapezoidal rule for nonhardening materials algorithm in contrast to the fully implicit radial return algorithm and the fully explicit tangent stiffness approach. The radial return method (shown in [Figure 5-15](#)) was found to be the most accurate among the three analyzed, particularly when large strain increments were used. The radial return and mean normal method are available in MSC.Marc and are described next.

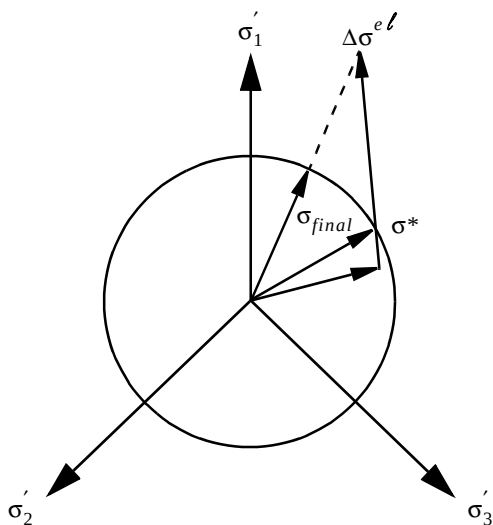


Figure 5-15 Radial Return Method

1. Closest Point Projection procedure:

The closest point projection or the backward Euler algorithm reduces to a radial return scheme for cases where there is no anisotropy or plane stress condition. The trial stress is determined as:

$$\sigma_d^{rr} = \sigma_d + 2G\Delta\epsilon_d \quad (5-78)$$

The final stress state is obtained by simply scaling the trial stress by a scale factor, β

$$\text{where, } \beta = 1 - \frac{2G\Delta\lambda}{\|\sigma_d^{rr}\|} \quad (5-79)$$

the plastic consistency parameter $\Delta\lambda$, is obtained by solving for consistency condition iteratively.

An explicit form of the material tangent for the isotropic, von Mises yield surface can be given as:

$$L^{ep} = 2G\beta\left(I - \frac{1}{3}1 \otimes 1\right) - 2G\gamma\hat{N} \otimes \hat{N} \quad (5-80)$$

$$\text{where, } \gamma = \frac{1}{1 + \frac{H}{3G}} - 1 + \beta \quad (5-81)$$

Here $\hat{N}$ is the return direction normal to the current yield surface. For the plane stress case, the zero normal stress condition is explicitly imposed in the yield condition. Since the yield surface is not a circle in the plane, the return direction is not radial anymore.

2. Mean-Normal method:

Assuming the entire deformation to be elastic in nature, the trial stress is evaluated as:

$$\sigma_d^{mn} = \sigma_d + 2\alpha G\Delta\epsilon_d + (1 - \alpha)G\Delta\epsilon_d \quad (5-82)$$

Due to the use of pressure independent yield function, only the deviatoric stresses are evaluated. As shown in [Figure 5-16](#), $(1 - \alpha)$ represents the fraction of strain increment for which the plastic flow occurs.

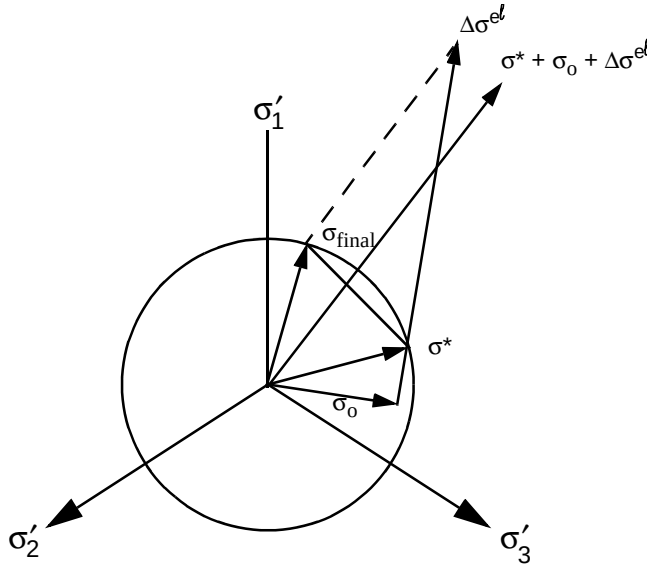


Figure 5-16 Mean-Normal Method

σ_d includes the deviatoric stresses at the previous increment scaled such that it satisfies the yield condition. The yield criterion might not be exactly satisfied due to temperature effects, numerical integration of elastic-plastic relationships, or accumulated numerical inaccuracy.

$F(\sigma) = 0$ indicates that the stress state is exactly on the yield surface. If $F(\sigma) > 0$, scale σ by a factor λ such that:

$$F(\lambda\sigma) = 0 \quad 0 < \lambda < 1 \quad (5-83)$$

The equivalent plastic strain, plastic strain tensor and the elasto-plastic moduli are obtained as:

$$\Delta \bar{\epsilon}^p = \frac{(\mathbb{C}^e : \hat{N}) : \Delta \hat{\epsilon} (1 - \alpha)}{\hat{N} : \mathbb{C}^e : \hat{N} + H} \quad (5-84)$$

$$\Delta \epsilon^p = \Delta \bar{\epsilon}^p \hat{N} \quad (5-85)$$

$$L^{ep} = \mathbb{C}^e - (1 - \alpha) \frac{(\mathbb{C}^e : \hat{N}) \otimes (\mathbb{C}^e : \hat{N})}{\hat{N} : \mathbb{C}^e : \hat{N} + H} \quad (5-86)$$

where, $\hat{N}$ is the mean-normal direction to the yield surface, H is the hardening modulus and $\hat{C}^e$ is the elastic tangent moduli. For nonhardening materials, the mean normal algorithm is a special case of trapezoidal rule. MSC.Marc plasticity algorithms are unconditionally stable and accurate for moderate strain increments. However, overall global stability of the system can be dictated by other considerations like contact, buckling which then guide the selection of appropriate load increment size. It should also be recognized that the algorithmically consistent tangent in the closest point projection algorithm is based on the current state, and a lack of embedded directionality (unlike secant methods) in the tangent can lead to divergent solutions unless line search or automatic time stepping algorithms are used. However, when it converges it shows quadratic convergence as compared to linear convergence for the mean-normal scheme.

3. Multistage Return Mapping:

The return mapping scheme described in this section is used for isotropic and anisotropic plasticity when **PLASTICITY**, 4 is used to describe the Updated Lagrange Procedure. Also, regardless of which **PLASTICITY** parameter is used to describe the Updated Lagrange procedure, MSC.Marc automatically switches to this scheme for the Hill and Barlat yield criteria. This scheme is not available for pressure dependent yield criteria like Mohr Coulomb. In a large strain analysis, MSC.Marc automatically chooses the return mapping scheme that is most suited for the given material.

The increment of Cauchy stress for elasto-plasticity becomes

$$\Delta \hat{\sigma} = \hat{C}^e : \Delta \hat{\epsilon}^e = \hat{C}^e : (\Delta \hat{\epsilon} - \Delta \hat{\epsilon}^p). \quad (5-87)$$

In Equation 5-87, superscript ' $\hat{}$ ' means a materially embedded co-rotational coordinate system. For the numerical implementation of Equation 5-87, the general midpoint rule can be expressed as follows:

$$\hat{\sigma}_{n+1} = \hat{\sigma}_{n+1}^T - \lambda \hat{C}^e \hat{N}_{n+\alpha} \quad (5-88)$$

$$\text{where } \hat{\sigma}_{n+1}^T = \hat{\sigma}_n + \hat{C}^e \Delta \hat{\epsilon}$$

$$(\text{Case 1}) \text{ if } F(\hat{\sigma}_{n+1}^T) \leq 0, \lambda = 0, \hat{\sigma}_{n+1} = \hat{\sigma}_{n+1}^T$$

$$(\text{Case 2}) \text{ if } F(\hat{\sigma}_{n+1}^T) > 0, \lambda = \bar{\lambda} \text{ such that } F(\hat{\sigma}_{n+1}) = 0$$

where superscript T stands for a trial state. For Case 2, the condition stating that the updated stress stays on the strain-hardening curve ($\bar{\sigma} = \rho(\bar{\epsilon}^p)$) provides the following condition:

$$F(\lambda) = \bar{\sigma}(\hat{\sigma}_{n+1}^T - \lambda \hat{C}^e \hat{N}_{n+\alpha}) - \rho(\bar{\epsilon}_n^p + \lambda) = 0. \quad (5-89)$$

Equation 5-89 is a nonlinear equation to solve for λ .

If the strain increment is not small enough, it is difficult to obtain the numerical solution of Equation 5-89 even though it has a mathematical solution. Therefore, an iterative scheme, which utilizes the control of the potential residual, is introduced. The method is applicable to a nonquadratic yield function and a general hardening law. For the implementation of the algorithm, Equation 5-89 has been modified to the following relationship with $\alpha = 1$; that is,

$$F(\lambda_{(k)}) = \bar{\sigma}(\hat{\sigma}^T - \lambda_{(k)} \hat{C}^e \hat{N}_{(k)}) - \rho(\bar{\epsilon}_n^p + \lambda_{(k)}) = F_{(k)} \quad (5-90)$$

where

$$F(\lambda_{(0)} = 0) = F_{(0)}, \{F_{(k)} | F_{(0)} > F_{(1)} > F_{(k)} > \dots > F_{(N)} (F_{(N)} = 0), k = 0 \sim N\}, \\ \Delta F = (F_{k-1} - F_k) < \sigma_Y (= (\bar{\sigma}(\bar{\epsilon}^p = 0))) \text{ and } F_{k=1 \sim N-1} \text{ are prescribed values.}$$

As shown in Equation 5-90 and Figure 5-17, the direction of the first substep $\hat{N}_{(1)}$ is guessed from the direction $\hat{N}_{(0)}$, which is normal to the yield surface at the trial stress $\hat{\sigma}^T$. Then, the exact direction $\hat{N}_{(1)}$ can be obtained from the first substep nonlinear solution based on Euler backward method. In general, the new direction for the second substep $\hat{N}_{(i+1)}$ is guessed from the direction $\hat{N}_{(i)}$ based on the previous substep stress $\hat{\sigma}_{(i)}$. This procedure is completed when $F = F_{(N)} (= 0)$. In fact, the equation for the N^{th} step is equivalent to Equation 5-89 and $\hat{\sigma}_{n+1} = \hat{\sigma}_{(N)}$. Finally, the proportional logarithmic plastic strain remains normal to the yield surface at the final stress $\hat{\sigma}_{n+1}$; that is,

$$\Delta \hat{\epsilon}^p = \Delta \bar{\epsilon}^p \frac{\partial \bar{\sigma}}{\partial \hat{\sigma}}(\hat{\sigma}_{n+1}) (\equiv \lambda_{(N)} \hat{N}_{(N)}). \quad (5-91)$$

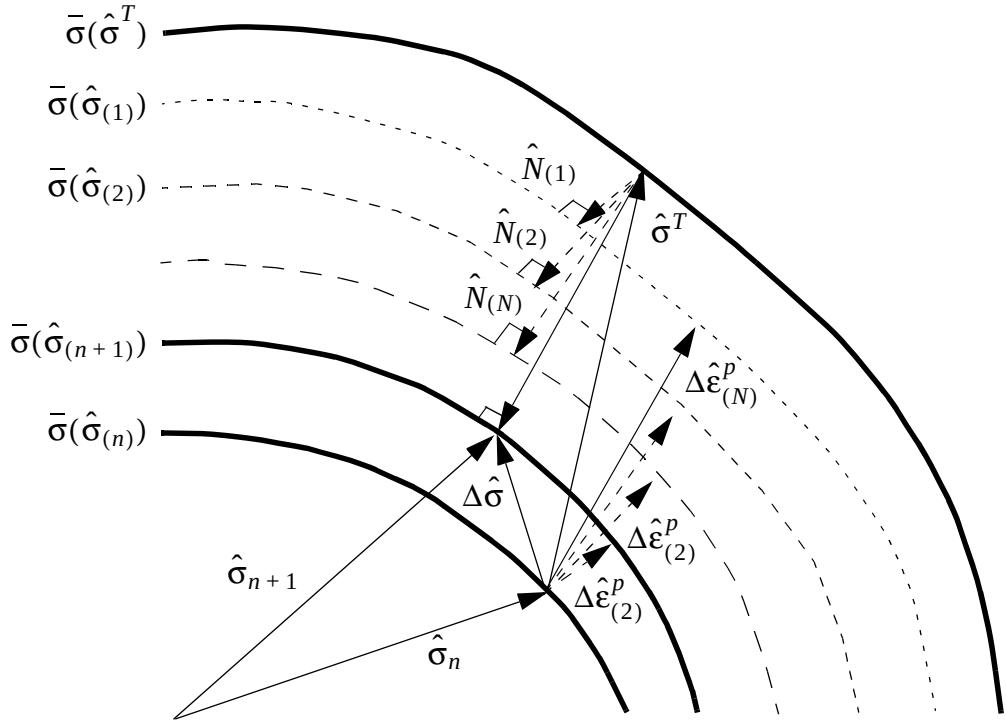


Figure 5-17 Schematic View for Multistage Return Mapping Method

Therefore, the normality condition of the incremental deformation theory is satisfied at the current state $(n + 1)$ for $\alpha = 1$.

In order to solve Equation 5-90, the Euler Backward method is employed to solve the iteration procedure for the k^{th} substep. At each iteration, $\Delta \lambda_{(k)}^{(i)}$ (at (k) substep and (i) iteration) becomes

$$\Delta \lambda_{(k)}^{(i)} = \frac{g_1(\lambda_{(k)}^{(i)}) - \hat{N}_{(k)}^{(i)} E_{(k)}^{(i)-1} g_2(\lambda_{(k)}^{(i)}) + g_3(\lambda_{(k)}^{(i)}) H}{\hat{N}_{(k)}^{(i)} E_{(k)}^{(i)-1} \hat{N}_{(k)}^{(i)} + H} \quad (5-92)$$

where

$$E_{(k)}^{(i)-1} = \left[\hat{C}^{e-1} + \lambda_{(k)}^{(i)} \frac{\partial \hat{N}}{\partial \hat{\sigma}} \right]^{-1},$$

$$g_1\left(\lambda_{(k)}^{(i)}\right) = \bar{\sigma}(\hat{\sigma}) - \rho\left(\bar{\varepsilon}_h^p + \lambda_{(k)}^{(i)}\right) - F_{(k)},$$

$$g_2\left(\lambda_{(k)}^{(i)}\right) = \hat{C}^{e-1}(\hat{\sigma} - \hat{\sigma}^T) + \lambda_{(k)}^{(i)} \hat{N},$$

$$g_3\left(\lambda_{(k)}^{(i)}\right) = H^{-1}(\rho - \rho_n) - \lambda_{(k)}^{(i)},$$

and H is the hardening modulus in stress-strain curve.

The detailed derivations of Equation 5-92 are shown in the work of Yoon et al. [Ref. 33].

In order to solve the equilibrium equation iteratively, the elasto-plastic tangent modulus consistent with the current return mapping method is obtained as follows:

$$d\hat{\sigma}^j = d\hat{\sigma}_{n+1} = L^{ep} d\hat{\varepsilon}_{n+1} \quad (5-93)$$

$$\text{where } L^{ep} = \left(\bar{C} - \frac{\bar{C} \hat{N}_{n+\alpha} \otimes \bar{C} \hat{N}_{n+\alpha}}{\hat{N}_{n+\alpha} \bar{C} \hat{N}_{n+\alpha} + h'} \right) \text{ and } \bar{C} = \left[\hat{C}^{e-1} + \lambda \frac{\partial \hat{N}_{n+\alpha}}{\partial \hat{\sigma}_{n+1}} \right]^{-1}.$$

In Equation 5-93, h' is instantaneous slope and $\alpha = 1$ is used.

CREEP

Creep is a time-dependent inelastic behavior that can occur at any stress level, either below or above the yield stress of a material. Creep is an important factor at elevated temperatures. In many cases, creep is also accompanied by plasticity, which occurs above the yield stress of the material.

MSC.Marc offers two schemes for modeling creep in conjunction with plasticity:

- a. treating creep strains and plastic strains separately using an explicit procedure (where the creep is treated explicitly) or an implicit procedure (where both creep and plasticity are treated implicitly). These procedures are available with standard options via data input or with user-specified options via user subroutines. More details are provided below.
- b. modeling creep strains and plastic strains in a unified fashion (viscoplasticity). Both explicit and implicit procedures are again available for modeling unified viscoplasticity. More details are provided in the section titled [Viscoplasticity](#) in this chapter.

The options offered by MSC.Marc for modeling creep are as follows:

- Creep data can be entered directly through data input or user subroutine. For explicit creep, the user subroutine to be used is **CRPLAW** and for implicit creep, the user subroutine to be used is **UCRPLW**.
- An automatic time stepping scheme can be used to maximize the time step size in the analysis.
- Eigenvalues can be extracted for the estimation of creep buckling time.

In addition, for explicit creep, the following additional options can be used:

- Creep behavior can be either isotropic or anisotropic.
- The Oak Ridge National Laboratory (ORNL) rules on creep can be activated.

The creep analysis option is activated in MSC.Marc through the **CREEP** parameter. The creep time period and control tolerance information are input through the history definition option **AUTO CREEP**. This option can be used repeatedly to define a new creep time period and new tolerances. These tolerances are defined in the section on Creep Control Tolerances. Alternatively, a fixed time step can be specified through the **CREEP INCREMENT** history definition option. In this case, no additional tolerances are checked for controlling the time step.

Creep analysis is often carried out in several runs using the **RESTART** option. Save restart files for continued analysis. The **REAUTO** option allows you to reset the parameters defined in the **AUTO CREEP** option upon restart.

Adaptive Time Control

The **AUTO CREEP** option takes advantage of the diffusive characteristics of most creep solutions. Specifically, this option controls the transient creep analysis. You specify a period of creep time and a suggested time increment. The program automatically selects the largest possible time increment that is consistent with the tolerance set on stress and strain increments (see **Creep Control Tolerances** in this chapter).

The algorithm is: for a given time step Δt , a solution is obtained. MSC.Marc then finds the largest values of stress change per stress, $\Delta\sigma/\sigma$, and creep strain change per elastic strain, $\Delta\epsilon^{cr}/\epsilon^{el}$. It compares these values to the tolerance values, T_s (stress change tolerance) and T_e (strain change tolerance), for this period.

The value p is calculated as the larger of

$$\begin{aligned} &(\Delta\sigma/\sigma)/T_s \\ &\text{or} \\ &(\Delta\epsilon^{cr}/\epsilon^{el})/T_e \end{aligned} \tag{5-94}$$

If $p > 1$, the program resets the time step as

$$\Delta t_{new} = 0.8 \Delta t_{old} / p \quad (5-95)$$

The time increment is repeated until convergence is obtained or the maximum recycles control is exceeded. In the latter case, the run is ended.

If the first repeat does not satisfy tolerances, the possible causes are:

- excessive residual load correction
- strong additional nonlinearities such as creep buckling-creep collapse
- incorrect coding in user subroutine [CRPLAW](#), [VSWELL](#), or [UVSCPL](#).

Appropriate action should be taken before the solution is restarted.

If $p < 1$, the solution is stepped forward to $t + \Delta t$ and the next step is begun. The time step used in the next increment is chosen as

$$\Delta t_{new} = \Delta t_{old} \quad \text{if } 0.8 \leq p < 1 \quad (5-96)$$

$$\Delta t_{new} = 1.25 \Delta t_{old} \quad \text{if } 0.65 \leq p < 0.8 \quad (5-97)$$

$$\Delta t_{new} = 1.5 \Delta t_{old} \quad \text{if } p < 0.65 \quad (5-98)$$

Since the time increment is adjusted to satisfy the tolerances, it is impossible to predetermine the total number of time increments for a given total creep time.

Creep Control Tolerances

MSC.Marc performs a creep analysis under constant load or displacement conditions on the basis of a set of tolerances and controls you provide. These are as follows:

1. *Stress change tolerance* – This tolerance controls the allowable stress change per time step during the creep solution, as a fraction of the total stress at a point. Stress change tolerance governs the accuracy of the transient creep response. If you need accurate tracking of the transient response, specify a tight tolerance of 1 percent or 2 percent stress change per time step. If you need only the steady-state solution, supply a relatively loose tolerance of 10-20 percent. It is also possible to check the absolute rather than the relative stress.
2. *Creep strain increment per elastic strain* – MSC.Marc uses either explicit or implicit integration of the creep rate equation. When the explicit procedure is used, the creep strain increment per elastic strain is used to

control stability. In almost all cases, the default of 50 percent represents the stability limit, so that you need not provide any entry for this value. It is also possible to check the absolute rather than the relative strain.

3. *Maximum number of recycles for satisfaction of tolerances* – During **AUTO CREEP**, MSC.Marc chooses its own time step. In some cases, the program recycles to choose a time step that satisfies tolerances, but recycling rarely occurs more than once per step. Excessive recycling can be caused by physical problems such as creep buckling, poor coding of user subroutine **CRPLAW**, **VSWELL**, or **UVSCPL** or excessive residual load correction that can occur when the creep solution begins from a state that is not in equilibrium.

The maximum number of recycles allows you to avoid wasting machine time under such circumstances. If there is no satisfaction of tolerances after the attempts at stepping forward, the program stops. The default of five recycles is conservative in most cases.

4. *Low stress cut-off* – Low stress cut-off avoids excessive iteration and small time steps caused by tolerance checks that are based on small (round off) stress states. A simple example is a beam in pure bending. The stress on the neutral axis is a very small roundoff-number, so that automatic time stepping scheme should not base time step choices on tolerance satisfaction at such points. The default of five percent of the maximum stress in the structure is satisfactory for most cases.
5. *Choice of element for tolerance checking* – Creep tolerance checking occurs as a default for all integration points in all elements. You might wish to check tolerances in only 1 element or in up to 14 elements of your choice. Usually, the most highly stressed element is chosen.

When you enter the tolerances and controls, the following conventions apply:

- All stress and strain measures in tolerance checks are second invariants of the deviatoric state (that is, equivalent von Mises uniaxial values).
- You can reset all tolerances and controls upon the completion of one **AUTO CREEP** sequence.

Background Information

Creep behavior is based on a von Mises creep potential with isotropic behavior described by the equivalent creep law:

$$\dot{\bar{\epsilon}}^{cr} = f(\bar{\sigma}, \bar{\epsilon}^{cr}, T, t) \quad (5-99)$$

The material behavior is therefore described by:

$$\Delta \epsilon^{cr} = \dot{\epsilon}^{cr} \frac{\partial \bar{\sigma}}{\partial \sigma'} \quad (5-100)$$

where $\frac{\partial \bar{\sigma}}{\partial \sigma'}$ is the outward normal to the current von Mises stress surface and $\dot{\epsilon}^{cr}$ is the equivalent creep strain rate.

There are two numerical procedures used in implementing creep behavior. The default is an explicit procedure in which the above relationship is implemented in the program by an initial strain technique. In other words, a pseudo-load vector due to the creep strain increment is added to the right-hand side of the stiffness equation.

$$K \Delta u = \Delta P + \int_V \beta^T D \Delta \epsilon^{cr} dv \quad (5-101)$$

where K is the stiffness matrix, and Δu and ΔP are incremental displacement and incremental nodal force vectors, respectively. The integral

$$\int_V \beta^T D \Delta \epsilon^{cr} dv \quad (5-102)$$

is the pseudo-load vector due to the creep strain increment in which β is the strain displacement relation and D is the stress-strain relation. When plasticity is also specified through a suitably defined yield criterion and yield stress in MSC.Marc, the plasticity is treated implicitly while the creep is treated explicitly.

As an alternative, an implicit creep procedure can be requested with the **CREEP** parameter. In this case, the inelastic strain rate has an influence on the stiffness matrix. Using this technique, significantly larger steps in strain space can be used. In MSC.Marc, this option is only to be used for isotropic materials with the creep strain rate defined by

$$\dot{\epsilon}^{cr} = A \bar{\sigma}^n f(\bar{\epsilon}^{cr}) g(T) h(t)$$

where A is the creep constant that can be defined through input data or through user subroutine **UCRPLW**; power law expression is always to be used for the effective stress with the coefficient provided through the input data or user subroutine

UCRPLW; and power law coefficients or more general expressions can be provided for the creep strain, temperature, and time through the input data or user subroutine **UCRPLW**, respectively.

When plasticity is also specified through a suitably defined yield stress in conjunction with the von Mises yield criterion, a sub-iterative scheme within each Newton-Raphson cycle is used to determine the plastic strains needed to keep the stress state on the yield surface and the creep strains that develop due to the equivalent stress being greater than a user-defined back stress. The yield stress for the plastic component can be varied as a function of the equivalent plastic strain, temperature and spatial coordinates. Similarly, the back stress for the creep component can be varied as a function of the equivalent creep strain, temperature and spatial coordinates.

Creep Buckling

MSC.Marc also predicts the creep time to buckling due to stress redistribution under given load or repeated cyclic load.

The buckling option solves the following equation for the eigenvalue

$$(K + \lambda K_G)\Phi = 0 \quad (5-103)$$

The geometric stiffness matrix, K_G , is a function of the increments of stress and displacement. These increments are calculated during the last creep time step increment. To determine the creep time to buckle, perform a buckle step after a converged creep increment. Note that the incremental time must be scaled by the calculated eigenvalue, and added to the total (current) time to get an estimate as to when buckling occurs.

AUTO THERM CREEP (Automatic Thermally Loaded Elastic-Creep/ Elastic-Plastic-Creep Stress Analysis)

The **AUTO THERM CREEP** option is intended to allow automatic, thermally loaded elastic-creep/elastic-plastic-creep stress analysis, based on a set of temperatures defined throughout the mesh as a function of time. The temperatures and transient times are presented to the program through the **CHANGE STATE** option, using input option 3 (post file), and the program creates its own set of temperature steps (increments) based on a temperature change tolerance provided in this option. The times at all temperature steps are calculated by the program for creep analyses.

At each temperature step (increment), an elastic/elastic-plastic analysis is carried out first to establish stress levels in the structure. A creep analysis is performed next on the structure for the time period between current and previous temperature steps (increments). Both the elastic/elastic-plastic stress and the creep analyses are repeated

until the total creep time provided in this option is reached. Convergence controls are provided on the [CONTROL](#) option for elastic-plastic analysis and in the [AUTO THERM CREEP](#) option for creep analysis. The analysis can be restarted at temperature steps (increments) or at creep steps (subincrements). The results can be saved on a post file ([POST](#) option) for postprocessing.

If no [DIST LOADS](#), [POINT LOAD](#), or [PROPORTIONAL INCREMENT](#) option appears with the [AUTO THERM CREEP](#) set, all mechanical loads and kinematic boundary conditions are held constant during the [AUTO THERM CREEP](#). However, [DIST LOADS](#), [POINT LOAD](#), [PROPORTIONAL INCREMENT](#), or [DISP CHANGE](#) can be included in the set – the mechanical loads and kinematic boundary conditions which are then defined are assumed to change in proportion to the time scale of the temperature history defined by the [CHANGE STATE](#) option and are applied accordingly. This is based on the fact that the increments of load and displacement correspond to the end of the transient time of the [AUTO THERM CREEP](#) input.

Viscoelasticity

In a certain class of problems, structural materials exhibit viscoelastic behavior. Two examples of these problems are quenching of glass structures and time-dependent deformation of polymeric materials. The viscoelastic material retains linearity between load and deformation; however, this linear relationship depends on time. Consequently, the current state of deformation must be determined from the entire history of loading. Different models consisting of elastic elements (spring) and viscous elements (dashpot) can be used to simulate the viscoelastic material behavior described in [Chapter 7](#). A special class of temperature dependence known as the Thermo-Rheologically Simple behavior (TRS) is also applicable to a variety of thermal viscoelastic problems. Both the equation of state and the hereditary integral approaches can be used for viscoelastic analysis.

To model the thermo-rheologically simple material behavior, the [SHIFT FUNCTION](#) model definition option can be used to choose the Williams-Landel-Ferry equation or the power series expression or Narayanaswamy model.

In MSC.Marc, two options are available for small strain viscoelastic analysis. The first option uses the equation of state approach and represents a Kelvin model. The second option is based on the hereditary integral approach and allows the selection of a generalized Maxwell model. The thermo-rheologically simple behavior is also available in the second option for thermal viscoelastic analysis. The [Time-independent Inelastic Behavior](#) section in [Chapter 7](#) discusses these models in detail. Automatic time stepping schemes [AUTO CREEP](#) and [AUTO STEP](#) can be used in a viscoelastic analysis for first and second options, respectively.

The first option for viscoelastic analysis uses the Kelvin model. To activate the generalized Kelvin model in MSC.Marc, use the **VISCO ELAS** or **CREEP** parameter. To input the matrices [A] and [B] for the Kelvin strain rate computations, use the user subroutine **CRPVIS**. To input creep time period and the tolerance control for the maximum strain in an increment, use the history definition option **AUTO CREEP**.

Simo model for large strain viscoelasticity can be used in conjunction with the damage and hyperelastic Mooney or Ogden material model. The large strain viscoelastic material behavior can be simulated by incorporating the model definition option **VISCELMOON** or **VISCELOGDEN**. Viscoelasticity for hyperelastic materials is available only in the total Lagrangian framework.

Nonlinear structural relaxation behavior of materials can be modeled by the Narayanaswamy model which accounts for memory effect. This model allows simulation of evolution of physical properties of glass subjected to complex time temperature histories. The thermal expansion behavior for the Narayanaswamy model is controlled via the model definition option **VISCEL EXP**.

Viscoplasticity

There are two procedures in MSC.Marc for viscoplastic analysis: explicit and implicit. A brief description of each procedure follows:

Explicit Method

The elasto-viscoplasticity model in MSC.Marc is a modified creep model to which a plastic element is added. The plastic element is inactive when the stress is less than the yield stress of the material. You can use the elasto-viscoplasticity model to solve time-dependent plasticity and creep as well as plasticity problems with a nonassociated flow law.

The **CREEP** option in MSC.Marc has been modified to enable solving problems with viscoplasticity. The method is modified to allow solving elastic-plastic problems with nonassociated flow rules which result in nonsymmetric stress-strain relations if the tangent modulus method is used.

The requirements for solving the viscoplastic problem are:

- **CREEP** parameter and creep controls
- Load incrementation immediately followed by a series of creep increments specified by **AUTO CREEP**
- Use of user subroutine **CRPLAW** and/or user subroutine **NASSOC**

The following load incrementation procedure enables a viscoplastic problem to be solved:

1. Apply an elastic load increment that exceeds the steady-state yield stress.
2. Relieve the high yield stresses by turning on the **AUTO CREEP** option.

You may repeat steps 1 and 2 as many times as necessary to achieve the required load history.

Note: The size of the load increments are not altered during the **AUTO CREEP** process so that further load increments can be effected by using the **PROPORTIONAL INCREMENT** option.

The viscoplastic approach converts an iterative elastic-plastic method to one where a fraction of the initial force vector is applied at each increment with the time step controls. The success of the method depends on the proper use of the automatic creep time step controls. This means that it is necessary to select an initial time step that will satisfy the tolerances placed on the allowable stress change.

$$\text{The initial time step } \Delta t = \frac{\text{allowable stress change} \times 0.7}{\text{Maximum viscoplastic strain rate} \times \text{Young's modulus}}$$

The allowable stress change is specified in the creep controls. The most highly stressed element usually yields the maximum strain rate. It is also important to select a total time that gives sufficient number of increments to work off the effects of the initial force vector. A total time of 30 times the estimated Δt is usually sufficient.

MSC.Marc does not distinguish between viscoplastic and creep strains. A user subroutine **NASSOC** allows you to specify a nonassociated flow rule for use with the equivalent creep strains (viscoplastic) that are calculated by subroutine **CRPLAW**. A flag is set in the **CREEP** parameter in order to use the viscoplastic option with a nonassociated flow rule.

The viscoplasticity feature can be used to implement very general constitutive relations with the aid of user subroutines **ZERO** and **YIEL**.

Since the viscoplasticity model in MSC.Marc is a modified creep model, you should familiarize yourself with the creep analysis procedure (see **Nonlinear Analysis** at the beginning of this chapter).

Implicit Method

A general unified viscoplastic material law can be implemented through user subroutine **UVSCPL**. When using this method, you are responsible for defining the inelastic strain increment and the current stress.

Fracture Mechanics

Linear Fracture Mechanics

Linear fracture mechanics presupposes existence of a crack and examines the conditions under which crack growth occurs. In particular, it determines the length at which a crack propagates rapidly for specified load and boundary conditions. The concept of linear fracture mechanics stems from Griffith's work on purely brittle materials. Griffith stated that, for crack propagation, the rate of elastic energy release should at least equal the rate of energy needed for creation of a new crack surface. This concept was extended by Irwin to include limited amounts of ductility. In Irwin's considerations, the inelastic deformations are confined to a very small zone near the tip of a crack.

The basic concept presented by Griffith and Irwin is an energy balance between the strain energy in the structure and the work needed to create a new crack surface. This energy balance can be expressed using the *energy release rate* G as

$$G = G_c \quad (5-104)$$

G is defined as

$$G = -\frac{d\Pi}{da} \quad (5-105)$$

where Π is the strain energy and a is the crack length. G depends on the geometry of the structure and the current loading. G_c is called the *fracture toughness* of the material. It is a material property which is determined from experiments. Note that the energy release rate is not a time derivative but a rate of change in potential energy with crack length. An important feature of Equation 5-104 is that it can be used as a fracture criterion; a crack starts to grow when G reaches the critical value G_c .

The stress and strain fields near the tip of a crack are singular for a linear elastic material model. The stresses and strains have the principal form

$$\begin{aligned} \sigma &= \frac{K}{\sqrt{r}} f(\theta) \\ \varepsilon &= \frac{K}{\sqrt{r}} g(\theta) \end{aligned} \quad (5-106)$$

in a polar coordinate system centered at the crack tip. Thus, a linear elastic material is said to have a $1/\sqrt{r}$ singularity near a crack tip. It is easy to demonstrate that in both Griffith's and Irwin's considerations, the elastic energy release rate is determined by a single parameter: the strength of the singularity in the elastic stress field at the crack tip. This is the so-called *stress intensity factor*, and is usually denoted by capital K . The magnitude of K depends on the crack length, the distribution and intensity of applied loads, and the geometry of the structure. Crack propagation occurs when any combination of these factors causes a stress intensity factor K to be equal to or greater than the experimentally determined material property K_c , which is equivalent to [Equation 5-104](#). Hence, the objective of linear fracture mechanics calculations is to determine the value of K .

There are three possible modes of crack extension in linear elastic fracture mechanics: the opening mode, sliding mode, and tearing mode (see [Figure 5-18](#)).

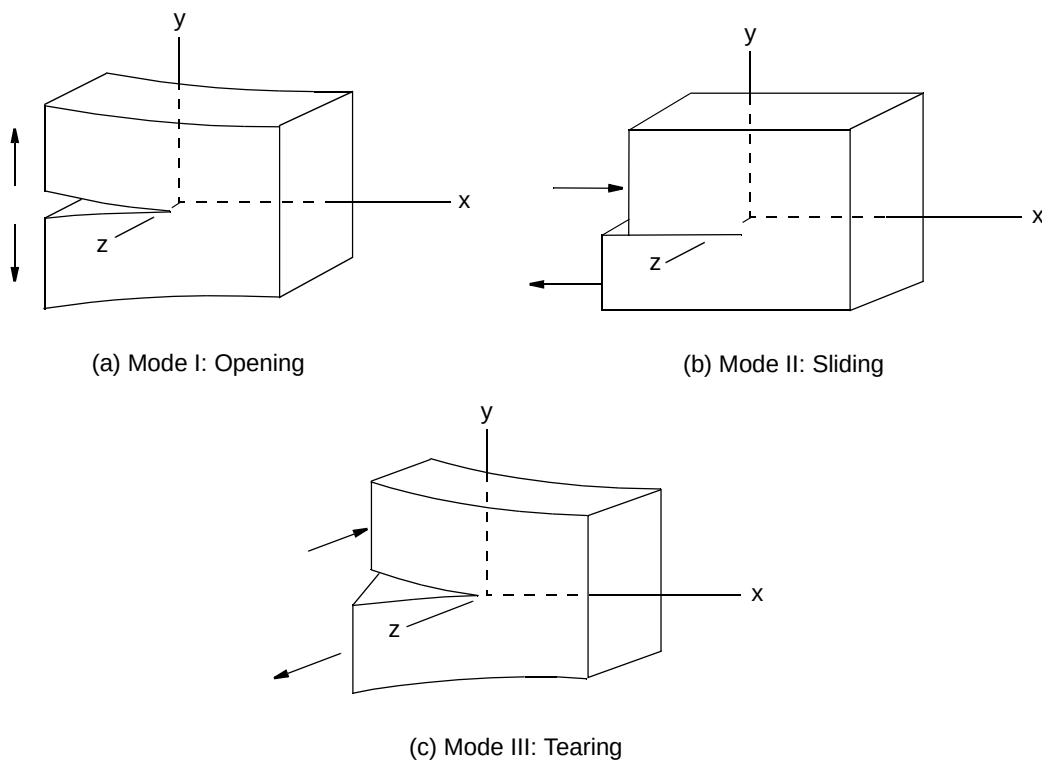


Figure 5-18 Irwin's Three Basic Modes of Crack Extension

The opening mode (see [Figure 5-18a](#)), Mode I, is characterized by the symmetric separation of the crack surfaces with respect to the plane, prior to extension (symmetric with respect to the X-Y and X-Z planes).

The sliding mode, Mode II, is characterized by displacements in which the crack surfaces slide over one another perpendicular to the leading edge of the crack (symmetric with respect to the X-Y plane and skew-symmetric with respect to the X-Z plane).

The tearing mode, Mode III, finds the crack surfaces sliding with respect to one another parallel to the leading edge (skew-symmetric with respect to the X-Y and X-Z planes).

It is customary to associate a stress intensity factor with each of these mode: K_I , K_{II} , and K_{III} . There is also an associated fracture toughness associated with each mode: K_{Ic} , K_{IIc} , and K_{IIIc} . The most critical mode is usually mode I and in many cases the other modes are not considered. The connection between the energy release rate and the stress intensity factors is given by

$$G = \frac{K_I^2}{E'} + \frac{K_{II}^2}{E'} + \frac{1 + \nu}{E} K_{III}^2 \quad (5-107)$$

where

$$E' = E \text{ for plane stress} \quad (5-108)$$

and

$$E' = \frac{E}{1 - \nu^2} \text{ for plane strain} \quad (5-109)$$

MSC.Marc uses the so-called J-integral for evaluating the energy release rate, see below. The J-integral is similar to G but is more general and is also used for nonlinear applications. J is equivalent to G when a linear elastic material model is used.

Nonlinear Fracture Mechanics

Nonlinear fracture mechanics is concerned with determining under which conditions crack propagation (growth) occurs. In this sense, nonlinear fracture mechanics is similar to linear fracture mechanics. However, there are additional questions addressed in nonlinear fracture mechanics. Is the crack propagation stable or unstable? If it is stable, at which speed does it occur? After some propagation, is the crack arrested?

Nonlinear fracture mechanics is still a research topic. Much work still needs to be done, especially on propagating cracks. The main problems with propagating cracks are the large number of parameters and the high cost of numerical analysis. Research on stationary cracks has been developed over the past years, and some important results have been obtained.

There also exists a singularity at the crack tip in fracture mechanics problems with nonlinear elastic-plastic material behavior, though the singularity is of a different nature. If one takes an exponential hardening law of the form

$$\frac{\sigma}{\sigma_0} = \left(\frac{\varepsilon}{\varepsilon_0} \right)^{1/n} \quad (5-110)$$

it can be shown that the singularities in the strain and the stresses at the crack tip are of the form

$$\begin{aligned} \varepsilon &= f(\theta) r^{-n/(n+1)} \\ \sigma &= g(\theta) r^{-1/(n+1)} \end{aligned} \quad (5-111)$$

If n approaches infinity, the material behavior becomes perfectly plastic and the singularity in the stresses vanishes. The singularity in the strains, however, takes the form of

$$\varepsilon = f(\theta) r^{-1} \quad (5-112)$$

It has not been possible to establish that the strength of the singularity is the only factor that influences initiation of crack propagation for nonlinear situations. In fact, it is doubted that initiation of crack propagation is dependent on only a single factor. The J-integral probably offers the best chance to have a single parameter to relate to the initiation of crack propagation.

The J-integral was introduced by Rice as a path-independent contour integral for the analysis of cracks. As previously mentioned, it is equivalent to the energy release rate for a linear elastic material model. It is defined in two dimensions as

$$J = \int_{\Gamma} \left((W + T) n_1 - \sigma_{ij} n_i \frac{\partial u_j}{\partial x_1} \right) d\Gamma \quad (5-113)$$

where W is the strain energy density, T is the kinetic energy density, σ_{ij} is the stress tensor and u_i is the displacement vector. The integration path Γ is a curve surrounding the crack tip, see [Figure 5-19](#). The J-integral is independent of the path Γ

as long as it starts and ends at the two sides of the crack face and no other singularities are present within the path. This is an important feature for the numerical evaluation since the integral can be evaluated using results away from the crack tip.

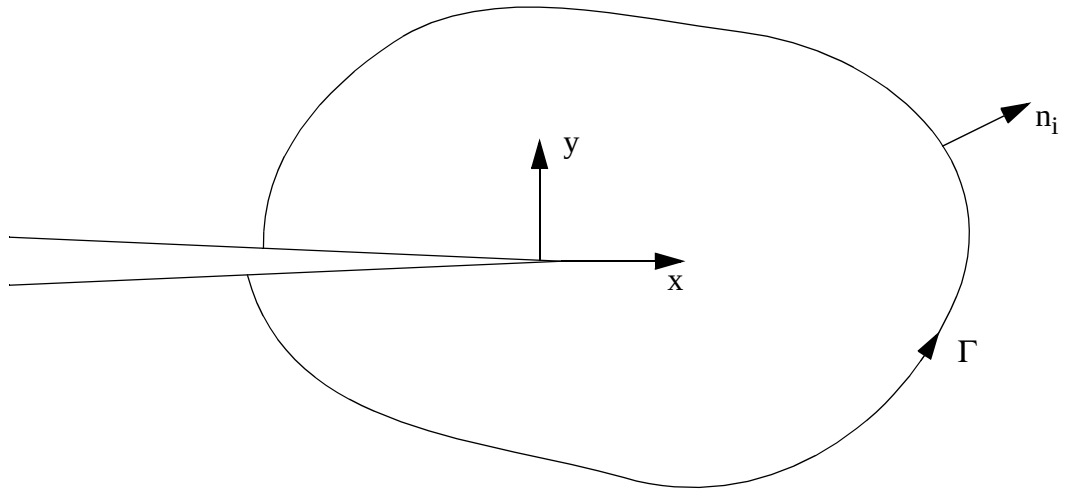


Figure 5-19 Definition of the J-integral

Numerical Evaluation of the J-integral

The J-integral evaluation in MSC.Marc is based upon the domain integration method as described in reference [1]. A direct evaluation of Equation 5-106 is not very practical in a finite element analysis due to the difficulties in defining the integration path Γ . In the domain integration method for two dimensions, the line integral is converted into an area integration over the area inside the path Γ . This conversion is exact for the linear elastic case and also for the nonlinear case if the loading is proportional, that is, if no unloading occurs. By choosing this area as a set of elements, the integration is straightforward using the finite element solution. In two dimensions, the converted expression is

$$\bar{J} = \int_A \left(\sigma_{ij} \frac{\partial u_j}{\partial x_1} - w \delta_{1i} \right) \frac{\delta q_1}{\delta x_i} dA \quad (5-114)$$

for the simplified case of no thermal strains, body forces or pressure on the crack faces. For the general expression, see reference [Ref. 1]. A is the area inside Γ and q_1 is a function introduced in the conversion into an area integral. The function q_1 can

be chosen fairly generally, as long it is equal to one at the crack tip and zero on Γ . The form of the function chosen in MSC.Marc is that it has the constant value of one at all nodes inside Γ , and decreases to zero over the outermost ring of elements in A . It can be interpreted as a rigid translation of the nodes inside Γ while the nodes on Γ remain fixed. Thus, the contribution to Equation 5-114 comes only from the elements in a ring away from the crack tip. This interpretation is that of virtual crack extension and this method can be seen as a variant of such a technique, although it is extended with the effects of thermal strains, body forces, and pressure on the crack faces. The set of nodes moved rigidly is referred to as the *rigid region* and the function q_1 as the *shift function* or *shift vector*.

In three dimensions, the line integral becomes an area integral where the area is surrounding a part of the crack front. In this case, the selection of the area is even more cumbersome than in two dimension. The converted integral becomes a volume integral which is evaluated over a set of elements. The rigid region is a set of nodes which contains a part of the crack front, and the contribution to the integral comes from the elements which have at least one but not all its nodes in the rigid region.

Determination of the rigid region and the shift vector

For the evaluation of the J-integral, MSC.Marc requires the nodes along the crack front, the shift vector, and the nodes of the rigid region. MSC.Marc allows three ways of defining the rigid region. The first is direct input; the nodes in the rigid region are listed explicitly. With this variant, the shift vector is also specified directly. The second and third variants use an automatic search for the nodes of the rigid region. The second variant is based on the mesh topology (connectivity) where a number of regions of increasing size are found by MSC.Marc. The first region for two dimensions consists of the nodes of all elements connected to the crack tip node. The second region consists of all nodes in the first region and the nodes of all elements connected to any node in the first region and so on for a given number of regions. This way, contours of increasing size are determined. In the third way of determining the rigid region, a radius is given and all nodes within that radius are part of the rigid region. For the automatic search methods the shift vector can also be determined automatically. It is then determined using the first element edge on the crack face (for meshes with notches see below). Symmetry at the crack face is automatically detected by MSC.Marc, also for the case that the symmetry condition is applied by means of a rigid contact body.

In three dimensions, it is a bit more complicated. The crack front is defined by an unsorted list of nodes. In order to obtain the variation of the J-integral along the crack front, a disk of nodes with the normal of the disk directed along the tangent to the crack front can be determined. This type of disk can readily be defined if the mesh

around the crack front consists of brick elements in a regular mesh. This mesh is typically created from a two-dimensional mesh which is extruded along the crack front. The topology based determination of the rigid region assumes such a mesh and creates disks of increasing size at each crack front node from element faces. The shift vector at each crack front node is automatically determined to be perpendicular to both the tangent to the crack front and the normal to the crack face at each crack front node. At the first and last crack front node, where a free surface is assumed to exist, the shift direction is projected to the tangent to the free surface. This is important since the shift must not change the outer boundary of the model. The geometry based search method here works with a cylinder with the axis aligned with the tangent of the crack front and centered at the crack front node. The length of the cylinder is given as a fraction of the distance to the neighboring crack front nodes. All nodes within the cylinder are part of the rigid region. This method is useful if, for instance, the mesh around the crack front is created with an automatic mesh generator. The shift vector is determined in the same way as for the topology based search method.

In elasto-plastic analyses, it is often advantageous to use a mesh where there are several (multiple) nodes at the crack tip. If collapsed elements are used at the crack tip, the nodes are kept separate and a notch is formed as the crack tip deforms. For this kind of mesh, a “multiple nodes” distance is given, which should be smaller than the smallest element at the crack tip. All nodes within that distance are then considered part of the crack tip, and the first contour for the topology based search will consist of all elements connected to any of these nodes (in three-dimensions of element faces connected to any of these nodes). The normal to the crack face used for determining the shift vector is taken as the first face outside the crack tip nodes. Thus, it is also possible to model the crack with an initial notch.

Modeling Considerations

The main difficulty in the finite element analysis of linear elastic fracture mechanics is representing the solution near the crack front. Typically, a focused mesh is used with the elements in a spider web shape. This gives a mesh with rings of elements in a radial fashion and is particularly useful for studying the variation of the values of J with contour radius. If higher order elements are used for an elastic analysis, it is advantageous to use so-called quarter point elements at the crack tip. These are regular elements where the midside nodes at element sides for which one node is part of the crack tip are shifted towards the crack tip to the quarter point location along the side. This gives a more accurate description of the singularity at the crack tip and is important to use if a coarse mesh is used.

In three-dimensions, the mesh along the crack front is typically created from a planar mesh which is extruded along the crack front as mentioned above. This allows the rigid regions to be created in a regular manner which is advantageous for accuracy. Note that MSC.Marc allows rigid contact surfaces to be used for applying symmetry conditions. For a three-dimensional example, see the *New Features Guide, Chapter 1*.

Dynamic Crack Propagation

The concepts of fracture mechanics discussed in previous sections have been applied to the prediction of crack initiation, as well as slow stable crack growth of statically loaded structures and for the prediction of fatigue crack growth in cyclically loaded structures. In problems where inertial effects cannot be ignored, application of quasi-static fracture mechanics techniques can lead to erroneous conclusions. The use of dynamic fracture mechanics concepts for these problems is clearly of necessity. The main emphasis on dynamic fracture mechanics is to predict the initiation of stationary cracks in structures, which are subjected to impact loading. It also focuses on the conditions for the continuous growth of fast propagating cracks, and on the conditions under which a crack is arrested.

The problem of predicting the growth rate and the possible crack arrest point is quite complicated. It is often treated by means of a so-called dynamic fracture methodology, which requires the combined use of experimental measurements and of detailed finite element analyses. An essential step in this approach is formed by the numerical simulation of propagating cracks by means of the finite element method.

The J-integral as implemented in MSC.Marc takes into account the effect of inertial and body forces, thermal and mechanical loading and initial strains.

The use of the **DYNAMIC** parameter and the **LORENZI** model definition option allows for the calculation of dynamic energy release rates for cracked bodies which are subjected to arbitrary thermal and mechanical loadings including initial stresses.

Dynamic Fracture Methodology

In complete similarity with static fracture mechanics concepts, it is assumed that dynamic crack growth processes for linear materials are governed by the following condition:

$$K_I(t) = R_{ID}(\dot{a}, T, B) \quad \dot{a} \geq 0 \quad (5-115)$$

where $K_I(t)$ is the dynamic stress intensity factor for mode I, $\dot{a}$ is the crack velocity, and R_{ID} is the dynamic crack propagation toughness, which is assumed to be a material parameter that in general depends on crack velocity $\dot{a}$, temperature T , and specimen thickness B .

The dynamic stress intensity factor depends on crack length (a), applied loading (σ), time (t), specimen dimensions (D), temperature (T) and initial stress fields (σ_i) caused by residual stresses or by an initial strain field. The prediction of the crack propagation history and crack arrest event demands complete knowledge of the R_{ID} vs. $\dot{a}$ relation. The Dynamic Fracture Methodology procedure consists of the following two phases:

1. *Generation phase* – in this phase, a crack arrest experiment is performed, yielding a crack propagation-versus-time curve. In addition, a numerical simulation of the experiment is carried out by using the measured crack propagation curve. This is used as input for the numerical model. This allows the calculation of dynamic stress intensity factors as a function of time. Combination of the latter relation with the measured crack propagation curve results in a curve, which can be considered the dynamic crack propagation toughness-versus-crack velocity relation.
2. *Application phase* – in order to predict the crack growth and possible crack arrest point in a structural component, the inverse problem is solved. Now, the actual stress intensity factors are calculated for the structural component, that is subjected to a particular loading history, by means of a dynamic finite element analysis. These calculated values are compared to the fracture toughness curve obtained during the generation phase, Equation 5-115, and from this the crack growth is predicted.

Dynamics

MSC.Marc's dynamic analysis capability allows you to perform the following calculations:

- [Eigenvalue Analysis](#)
- [Transient Analysis](#)
- [Harmonic Response](#)
- [Spectrum Response](#)

The program contains two methods for eigenvalue extraction and three time integration operators. Nonlinear effects, including material nonlinearity, geometric nonlinearity, and boundary nonlinearity, can be incorporated.

Linear problems can be analyzed using modal superposition or direct integration. All nonlinear problems should be analyzed using direct integration methods.

In addition to distributed mass, you can also attach concentrated masses associated with each degree of freedom of the system. You can include damping in either the modal superposition or the direct integration methods. You can also include (nonuniform) displacement and/or velocity as an initial condition, and apply time-dependent forces and/or displacements as boundary conditions.

Eigenvalue Analysis

MSC.Marc uses either the inverse power sweep method or the Lanczos method to extract eigenvalues and eigenvectors. The [DYNAMIC](#) parameter is used to determine which procedure to use, and how many modes are to be extracted. The inverse power sweep method is typically used for extracting a few modes while the Lanczos method is optimal for several modes. After the modes are extracted, they can be used in a transient analysis or spectrum response calculation.

In dynamic eigenvalue analysis, we find the solution to an undamped linear dynamics problem:

$$(K - \omega^2 M)\phi = 0$$

where K is the stiffness matrix, M is the mass matrix, ω are the eigenvalues (frequencies) and ϕ are the eigenvectors. In MSC.Marc, if the extraction is performed after increment zero, K is the tangent stiffness matrix, which can include material and geometrically nonlinear contributions. The mass matrix is formed from both distributed mass and point masses.

Inverse Power Sweep

MSC.Marc creates an initial trial vector. To obtain a new vector, the program multiplies the initial vector by the mass matrix and the inverse (factorized) stiffness matrix. This process is repeated until convergence is reached according to either of the following criteria: single eigenvalue convergence or double eigenvalue convergence. In single eigenvalue convergence, the program computes an eigenvalue at each iteration. Convergence is assumed when the values of two successive iterations are within a prescribed tolerance. In double eigenvalue convergence, the program assumes that the trial vector is a linear combination of two eigenvectors.

Using the three latest vectors, the program calculates two eigenvalues. It compares these two values with the two values calculated in the previous step; convergence is assumed if they are within the prescribed tolerance.

When an eigenvalue has been calculated, the program either exits from the extraction loop (if a sufficient number of vectors has been extracted) or it creates a new trial vector for the next calculation. If a single eigenvalue was obtained, MSC.Marc uses the double eigenvalue routine to obtain the best trial vector for the next eigenvalue. If two eigenvalues were obtained, the program creates an arbitrary trial vector orthogonal to the previously obtained vectors.

After MSC.Marc has calculated the first eigenvalue, it orthogonalizes the trial vector at each iteration to previously extracted vectors (using the Gram-Schmidt orthogonalization procedure). Note that the power shift procedure is available with the inverse power sweep method.

To select the power shift, set the following parameters:

- *Initial shift frequency* – This is normally set to zero (unless the structure has rigid body modes, preventing a decomposition around the zeroth frequency).
- *Number of modes to be extracted between each shift* – A value smaller than five is probably not economical because a shift requires a new decomposition of the stiffness matrix.
- *Auto shift parameter* – When you decide to do a shift, the new shift point is set to

$$\text{Highest frequency}^2 + \text{scalar} \times (\text{highest frequency} - \text{next highest frequency})^2$$

You can define the value of the scalar through the **MODAL SHAPE** option.

The Lanczos Method

The Lanczos algorithm converts the original eigenvalue problem into the determination of the eigenvalues of a tri-diagonal matrix. The method can be used either for the determination of all modes or for the calculation of a small number of modes. For the latter case, the Lanczos method is the most efficient eigenvalue extraction algorithm. A simple description of the algorithm is as follows.

Consider the eigenvalue problem:

$$-\omega^2 M u + K u = 0 \quad (5-116)$$

Equation 5-116 can be rewritten as:

$$\frac{1}{\omega^2} M u = M K^{-1} M u \quad (5-117)$$

Consider the transformation:

$$u = Q \eta \quad (5-118)$$

Substituting Equation 5-118 into Equation 5-117 and premultiplying by the matrix Q^T on both sides of the equation, we have

$$\frac{1}{\omega^2} Q^T M Q \eta = Q^T M K^{-1} M Q \eta \quad (5-119)$$

The Lanczos algorithm results in a transformation matrix Q such that:

$$Q^T M Q = I \quad (5-120)$$

$$Q^T M K^{-1} M Q = T \quad (5-121)$$

where the matrix T is a symmetrical tri-diagonal matrix of the form:

$$T = \begin{bmatrix} \alpha_1 & \beta_2 & & 0 \\ \beta_2 & \alpha_2 & \beta_3 & \\ & & & \beta_m \\ 0 & \beta_m & \alpha_m & \end{bmatrix} \quad (5-122)$$

Consequently, the original eigenvalue problem, Equation 5-117, is reduced to the following new eigenvalue problem:

$$\frac{1}{\omega^2} \eta = T \eta \quad (5-123)$$

The eigenvalues in Equation 5-123 can be calculated by the standard QL-method.

Through the MODAL SHAPE option, you can either select the number of modes to be extracted, or a range of modes to be extracted. The Sturm sequence check can be used to verify that all of the required eigenvalues have been found. In addition, you can select the lowest frequency to be extracted to be greater than zero.

The Lanczos procedure also allows you to restart the analysis at a later time and extract additional roots. It is unnecessary to recalculate previously obtained roots using this option.

Convergence Controls

Eigenvalue extraction is controlled by:

- a. The maximum number of iterations per mode in the power sweep method; or the maximum number of iterations for all modes in the Lanczos iteration method,
- b. an eigenvalue has converged when the difference between the eigenvalues in two consecutive sweeps divided by the eigenvalue is less than the tolerance, and
- c. the Lanczos iteration method has converged when the normalized difference between all eigenvalues satisfies the tolerance.

Modal Stresses and Reactions

After the modal shapes (and frequencies) are extracted, the RECOVER history definition option allows for the recovery of stresses and reactions at a specified mode. This option can be repeated for any of the extracted modes. The stresses are computed from the modal displacement vector ϕ ; the nodal reactions are calculated from

$F = K\phi - \omega^2 M\phi$. The RECOVER option is also used to place eigenvectors on the post file.

Transient Analysis

Transient dynamic analysis deals with an initial-boundary value problem. In order to solve the equations of motion of a structural system, it is important to specify proper initial and boundary conditions. You obtain the solution to the equations of motion by using either modal superposition (for linear systems) or direct integration (for linear or nonlinear systems). In direct integration, selecting a proper time step is very important. For both methods, you can include damping in the system.

The following sections discuss the six aspects of transient analysis listed below.

- [Modal Superposition](#)
- [Direct Integration](#)
- [Time Step Definition](#)
- [Initial Conditions](#)
- [Time-Dependent Boundary Conditions](#)
- [Damping](#)

Modal Superposition

The modal superposition method predicts the dynamic response of a linear structural system. In using this method, we assume that the dynamic response of the system can be expressed as a linear combination of the mode shapes of the system. For the principle of superposition to be valid, the structural system must be linear. Damping can be applied to each mode used in the superposition procedure.

To select the eigenvalue extraction method and the number of modal shapes, use the parameter [DYNAMIC](#). To select a fraction of critical damping for each mode, use the [DAMPING](#) model definition option. The history definition option [DYNAMIC CHANGE](#) can be used to select the time step.

MSC.Marc obtains the transient response on the basis of eigenmodes extracted. The number of modes extracted (rather than the choice of time step) governs the accuracy of the solution.

Consider the general linear undamped problem

$$M \ddot{u} + K u = f(t) \quad (5-124)$$

and suppose that n eigenvectors $\phi_1, \dots, \phi_n$ are known.

$$u = \sum \phi_i u_i(t) \quad (5-125)$$

The eigenmodes are orthogonal with respect to the M and K matrices. After substitution in [Equation 5-124](#), we find a set of uncoupled dynamic equations

$$m_i \ddot{u}_i + k_i u_i = f_i(t) \quad (5-126)$$

where:

$$m_i = \phi_i^T M \phi_i \quad (5-127)$$

$$k_i = \phi_i^T K \phi_i \quad (5-128)$$

$$f_i(t) = \phi_i^T f(t) \quad (5-129)$$

We can introduce damping on each mode as a fraction of critical damping (μ_i) for that mode. We can rewrite [Equation 5-126](#) in the form

$$m_i \ddot{u}_i + 2m_i \omega_i \mu_i \dot{u}_i + k_i u_i = f_i(t) \quad (5-130)$$

or if we normalize Φ_i such that $m_i = 1$:

$$\ddot{u}_i + 2\omega_i \mu_i \dot{u}_i + \omega_i^2 u_i = f_i(t) \quad (5-131)$$

The response of a particular mode is then given by the solution of [Equation 5-131](#) which is:

$$u_i(t) = \int_0^t f_i(\tau) h(t-\tau) d\tau \quad (5-132)$$

with:

$$h(t) = \frac{1}{\omega_i^d} e^{-\mu_i \omega_i t} \sin(\omega_i^d t) \quad t \geq 0$$

$$h(t) = 0 \quad t < 0$$

$$\omega_i^d = \sqrt{(1 - \mu_i^2)} \cdot \omega_i$$

The evaluation of the Duhamel integral, [Equation 5-132](#), can be performed exactly if the load changes linearly in a specific time increment. Hence, if the load changes rapidly in a specific time period, small load steps have to be taken.

In case initial displacements u^0 or initial velocities $\dot{u}^0$ are given a transformation to the reduced modal system is needed for those conditions:

$$u_i^0 = \Phi_i^T \cdot M \cdot u^0 \quad \text{and} \quad \dot{u}_i^0 = \Phi_i^T \cdot M \cdot \dot{u}^0$$

The solution of these initial conditions which must be added to the response given in [Equation 5-132](#) is:

$$u_i(t) = e^{-\mu_i \omega_i t} \left[\frac{\dot{u}_i^0 + u_i^0 \mu_i \omega_i}{\omega_i^d} \cdot \sin(\omega_i^d \cdot t) + u_i^0 \cdot \cos(\omega_i^d \cdot t) \right] \quad (5-133)$$

The initial accelerations due to given initial displacements u^0 and initial velocities $\dot{u}^0$ can be obtained by differentiation of [Equation 5-133](#):

$$\ddot{u}_i^0 = -(\omega_i^2 \cdot u_i^0) - (2 \cdot \mu_i \cdot \omega_i \cdot \dot{u}_i^0)$$

Direct Integration

Direct integration is a numerical method for solving the equations of motion of a dynamic system. It is used for both linear and nonlinear problems. In nonlinear problems, the nonlinear effects can include geometric, material, and boundary nonlinearities. For transient analysis, MSC.Marc offers three direct integration operators listed below.

- [Newmark-beta Operator](#)
- [Houbolt Operator](#)
- [Central Difference Operator](#)

To select the direct integration operator, use the [DYNAMIC](#) parameter. Specify the time step size through the [DYNAMIC CHANGE](#) or [AUTO STEP](#) option. Direct integration techniques are imprecise; this is true regardless of which technique you use. Each technique exhibits at least one of the following problems: conditional stability, artificial damping, and phase errors.

Newmark-beta Operator

This operator is probably the most popular direct integration method used in finite element analysis. For linear problems, it is unconditionally stable and exhibits no numerical damping. The Newmark-beta operator can effectively obtain solutions for linear and nonlinear problems for a wide range of loadings. The procedure allows for change of time step, so it can be used in problems where sudden impact makes a reduction of time step desirable. This operator can be used with adaptive time step

control. Although this method is stable for linear problems, instability can develop if nonlinearities occur. By reducing the time step and/or adding (stiffness) damping, you can overcome these problems.

Houbolt Operator

This operator has the same unconditional stability as the Newmark-beta operator. In addition, it has strong numerical damping characteristics, particularly for higher frequencies. This strong damping makes the method very stable for nonlinear problems as well. In fact, stability increases with the time step size. The drawback of this high damping is that the solution can become inaccurate for large time steps. Hence, the results obtained with the Houbolt operator usually have a smooth appearance, but are not necessarily accurate. The Houbolt integration operator, implemented in MSC.Marc as a fixed time step procedure, is particularly useful in obtaining a rough *scoping* solution to the problem.

Single Step Houbolt Operator

Two computational drawbacks of the Houbolt operator are the requirement of a special starting procedure and the restriction to fixed time steps. In [Ref. 30], a Single Step Houbolt procedure has been presented, being unconditionally stable, second order accurate and asymptotically annihilating. In this way, the algorithm is computationally more convenient compared to the standard Houbolt method, but because of its damping properties, the time steps have to be chosen carefully. This algorithm is recommended for implicit dynamic contact analyses.

Central Difference Operators

These explicit operators for IDYN = 4 and IDYN = 5 are only conditionally stable. The program automatically calculates the maximum allowable time step. This method is not very useful for shell or beam structures because the high frequencies result in a very small stability limit. This method is particularly useful for analysis of shock-type phenomena. In this procedure, since the operator matrix is a diagonal mass matrix, no inverse of operator matrix is needed. However, this fact also implies that you cannot use this method in problems having degrees of freedom with zero mass. This *restriction* precludes use of the Herrmann elements, gap-friction elements, the pipe bend element, shell elements 72 and 89 and beam elements 76 and 77. These shell and beam elements are precluded because they have a rotational degree of freedom, which do not have an associated mass. The mass is updated only if the **UPDATE** parameter or the **CONTACT** option is used. The elastomer capability can be used with explicit dynamics in an updated Lagrange framework where the pressure variables are condensed out before going into the solver.

Technical Background

Consider the equations of motion of a structural system:

$$Ma + Cv + Ku = F \quad (5-134)$$

where M , C , and K are mass, damping, and stiffness matrices, respectively, and a , v , u , and F are acceleration, velocity, displacement, and force vectors. Various direct integration operators can be used to integrate the equations of motion to obtain the dynamic response of the structural system. The technical background of the three direct integration operators available in MSC.Marc is described below.

Newmark-beta Operator

The generalized form of the Newmark-beta operator is

$$u^{n+1} = u^n + \Delta t v^n + (1/2 - \beta)\Delta t^2 a^n + \beta\Delta t^2 a^{n+1} \quad (5-135)$$

$$v^{n+1} = v^n + (1 - \gamma)\Delta t a^n + \gamma\Delta t a^{n+1} \quad (5-136)$$

where superscript ^{n} denotes a value at the n th time step and u , v , and a take on their usual meanings.

The particular form of the dynamic equations corresponding to the trapezoidal rule

$$\gamma = 1/2, \quad \beta = 1/4$$

results in

$$\left(\frac{4}{\Delta t^2}M + \frac{2}{\Delta t}C + K\right)\Delta u = F^{n+1} - R^n + M\left(a^n + \frac{4}{\Delta t}v^n\right) + Cv^n \quad (5-137)$$

where the internal force R is

$$R = \int_V \beta^T \sigma dv \quad (5-138)$$

Equation 5-137 allows implicit solution of the system

$$u^{n+1} = u^n + \Delta u \quad (5-139)$$

Notice that the operator matrix includes K , the tangent stiffness matrix. Hence, any nonlinearity results in a reformulation of the operator matrix. Additionally, if the time step changes, this matrix must be recalculated because the operator matrix also depends on the time step. It is possible to change the values of γ and β through the [PARAMETERS](#) option.

Houbolt Operator

The Houbolt operator is based on the use of a cubic fitted through three previous points and the current (unknown) in time. This results in the equations

$$v^{n+1} = \left(\frac{11}{6}u^{n+1} - 3u^n + \frac{3}{2}u^{n-1} - \frac{1}{3}u^{n-2} \right) / \Delta t \quad (5-140)$$

and

$$a^{n+1} = (2u^{n+1} - 5u^n + 4u^{n-1} - u^{n-2}) / \Delta t^2 \quad (5-141)$$

Substituting this into the equation of motion results in

$$\left(\frac{2}{\Delta t^2}M + \frac{11}{6\Delta t}C + K \right) \Delta u = F^{n+1} - R^n + \frac{1}{\Delta t^2}(3u^n - 4u^{n-1} + u^{n-2})M + \frac{1}{\Delta t} \left(\frac{7}{6}u^n - \frac{3}{2}u^{n-1} + \frac{1}{3}u^{n-2} \right) C \quad (5-142)$$

This equation provides an *implicit* solution scheme. By solving [Equation 5-139](#) for Δu , you obtain [Equation 5-143](#), and so obtain v^{n+1} and a^{n+1} .

$$u^{n+1} = u^n + \Delta u \quad (5-143)$$

[Equation 5-142](#) is based on uniform time steps – errors occur when the time step is changed. Also, a special starting procedure is necessary since u^{n-1} and u^{n-2} appear in [Equation 5-142](#).

Single Step Houbolt Operator

The Single Step Houbolt operator according to [Ref. 30] starts with the following equilibrium equation and expressions for the velocity and acceleration:

$$\alpha^{m1}Ma^{n+1} + \alpha^{c1}Cv^{n+1} + \alpha^{k1}Ku^{n+1} + \alpha^m Ma^n + \alpha^c Cv^n + \alpha^k Ku^n = \alpha^{f1}F^{n+1} + a^f F^n \quad (5-144)$$

$$u^{n+1} = u^n + \Delta t v^n + \beta \Delta t^2 a^n + \beta^1 \Delta t^2 a^{n+1} \quad (5-145)$$

$$v^{n+1} = v^n + \gamma \Delta t a^n + \gamma^1 \Delta t a^{n+1} \quad (5-146)$$

Notice that in contrast to the Newmark and the standard Houbolt method, the equilibrium equation also contains terms corresponding to the beginning of the increment. Without loss of generality, the parameter α^{m1} can be set to 1. Based on asymptotic annihilation and second order accuracy, the remaining parameters can be shown to fulfill:

$$\alpha^k = 0, \beta = \gamma, \beta^1 = \gamma + \gamma^1, \alpha^m = -1/2, \alpha^{k1} = 1/2\beta^1,$$

$$\alpha^c = -(2\beta + \beta^1)/4\beta^{12}, \alpha^{c1} = (2\beta + 3\beta^1)/4\beta^{12}, \alpha^f = \alpha^k, \alpha^{f1} = \alpha^{k1}$$

In this way, the number of unknown parameters has been reduced to two. Based on a Taylor series expansion of the displacement about the nth time step, β and β^1 should be related by $\beta + \beta^1 = 1/2$, which finally yields $\gamma = 1/2(1/2 - \gamma^1)$. According to [Ref. 30], γ^1 should be set to 3/2 (with $\gamma = -1/2$) to minimize the velocity error and to 1/2 (with $\gamma = 0$) to avoid velocity overshoot. The default values in MSC.Marc are $\gamma^1 = 3/2$ and $\gamma = -1/2$, but the user can modify γ^1 and γ using the [PARAMETERS model definition](#) and [history definition option](#).

Substitution of the velocity and acceleration into the equilibrium equation results in:

$$\left\{ \frac{1}{\beta^1 \Delta t^2 \alpha^{k1}} M + \frac{\alpha^{c1} \gamma^1}{\beta^1 \Delta t \alpha^{k1}} C + K \right\} \Delta u = F^{n+1} - K u^n + \frac{1}{\beta^1 \Delta t^2 \alpha^{k1}} M \{ \Delta t v^n + \beta \Delta t^2 a^n \} - \frac{\alpha^m}{\alpha^{k1}} M a^n - \frac{\alpha^{c1}}{\alpha^{k1}} C \left\{ v^n + \gamma \Delta t a^n - \frac{\gamma^1}{\beta^1 \Delta t} \{ \Delta t v_n + \beta \Delta t^2 a_n \} \right\} - \frac{\alpha^c}{\alpha^{k1}} C v^n \quad (5-147)$$

Central Difference Operator

The central difference operator assumes a quadratic variation in the displacement with respect to time.

$$a^n = (v^{n+1/2} - v^{n-1/2}) / (\Delta t) \quad (5-148)$$

$$v^n = (u^{n+1/2} - u^{n-1/2}) / (\Delta t) \quad (5-149)$$

so that

$$a^n = (\Delta u^{n+1} - \Delta u^n) / (\Delta t^2) \quad (5-150)$$

where

$$\Delta u^n = u^n - u^{n-1} \quad (5-151)$$

for IDYN=4:

$$\frac{M}{\Delta t^2} \Delta u^{n+1} = F^n - R^n + \frac{M}{\Delta t^2} \Delta u^n \quad (5-152)$$

for IDYN=5:

$$\frac{M}{\Delta t^2} \Delta u^{n+1} = F^n - R^n + \frac{M}{\Delta t^2} \Delta u^n - C v^{n-\frac{1}{2}} \quad (5-153)$$

Since the mass matrix is diagonal, no inverse of the operator matrix is needed. Also, since the operator is only conditionally s, the critical time step is evaluated at the beginning of the analysis. For IDYN = 4, the critical time step is computed by a power sweep for the highest mode in the system only at the beginning of the analysis. For IDYN = 4, no damping is included. For IDYN = 5, an approximated method based on

element geometry is used to compute the highest eigenvalue. The critical time step is calculated at a user-specified frequency or every 100 steps. The variable time step can be used only for IDYN = 5. Unless there is significant distortion in an element or material nonlinearity, the change of critical time step is not significant.

Time Step Definition

In a transient dynamic analysis, time step parameters are required for integration in time. The **DYNAMIC CHANGE** option can be used for either the modal superposition or the direct integration procedure. The **AUTO STEP** option can be used for the Newmark-beta operator and the Single Step Houbolt operator to invoke the adaptive time control. Enter parameters to specify the time step size and period of time for this set of boundary conditions.

When using the Newmark-beta operator, decide which frequencies are important to the response. The time step in this method should not exceed 10 percent of the period of the highest relevant frequency in the structure. Otherwise, large phase errors will occur. The phenomenon usually associated with too large a time step is strong oscillatory accelerations. With even larger time steps, the velocities start oscillating. With still larger steps, the displacement eventually oscillates. In nonlinear problems, instability usually follows oscillation. When using adaptive dynamics, you should prescribe a maximum time step.

As in the Newmark-beta operator, the time step in Houbolt integration should not exceed 10 percent of the period of the highest frequency of interest. However, the Houbolt method not only causes phase errors, it also causes strong artificial damping. Therefore, high frequencies are damped out quickly and no obvious oscillations occur. It is, therefore, completely up to the engineer to determine whether the time step was adequate.

In nonlinear problems, the mode shapes and frequencies are strong functions of time because of plasticity and large displacement effects, so that the above guidelines can be only a coarse approximation. To obtain a more accurate estimate, repeat the analysis with a significantly different time step (1/5 to 1/10 of the original) and compare responses.

The central difference integration method is only conditionally stable; the program automatically calculates the stable time step. This step size yields accurate results for all practical problems.

Initial Conditions

In a transient dynamic analysis, you can specify initial conditions such as nodal displacements and/or nodal velocities. To enter initial conditions, use the following model definition options: **INITIAL DISP** for specified nodal displacements, and **INITIAL VEL** for specified nodal velocities. As an alternative, you can use the user subroutine **USINC**.

Time-Dependent Boundary Conditions

Simple time-dependent load or displacement histories can be entered on data blocks. However, in general cases with complex load histories, it is often more convenient to enter the history through a user subroutine. MSC.Marc allows the use of subroutines **FORCDT** and **FORCEM** for boundary conditions. Subroutine **FORCDT** allows you to specify the time-dependent incremental point loads and incremental displacements. Subroutine **FORCEM** allows you to specify the time-dependent magnitude of the distributed load.

Damping

In a transient dynamic analysis, damping represents the dissipation of energy in the structural system. It also retards the response of the structural system.

MSC.Marc allows you to enter two types of damping in a transient dynamic analysis: modal damping and Rayleigh damping. Use modal damping for the modal superposition method and Rayleigh damping for the direct integration method. For modal superposition, you can include damping associated with each mode. To do this, use the **DAMPING** option to enter the fraction of critical damping to be used with each mode.

During time integration, MSC.Marc associates the corresponding damping fraction with each mode. The program bases integration on the usual assumption that the damping matrix of the system is a linear combination of the mass and stiffness matrices, so that damping does not change the modes of the system.

For direct integration damping, you can specify the damping matrix as a linear combination of the mass and stiffness matrices of the system. You can specify damping coefficients on an element basis.

Stiffness damping should not be applied to either Herrmann elements or gap elements because of the presence of Lagrangian multipliers.

Numerical damping is used to damp out unwanted high-frequency chatter in the structure. If the time step is decreased (stiffness damping might cause too much damping), use the numerical damping option to make the damping (stiffness) coefficient proportional to the time step. Thus, if the time step decreases, high-frequency response can still be accurately represented. This type of damping is

particularly useful in problems where the characteristics of the model and/or the response change strongly during analysis (for example, problems involving opening or closing gaps).

Element damping uses coefficients on the element matrices and is represented by the equation:

$$C = \sum_{i=1}^n \left\{ \alpha_i M_i + \left(\beta_i + \gamma_i \frac{\Delta t}{\pi} \right) K_i \right\} \quad (5-154)$$

where

C is the global damping matrix

M_i is the mass matrix of i^{th} element

K_i is the stiffness matrix of the i^{th} element

α_i is the mass damping coefficient on the i^{th} element

β_i is the usual stiffness damping coefficient on the i^{th} element

γ_i is the numerical damping coefficient on the i^{th} element

Δt is the time increment

If the same damping coefficients are used throughout the structure, [Equation 5-154](#) is equivalent to Rayleigh damping.

The damping associated with springs and mass points can be controlled via the springs and masses input options. The damping on elastic foundations is the same as the damping on the element on which the foundation is applied. For springs, a dashpot can be added for nonlinear analysis.

Harmonic Response

Harmonic response analysis allows you to analyze structures vibrating around an equilibrium state. This equilibrium state can be unstressed or statically prestressed. Statically prestressed equilibrium states can include material and/or geometric nonlinearities. You can compute the damped response for prestressed structures at various states.

In many practical applications, components are dynamically excited. These dynamic excitations are often harmonic and usually cause only small amplitude vibrations. MSC.Marc linearizes the problem around the equilibrium state. If the

equilibrium state is a nonlinear, statically prestressed situation, MSC.Marc considers all effects of the nonlinear deformation on the dynamic solution. These effects include the following:

- initial stress
- change of geometry
- influence on constitutive law

The vibration problem can be solved as a linear problem using complex arithmetic.

The analytical procedure consists of the following steps:

1. MSC.Marc calculates the response of the structure to a static preload (which can be nonlinear) based on the constitutive equation for the material response. In this portion of the analysis, the program ignores inertial effects.
2. MSC.Marc calculates the complex-valued amplitudes of the superimposed response for each given frequency, and amplitude of the boundary tractions and/or displacements. In this portion of the analysis, the program considers both material behavior and inertial effects.
3. You can apply different loads with different frequencies or change the static preload at your discretion. All data relevant to the static response is stored during calculation of the complex response.

To initiate a harmonic response analysis, use the **HARMONIC** parameter. To define the excitation frequency, use the **HARMONIC** history definition option. If you enter the **HARMONIC** history definition option with a set of incremental data, MSC.Marc assumes those incremental data apply only for the harmonic excitation. This is true for applied boundary conditions as well as loads.

The small amplitude vibration problem can be written with complex arithmetic as follows

$$[K + i\omega D - \omega^2 M]\bar{u} = \bar{P} \quad (5-155)$$

where

$$\bar{u} = u_{re} + iu_{im} \text{ complex response vector}$$

$$\bar{P} = P_{re} + iP_{im} \text{ complex load vector}$$

The notation is further defined below:

$$K = \Sigma K_{el} + \Sigma K_{sp} \quad (5-156)$$

where

K_{el} are element stiffness matrices

K_{sp} are the spring stiffness matrices

$$M = \Sigma M_{el} + \Sigma M_{mp} \quad (5-157)$$

where

M_{el} are element mass matrices

M_{mp} are masspoint contributions

$$D = \Sigma D_{el} + \Sigma D_d + \Sigma \left(\alpha M + \left(\beta + \frac{2\gamma}{\omega} \right) K \right) \quad (5-158)$$

where

D_{el} are element damping matrices

D_d are damper contributions

α mass damping coefficient

β stiffness damping coefficient

γ numerical damping coefficient

$i = \sqrt{-1}$

ω = excitation frequencies

$u = u_{re} + i u_{im}$ complex response vector

$P = P_{re} + i P_{im}$ complex load vector

If all external loads and forced displacements are in phase and the system is undamped, this equation reduces to

$$(K - \omega^2 M) u_{re} = P_{re} \quad (5-159)$$

which could be solved without activating the complex arithmetic on the **HARMONIC** parameter.

The values of the damping coefficients (α , β , γ) are entered via the **DAMPING** model definition option. The spring and damper contributions are entered in the **SPRINGS** model definition option and mass points are specified in the **MASSSES** model definition option.

The element damping matrix (D_{el}) can be obtained for any material with the use of a material damping matrix which is specified in the user-entered subroutine **UCOMPL**. You specify the material response with the constitutive equation.

$$\sigma = B\varepsilon + C\dot{\varepsilon} \quad (5-160)$$

where B and C can be functions of deformation and/or frequency.

The global damping matrix is formed by the integrated triple product. The following equation is used:

$$D = \sum_{el} \int_{V_{el}} \beta^T C dV_{el} \quad (5-161)$$

where β is the strain-displacement relation.

Similarly, the stiffness matrix K is based on the elastic material matrix B . The program calculates the response of the system by solving the complex equations:

$$[K + i\omega D - \omega^2 M]\bar{u} = \bar{P} \quad (5-162)$$

where $\bar{u}$ now is the complex response vector

$$\bar{u} = u_{re} + iu_{im} \quad (5-163)$$

A special application of the harmonic excitation capability involves the use of the elastomeric analysis capability in MSC.Marc. Here, the Mooney formulation (used in conjunction with the various Herrmann elements) is used to model the stress-strain behavior of the elastomeric compound. In MSC.Marc, the behavior is derived from the third order invariant form of the strain energy density function.

$$\begin{aligned} W(I_1, I_2) = & C_{10}(I_1 - 3) + C_{01}(I_2 - 3) + C_{11}(I_1 - 3)(I_2 - 3) \\ & + C_{20}(I_1 - 3)^2 + C_{30}(I_1 - 3)^3 \end{aligned} \quad (5-164)$$

with the incompressibility constraint

$$I_3 = 1 \quad (5-165)$$

where I_1 , I_2 , and I_3 are the invariants of the deformation. For the harmonic excitation, the constitutive equation has the specific form

$$\Delta \bar{S}_{ij} = [D_{ijkl} + 2i\omega \bar{\Phi}_{ijkl}] \Delta \bar{E}_{kl} \quad (5-166)$$

with D_{ijkl} as the quasi-static moduli following from the Mooney strain energy density function and

$$\begin{aligned} \bar{\Phi}_{ijkl} = & \bar{\Phi}_0 [C_{ik}^{-1} C_{jl}^{-1} + C_{il}^{-1} C_{jk}^{-1}] + \bar{\Phi}_1 [\delta_{ik} C_{jl}^{-1} + C_{il}^{-1} \delta_{jk}] + \\ & \bar{\Phi}_2 [C_{ik} C_{jl}^{-1} + C_{il}^{-1} C_{jk}] + \bar{\Phi}_0 \delta_{ij} C_{kl}^{-1} + \bar{\Phi}_{11} \delta_{ij} \delta_{kl} + \\ & \bar{\Phi}_{12} \delta_{ij} C_{kl} + \bar{\Phi}_{20} C_{ij} C_{kl}^{-1} + \bar{\Phi}_{21} C_{ij} \delta_{kl} + \bar{\Phi}_{22} C_{ij} C_{kl} \end{aligned} \quad (5-167)$$

The output of MSC.Marc consists of stresses, strains, displacements and reaction forces, all of which may be complex quantities. The strains are given by

$$\bar{\epsilon} = \beta \bar{u} \quad (5-168)$$

and the stresses by

$$\sigma = B \bar{\epsilon} + C \dot{\bar{\epsilon}} \quad (5-169)$$

The reaction forces are calculated with

$$\bar{R} = K \bar{u} - \omega^2 M \bar{u} + i\omega \Sigma D_{el} \bar{u} + i\omega \Sigma D_d \bar{u} \quad (5-170)$$

where $-\omega^2 M \bar{u}$ is only included if requested on the **HARMONIC** parameter.

The printout of the nodal values consists of the real and imaginary parts of the complex values, but you can request that the amplitude and phase angle be printed. You do this with the **PRINT CHOICE** model definition option.

Spectrum Response

The spectrum response capability allows you to obtain maximum response of a structure subjected to known spectral base excitation response. This is of particular importance in earthquake analysis and random vibration studies. You can use the spectrum response option at any point in a nonlinear analysis and, therefore, ascertain the influence of material nonlinearity or initial stress.

The spectrum response capability technique operates on the eigenmodes previously extracted to obtain the maximum nodal displacements, velocities, accelerations, and reaction forces. You can choose a subset of the total modes extracted by either specifying the lowest n modes or by selecting a range of frequencies.

Enter the displacement response spectrum $S_D(\omega)$ for a particular digitized value of damping through user subroutine [USSD](#). MSC.Marc performs the spectrum analysis based on the latest set of modes extracted. The program lumps the mass matrix to produce $\bar{M}$. It then obtains the projection of the inertia forces onto the mode ϕ_j

$$P_j = \bar{M} \phi_j \quad (5-171)$$

The spectral displacement response for the j^{th} mode is

$$\alpha_j = S_D(\omega_j) P_j \quad (5-172)$$

MSC.Marc then calculates the root-mean-square values as

$$u = \left[\sum_j (\alpha_j \phi_j)^2 \right]^{1/2} \quad \text{DISPLACEMENT} \quad (5-173)$$

$$v = \left[\sum_j (\alpha_j \omega_j \phi_j)^2 \right]^{1/2} \quad \text{VELOCITY} \quad (5-174)$$

$$a = \left[\sum_j (\alpha_j \omega_j^2 \phi_j)^2 \right]^{1/2} \quad \text{ACCELERATION} \quad (5-175)$$

$$f = \left[\sum_j (\alpha_j \omega_j^2 \bar{M} \phi_j)^2 \right]^{1/2} \quad \text{FORCE} \quad (5-176)$$

Rigid-Plastic Flow

The rigid-plastic flow analysis is an approach to large deformation analysis which can be used for metal forming problems. Two formulations are available: an Eulerian (steady state) and Lagrangian (transient) approach. The effects of elasticity are not included. If these effects are important, this option should not be used.

In the steady state approach, the velocity field (and stress field) is obtained as the solution of a steady-state flow analysis. The time period is considered as 1.0 and, hence, the velocity is equal to the deformation. In the transient formulation, the incremental displacement is calculated.

The **R-P FLOW** parameter invokes the rigid-plastic procedure. This procedure needs to enforce the incompressibility condition, which is inherent to the strictly plastic type of material response being considered.

Incompressibility can be imposed in three ways:

1. by means of Lagrange multipliers. Such procedure requires Herrmann elements which have a pressure variable as the Lagrange multiplier.
2. by means of penalty functions. This procedure uses regular solid elements, and adds penalty terms to any volumetric strain rate that develops. It is highly recommended that the constant dilatation formulation be used – by entering a nonzero value in the second field of the **GEOMETRY** model definition option.

Penalty factor can be treated as constant or variable through the **R-P FLOW** parameter. The penalty value is entered through the **PARAMETERS** option.

3. in plane stress analysis (shell and membrane elements), the incompressibility constraint is satisfied exactly by updating the thickness. This capability is not available for steady state analysis.

In R-P flow analysis, several iterations are required at any given increment, the greatest number occurring in the first increment. Subsequent increments require fewer iterations, since the initial iteration can make use of the solution from the previous increment.

Due to the simplicity of the rigid-plastic formulation, it is possible to bypass stress recovery for all iterations but the last in each increment, provided that displacement control is used. In such cases, considerable savings in execution time are achieved. If nodal based friction is used in a contact analysis, then a stress recovery is always performed after each iteration.

Steady State Analysis

The steady state R-P flow formulation is based on an Eulerian reference system. For problems in which a steady-state solution is not appropriate, an alternative method is available to update the coordinates. User subroutine **UPNOD** is used to update the nodal coordinates at the end of a step according to the relationship.

$$x_i^n = x_i^{n-1} + v^n \Delta t \quad (5-177)$$

where n refers to the step number, v^n is the nodal velocity components, and Δt is an arbitrary time step. Δt is selected in such a way as to allow only a reasonable change in mesh shape while ensuring stability with each step.

Updating the mesh requires judicious selection of a time step. This requires some knowledge of the magnitude of the nodal velocities that will be encountered. The time step should be selected such that the strain increment is never more than one percent for any given increment.

The quantities under the title of ENGSTN in the printouts actually refer to the strain rate at an element integration point. The reaction forces output by the program gives the limit loads on the structure.

Transient Analysis

In the transient procedure, there is an automatic updating of the mesh at the end of each increment. During the analysis, the updated mesh can exhibit severe distortion and the solution might be unable to converge. Global adaptive meshing or manual mesh rezoning can be used to overcome this difficulty.

Technical Background

The rigid-plastic flow capability is based on iteration for the velocity field in an incompressible, non-Newtonian fluid. The normal flow condition for a nonzero strain rate can be expressed as:

$$\sigma'_{ij} = \left(\frac{2\bar{\sigma}}{3\bar{\epsilon}} \right) \dot{\epsilon}_{ij} = \mu(\dot{\epsilon}) \dot{\epsilon}_{ij} \quad (5-178)$$

where

$$\dot{\epsilon} = \sqrt{\frac{2}{3} \dot{\epsilon}_{ij} \dot{\epsilon}_{ij}} \quad (5-179)$$

is the equivalent strain rate, $\bar{\sigma}$ is the yield stress (which may be rate-dependent) and

$$\sigma'_{ij} = \sigma_{ij} - \frac{1}{3}\delta_{ij}\sigma_{kk} \quad (5-180)$$

gives the deviatoric stress. The effective viscosity is evaluated as:

$$\mu = \frac{2}{3} \frac{\bar{\sigma}}{\dot{\epsilon}} \quad (5-181)$$

Note that as $\dot{\epsilon} \rightarrow 0$, $\mu \rightarrow \infty$. A cutoff value of strain rate is used in the program to avoid this difficulty. An initial value for $\dot{\epsilon}$ is necessary to start the iterations. These values can be specified in the [PARAMETERS](#) option. The default cut-off value is 10^{-6} , and the default initial strain rate value is 10^{-4} .

The value of the flow stress is dependent upon both the equivalent strain, the equivalent strain rate, and the temperature. This dependence can be given through the [WORK HARD](#), [STRAIN RATE](#), and [TEMPERATURE EFFECTS](#) options, respectively. For steady state analysis, user subroutine [UNEWTN](#) can be used to define a viscosity. In this manner, a non-Newtonian flow analysis can be performed. For the transient procedure, user subroutine [URPFLO](#) can be used to define the flow stress.

Superplasticity

Superplasticity is the ability of a material to undergo extensive deformation such as strains of 1000% without necking. Superplastic behavior has been reported in numerous metal, alloys and ceramics. Every instance of superplasticity is associated with: (1) A fine grain size ($1 - 10 \mu m$), (2) deformation temperatures $> 0.4T_m$, and (3) a strain-rate sensitivity factor $m > 0.3$.

Using finite element analysis to simulate superplastic fabrication of complex parts used in the aerospace and automotive industries requires this material behavior and contact with friction. Furthermore, the process pressure needs to be automatically adjusted to keep the material within a target strain rate. The simulation can be used to predict thinning, forming time, areas of void formation, and can ultimately be employed in shape optimization; thus, reducing the number of prototypes of forming trials required to product an acceptable part.

Three mechanisms have been proposed to account for the high strain-rate sensitivity found in superplastic materials: (1) Vacancy creep, (2) creep by grain boundary diffusion, and (3) grain boundary sliding. According to Ghosh and Hamilton, the strain-rate sensitivity of metals arises from the viscous nature of the deformation process. The viscosity is a result of the resistance offered by internal obstacles within the material. In dislocation glide and climb processes, the obstacles are a fine dispersion of second phase particles within the grain interior, between which the dislocations are bent around and moved. At high homologous temperatures ($T > 0.4T_m$; where T_m is the melting temperature), the high diffusivities around grain boundary regions can lead to grain boundary sliding. The overall rate sensitivity of a material is then a result of the rate sensitivities of the grain boundary and the grain interior. The more the material behaves as a viscous liquid, the greater its superplasticity.

The superplastic behavior is characterized by the dependence of the flow stress upon the strain-rate, which is usually depicted by the logarithmic relationship shown in [Figure 5-20](#).

As indicated in [Figure 5-20](#), the stress-strain rate behavior of a superplastic material can be divided into three regions. Values of strain-rate sensitivity, m (the slope of flow stress versus strain-rate curve) which is a measure of resistance to localized necking, are relatively low in both the low stress-low strain rate region I and the high stress-high strain rate region III and superplasticity is not manifested. Rather, superplasticity is found only in region II, a transition region in which stress increases rapidly with

increasing strain-rate. As temperature increases and/or grain size decreases, region II is displaced to higher strain-rates. Moreover, the maximum observed values of m increase with similar changes in these parameters.

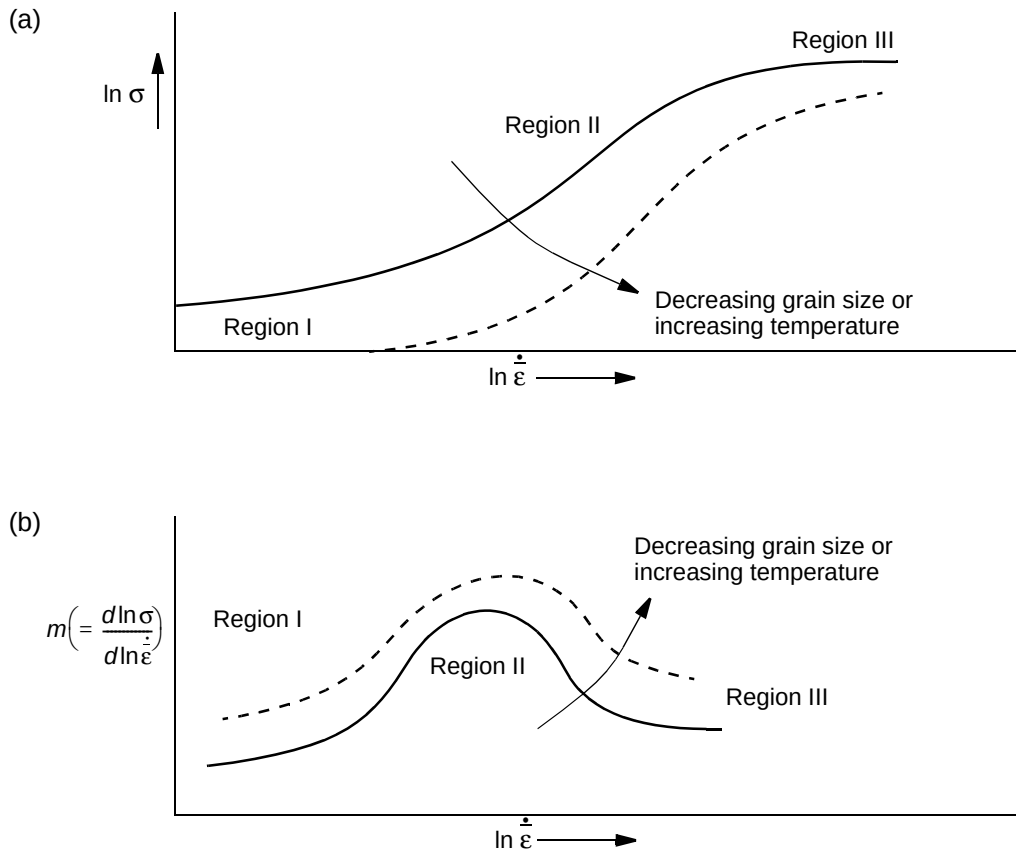


Figure 5-20 (a) Flow stress and (b) Strain-rate Sensitivity as a Function of Strain-rate

Certainly the forming process innovations evoked will need to be carefully studied and developed. Forming times are slow, and there will be a critical need for optimizing forming pressures, stress strain-rate and deflection in sheet forming. Based on the schematic flow stress-strain rate relationships given above, it is apparent that high values of m are requisite for superplastic materials. Since, for a given material and forming temperature, m , usually varies with strain-rate, it is desirable to control strain-rate during forming so that optimum or at least adequate strain-rate sensitivity is exhibited. Ductility is also dependent upon forming temperature, which must lie within a narrow range. If forming temperatures and pressure cycle are optimum, then

unlike conventional ductile materials, superplastic materials are much less susceptible to localized necking. Additionally, under such conditions, the flow stress occurring during forming is much lower than the mechanical yield stress.

Thus, the superplastic materials may be viewed as exhibiting time-dependent inelastic behavior with the yield stress as a function of time, temperature, strain-rate, total stress and total strain. Typical materials used in commercial superplastic forming applications include Ti-6Al-4V titanium alloy and 5083 aluminium alloy.

The form of constitutive equation used to simulate superplasticity is given as:

$$\sigma_y = \dot{\epsilon}^m \quad (5-182)$$

The form in [Equation 5-182](#) can be recovered by using appropriate constants in the [ISOTROPIC](#) model definition option to define power law or rate power law. Thus,

Power Law Model:
$$\sigma_y = A(\epsilon_o + \dot{\epsilon})^m + B\dot{\epsilon}^n \quad (5-183)$$

Rate Power Law Model:
$$\sigma_y = A\dot{\epsilon}^{-m}\dot{\epsilon}^n + B \quad (5-184)$$

The superplastic forming process requires the use of the [SPFLOW](#) parameter. The use of this parameter automatically activates the [FOLLOW FOR](#) and [PROCESSOR](#) parameters. The process parameters are controlled by the use of the [SUPERPLASTIC](#) history definition option.

Typical outputs that are available from the superplastic forming simulation are the thickness distribution for the part, the equivalent plastic strain rate, and the history of the process pressure.

The process pressure is automatically calculated during the analysis. The pressure magnitude is adjusted such that the equivalent strain rate in the part is at or close to the user-specified target strain rate. The equivalent strain rate in the part is an average value calculated by sampling a suitable subset of elements. The recommended scheme is one in which elements with a strain rate greater than a cut-off factor times the maximum element strain rate are sampled. This maximum strain rate is based on a smoothing algorithm described below. The cut-off factor can vary between 0 (all elements below the maximum are sampled) and 1 (only the elements with maximum strain rate are sampled). The recommended value for the cut-off factor is 0.7 to 0.9 (default value is 0.8). To reduce undesirable oscillations in the pressure-time history, a pressure smoothing algorithm is incorporated. The basis for this algorithm is a smoothing of the maximum strain rate in the mesh based on the fact that the maximum strain rate should be typically representative of a few elements in the mesh, rather than

an isolated individual value. The peak strain rates in a few elements are calculated. The number of elements that are used in this calculation varies with the cut-off factor (for a cut-off factor of 1, only 1 element is used; for the default of 0.8, 10 elements are used; for a cut-off factor of 0, 50 elements are used). The strain rate values are successively disregarded in descending order if the difference from the highest strain rate to the lowest differs by more than 10 percent from the mean.

The value of the cut-off factor has significant influence on the maximum strain rate control and on the smoothness of the pressure-time curve. Larger the factor (that is, 0.9 or higher) provides more control on the maximum strain rate, but may potentially cause oscillations in the the pressure history. Smaller the factor (that is, 0.7 or lower) provides less control on the maximum strain rate, but causes smoother pressure-time curves. The default of 0.8 should work in most cases - in situations where physically realistic localized strain rates occur and one desires good control on these localized values, a higher value could be used.

Soil Analysis

This section has the solution procedure for fluid-soil analysis. In the current formulation, it is assumed that the fluid is of a single phase, and only slightly compressible. This formulation will not be adequate if steam-fluid-solid analysis is required. The dry soil can be modeled using one of the three models: linear elasticity, nonlinear elasticity and the modified Cam-Clay model.

There are three types of soil analysis available in MSC.Marc. In the first type, you perform an analysis to calculate the fluid pressure in a porous medium. In such analyses, heat transfer elements [41](#), [42](#), or [44](#) are used. The **SOIL** model definition option is used to define the permeability of the solid and the bulk modulus and dynamic viscosity of the fluid. The porosity is given either through the **INITIAL POROSITY** or the **INITIAL VOID RATIO** options and does not change with time. The prescribed pressures can be defined using the **FIXED PRESSURE** option, while input mass flow rates are given using either the **POINT FLUX** or **DIST FLUXES** option.

In the second type of soil analysis, the pore (fluid) pressure is directly defined, and the structural analysis is performed. Element types [27](#), [28](#), or [21](#) are available. In such analyses, the pore pressure is prescribed using the **INITIAL PORE** and **CHANGE PORE** options. The characteristics of the soil material are defined using the **SOIL** option. If an elastic model is used, the Young's moduli and Poisson's ratio are important. If the Cam-Clay model is used, the compression ratios and the slope of the critical state line is important. For the Cam-Clay model, the preconsolidation pressure is defined using the **INITIAL PC** option. For this model, it is also important to define an initial (compressive stress) to ensure a stable model.

In the third type of soil analysis, a fully-coupled approach is used. Element types [32](#), [33](#), or [35](#) are available. These elements are "Herrmann" elements, which are conventionally used for incompressible analysis. In this case, the extra variable represents the fluid pressure. The **SOIL** option is now used to define both the soil and fluid properties. The porosity is given through the **INITIAL POROSITY** or the **INITIAL VOID RATIO** option. The prescribed nodal loads and mass flow rates are given through the **POINT LOAD** option, while distributed loads and distributed mass flow rates are given through the **DIST LOADS** option. The **FIXED DISP** option is used to prescribe either nodal displacements or pore pressures.

Technical Formulation

In soil mechanics, it is convenient to decompose the total stress σ into a pore pressure component p and effective stress σ' .

$$\sigma = \sigma' - pI \quad (5-185)$$

Note the sign convention used; a positive pore pressure results in a compressive stress. The momentum balance (equilibrium) equations are with respect to the total stresses in the system.

$$\nabla \sigma + f = \rho \ddot{u} \quad (5-186)$$

where ρ is the density, and f , $\ddot{u}$ are the body force and the acceleration. The equilibrium equation can then be expressed as

$$\nabla \sigma' - \nabla p + f = \rho \ddot{u} \quad (5-187)$$

The fluid flow behavior can be modeled using Darcy's law, which states that the fluid's velocity, relative to the soil's skeleton, is proportional to the total pressure gradient.

$$\dot{u}_f = \frac{-K}{\mu} (\nabla p + \rho_f g) \quad (5-188)$$

where

$\dot{u}_f$ is the fluid's bulk velocity

K is the soil permeability

μ is the fluid viscosity

ρ_f is the fluid density

g is the gravity vector.

The fluid is assumed to be slightly compressible.

$$\dot{p} = K_f \frac{\dot{\rho}_f}{\rho_f} \quad (5-189)$$

where K_f is the bulk modulus of the fluid.

However, the compressibility is assumed small enough such that the following holds:

$$\nabla \cdot \left[\rho_f \frac{K}{\mu} (\nabla p + \rho_f g) \right] \approx \rho_f \nabla \cdot \left[\frac{K}{\mu} (\nabla p + \rho_f g) \right] \quad (5-190)$$

It is also assumed that the bulk modulus of the fluid is constant, introducing the fluid's compressibility β_f .

$$\beta_f = 1/K_f \quad (5-191)$$

The solid grains are assumed to be incompressible. Under these assumptions, the governing equations for fluid flow is

$$\nabla \cdot \left[\frac{K}{\mu} (\nabla p + \rho_f g) \right] = \phi \beta_f \dot{p} + \nabla \dot{u} \quad (5-192)$$

where σ is the medium's porosity.

It is important to note that the medium's porosity is only dependent upon the original porosity and the total strains. Letting V_f and V_s stand for the fluid and solid's volume

$$\phi = dV_f / (dV_f + dV_s) = 1 - J^{-1}(1 - \phi_0) \quad (5-193)$$

where J is the determinant of the deformation gradient and ϕ_0 is the original porosity, both with respect to the reference configuration.

Introducing the weighting function η_u and η_p , the weak form, which is the basis for the finite element system, then becomes

$$\int_{V_{n+1}} [\eta_u \dot{f} - \rho \eta_u \ddot{u} - \nabla \eta_u \cdot \sigma' + \nabla \eta_u p] dv + \int_A \eta_u t da = 0 \quad (5-194)$$

where V and A are the conventional volumes and surface area and t is the applied tractions. Note that the applied tractions is the combined tractions from both the effective stress σ' and the pore pressure p .

and

$$\int_{V_{n+1}} \left[\nabla \eta_p \cdot \frac{K}{\mu} (\nabla p + \rho_f g) + \phi \beta_f \eta_p \dot{p} + \eta_p (\nabla \dot{u}) \right] dv - \int_{A_{n+1}} \eta_p q_n da = 0 \quad (5-195)$$

where the normal volumetric inflow, q_n , is:

$$q_n = \frac{k}{\mu} (\nabla p + \rho_f g) \cdot n \quad (5-196)$$

The weak form of equilibrium can be written as:

$$r_u = \int_{V_{n+1}} [\nabla \cdot \eta_u p] dv \quad (5-197)$$

$$r_p = \int_{V_{n+1}} \left[\nabla \eta_p \cdot \frac{K}{\mu} (\nabla p + \rho_f g) + \phi \beta_f \eta_p \dot{p} + \eta_p \nabla \cdot \dot{u} \right] dv \quad (5-198)$$

Application of the directional derivative formula yields:

$$D_u(r_u) \cdot \delta u = \left[\frac{d}{d\varepsilon} r_u(u + \varepsilon \delta u) \right]_{\varepsilon=0} \quad (5-199)$$

Hence,

$$D_u(r_u) \cdot \delta u = \int_{V_{n+1}} [\nabla \cdot \Delta u (\nabla \cdot \eta_u p) + \nabla \eta_u^T \cdot \nabla \Delta u p] dv \quad (5-200)$$

Similarly,

$$D_p(r_u) \cdot \delta p = \int_{V_{n+1}} \nabla \cdot \eta_u \Delta p dv \quad (5-201)$$

$$D_u(r_p) \cdot \delta u = \int_{V_{n+1}} \left\{ \left[\nabla \eta_p \cdot \frac{K}{\mu} (\nabla p + \rho_f g) + \phi \beta_f \eta_p \dot{p} + \eta_p \nabla \cdot \dot{u} \right] \nabla \cdot \Delta u - \eta_p \nabla \dot{u}^T \cdot \right. \quad (5-202)$$

$$\left. \nabla \Delta u + \eta_p \nabla \cdot \Delta \dot{u} - \nabla \Delta u \nabla \eta_p \cdot \frac{K}{\mu} (\nabla p + \rho_f g) - \nabla \eta_p \cdot \frac{K}{\mu} (\nabla \Delta u) \nabla p \right\} dv$$

$$D_p(r_p) \cdot \delta p = \int_{V_{n+1}} \left[\nabla \eta_p \cdot \frac{K}{\mu} \nabla \Delta p + \phi \beta_f \eta_p \Delta \dot{p} \right] dv \quad (5-203)$$

with the displacement and pressures interpolated independently as:

$$\Delta u = \sum N_i \Delta u_i$$

and

$$p = \sum N_j p_j$$

(5-204)

we get a linearized system of equilibrium equation,

$$\begin{bmatrix} K_{uu} & K_{up} \\ K_{pu} & K_{pp} \end{bmatrix} \begin{Bmatrix} \delta u \\ \delta p \end{Bmatrix} = F - \begin{Bmatrix} R_u \\ R_p \end{Bmatrix} \quad (5-205)$$

The resulting system of equations is highly nonlinear and nonsymmetric, and is solved by full Newton-Raphson solution scheme.

Note that it is assumed that the permeability, porosity, viscosity, and the bulk modulus of the fluid are considered independent of the state variables u and p . It is evident that, in general, this is not the case; however, in the analysis that follows, these dependencies are ignored for tangent purposes. Note that they are included in the calculation of the residuals R_u and R_p ; hence, convergence is always achieved at the true solution.

Three types of analyses can be performed. The simplest is a solution for only the fluid pressure based upon the porosity of the soil. In this case, a simple Poisson type analysis is performed and element types 41, 42, or 44 are used. In the second type of analysis, the pore pressures are explicitly defined and the structural analysis is performed. In this case, the element types 21, 27, and 28 should be used. Finally, a fully-coupled analysis is performed; in which case, you should use element types 32, 33, or 35. Of course, the soil can be combined with any other element types, material properties to represent the structure, such as the pilings.

Design Sensitivity Analysis

Design sensitivity analysis is used to obtain the sensitivity of various aspects of a design model with respect to changes in design parameters in order to facilitate structural modifications. The design parameters that are amenable to change are called “design variables”. The two major aspects of the design model for which design sensitivity is considered herein are:

- a. Design objective
- b. Design model response

As a result, the design sensitivity analysis capability in MSC.Marc is currently limited to finite element models of structures with linear response in the computation of

1. Gradients for
 - a. An objective function (or the design objective), if one is defined by you (for example, minimizing the material volume in the model).
 - b. Various types of design model responses under multiple cases of static mechanical loading, or free vibration, in linear behavior.
2. Element contributions to the responses of the model.

The gradient of the objective function or of a response function is simply the set of derivatives of such a function with respect to each of the design variables, at a given point in the design space (that is, for a given design).

For sensitivity analysis to proceed, the design model, the analysis requirements, the design variables, and the functions for which the gradients are to be found have to be specified by you.

The existing design sensitivity analysis capability in MSC.Marc can be applied in one of two ways:

1. As a stand-alone feature, where you are concerned only with obtaining sensitivity analysis results. Such an application is completed with the output of the sensitivity analysis results.
2. Within a design optimization process, where you are concerned mainly with the optimization of a design objective related to a finite element model. This type of an application of sensitivity analysis is transparent to you. The design optimization process is completed with the output of design optimization related data, such as the optimized objective function, related values of the design variables, and the analysis results for the optimized design.

These two procedures are described below.

1. Sensitivity analysis as a stand-alone feature

This type of solution usually aims at obtaining the derivatives of user prescribed response quantities at a given design, with respect to each of multiple design variables specified by the user. This set of derivatives is therefore the gradient of the response function at the given design in the design variable space (or, in short, in the design space). For example, for a prescribed response function r , given the design variables x_1 , x_2 , and x_3 , the gradient is defined as

$$\nabla_{\tilde{r}} = \left[\frac{dr}{dx_1} \quad \frac{dr}{dx_2} \quad \frac{dr}{dx_3} \right]^T \quad (5-206)$$

The number of response quantities for which gradients are computed is limited either by the program default or by a user-specified number. If you are interested in obtaining the sensitivity analysis results in order of criticality, the option to sort them in this order is also available. The responses are currently prescribed as constraints with user-defined bounds. If sorting is not required, the bounds can be mostly arbitrary, although they still have to conform to the type of constraint prescribed. However, if sorting is required and is to be meaningful, it is important for you to give realistic bounds on the response.

Element contributions to each response quantity are obtained as a by-product of the type of response sensitivity analysis capability in MSC.Marc. Thus, the response r can be represented as a sum of these element contributions:

$$r = \sum_{e=1}^N r_e + C \quad (5-207)$$

where the second term, C , involves work done elsewhere, such as in support settlement, if any.

This is helpful for a visual understanding of which regions of the model contribute in what manner to each of the response quantities at the given design, since it can be plotted in a manner similar to, say stress contours.

Finally, as an option, if you also prescribe an objective function, the gradient of the objective function with respect to the design variables is also computed at the given design. Thus, for the objective function W , MSC.Marc obtains

$$\nabla_{\tilde{W}} = \left[\frac{dW}{dx_1} \quad \frac{dW}{dx_2} \quad \cdots \quad \frac{dW}{dx_n} \right]^T \quad (5-208)$$

2. Sensitivity analysis within a design optimization process:

The design optimization algorithm in MSC.Marc requires the utilization of gradients of the objective function and of the constraint functions, which are very closely related to the response functions. The current algorithm, described under “design optimization”, ignores your initial prescribed design, but instead begins by generating other designs within the prescribed bounds for the design variables. Once the optimization algorithm is completed and the optimized design is available, if a sensitivity analysis is required at the optimized state, it will be necessary for you to modify the model accordingly and to use sensitivity analysis as a stand alone feature.

During design optimization, sensitivity analysis is performed for a maximum number of constraints either indicated by the program default or prescribed by you.

Theoretical Considerations

The method currently employed in MSC.Marc for response sensitivity analysis is the “virtual load” method. For sensitivity analysis of the objective function, finite differencing on the design variables is performed directly.

In the virtual load method, first a design is analyzed for the user-prescribed load cases, and, if also prescribed, for eigenfrequency response. The response of the structure having been evaluated for each of these analyses, the response quantities for which sensitivity analysis is to be performed are then decided upon and collected in a database. In sequence, a virtual load case is generated for each such response quantity. Reanalysis for a virtual load case leads to virtual displacements.

The principle of virtual work is then invoked. This defines the element contributed part of the response quantity, for which the virtual load was applied, as a dot product of the structural displacement vector, for the actual load case with which the response is associated, and the virtual load vector itself.

The j^{th} response r_j can be expressed as the dot product of the actual load vector with the virtual displacement vector as

$$r_j = p^T u_v \quad (5-209)$$

By differentiation of this expression, you can show that the derivative of the response r_j with respect to a given design variable x_i is given by

$$\frac{dr_j}{dx_i} = \frac{dp_v^T}{dx_i} u + u_v^T \left(\frac{dp}{dx_i} - \frac{dK}{dx_i} u \right)$$

where K is the stiffness matrix of the structure. The response derivative above is evaluated on the element basis as:

$$\frac{dr_j}{dx_i} = \sum_e \left[\frac{dp_v^T}{dx_i} u + u_v^T \left(\frac{dp}{dx_i} - \frac{dK}{dx_i} u \right) \right]_e \quad (5-210)$$

where the vectors u and p_v are the vector of element nodal displacements due to the actual load case and the vector of element nodal forces due to the virtual load, respectively. The case of eigenfrequencies follows the same logic except that an explicit solution for the virtual load case is not necessary. The derivatives are now evaluated at the element level via finite differencing. This is known as the semi-analytical approach.

Note that the derivative expression for the virtual load method is quite similar to that for the Adjoint Variable method. In fact, while they are conceptually different approaches, for certain cases they reduce to exactly the same expressions. However, for certain other cases, the terms take on different meanings although the end result is the same.

Design Optimization

Design optimization refers to the process of attempting to arrive at certain ideal design parameters, which, when used within the model, satisfy prescribed conditions regarding the performance of the design and at the same time minimize (or maximize) a measurable aspect of the design. In MSC.Marc, you can ask to minimize

1. total material volume
2. total material mass
3. total material cost.

When there is more than one material in the finite element model, the specification of different mass densities and unit costs are taken into account in the computation of the objective function.

The performance requirements might not necessarily have to be related to response, but also to different concepts such as packaging, design envelope, even maintenance. The current capability is based on optimization with constraints on response. Also, the lower and upper bounds on the design variables themselves define the limits of design modifications. Hence, the design optimization problem can be posed mathematically in the following format:

$$\begin{array}{ll} \text{Minimize} & W \\ \text{Subject to} & c_j \geq 0.0 \quad j = 1, \dots, m \end{array}$$

where W is the objective function, and c_j is the j th constraint function, either specified as an inequality related to a response quantity or as a limit on a design variable. For example, to limit the x-direction translation component at a node k , the constraint can first be written as

$$(u_x)_k \leq u_{xk}^*$$

Assuming that the displacement is constrained for positive values, the normalized constraint expression c (dropping the subscript j) becomes:

$$(u^* - u)/u^* \geq 0.0 \tag{5-211}$$

with its derivatives as:

$$\frac{dc}{dx} = \frac{-1}{u^*} \frac{du}{dx} \tag{5-212}$$

Within MSC.Marc, the constraints can be imposed on strain, stress, displacement, and eigenfrequency response quantities. For stresses and strains, the constraints are defined as being on the elements, and for displacements, the constraints are defined as being at nodes. Stress and strain components, as well as various functions of these components (the von Mises equivalent stress and principal stresses, stresses on prescribed planes) and generalized stress quantities can be constrained. Similarly, translation and rotation components of displacement, resultant and directed displacements as well as relative displacements between nodes can be constrained. For free vibration response, constraints can be placed on the modal frequencies as well as on differences between modal frequencies. A full listing of such constraints are given in [MSC.Marc Volume C: Program Input](#).

The upper and lower bounds on the design variables are posed as

$$x_i \leq x_u \quad \text{and} \quad x_i \geq x_l \quad (5-213)$$

after which they can be transformed into expressions similar to [Equation 5-211](#).

The response quantities associated with the model are implicit functions of the design variables. Analyses at sample design points are used to build explicit approximations to the actual functions over the design space. This approach minimizes the number of full scale analyses for problems which require long analysis times such as for nonlinear behavior. This method is summarized next.

Approximation of Response Functions Over the Design Space

The design space for the optimization problem is bounded by limits on the design variables of a model to be optimized. The simplest case is that of a single design variable, where the design space is a straight line, bounded at the two ends. For higher number of design variables, say n , the design space can be visualized as a bounded hyperprism with n^2 vertices. For such a construct, you can build approximations to the constraint functions by way of analyses conducted at the vertices. However, this requires n^2 analyses.

We now note that the minimum geometrical construct spanning n -dimensional space is a simplex with $n + 1$ vertices. Thus, an approximation based on analyses at vertices requires only $(n + 1)$ analyses. The simplex has already been used for first order response surface fitting based on only function values [\[Ref. 3\]](#). However, the use here involves higher order response functions.

Like the hyperprism, in one-dimension, the simplex degenerates into a straight line. However, in two dimensions it is a triangle, and in general it is a hyper-tetrahedron. [Figure 5-21](#) compares the simplex to the hyperprism in normalized two-dimensional design space.

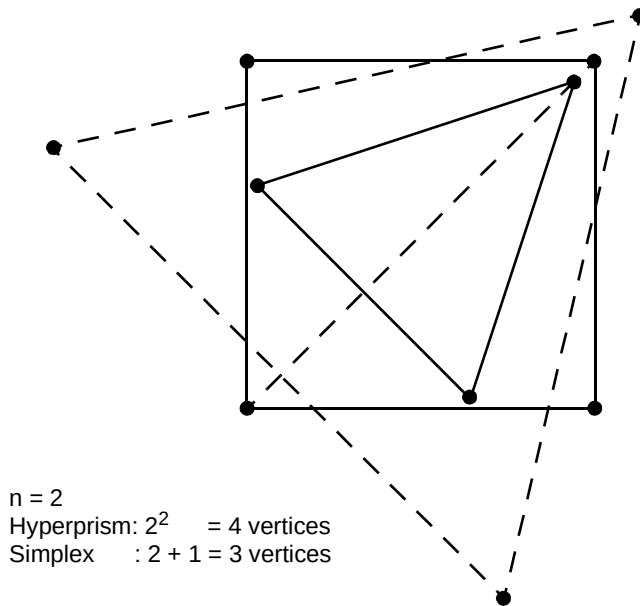


Figure 5-21 Comparison of the Simplex to the Hyperprism in Two-dimensions

While the orientation of the simplex in the design space appears to be a relatively arbitrary matter, once an origin and the size of the simplex are prescribed, a simple formula will locate all vertices of a simplex in n-dimensions [\[Ref. 4\]](#). The response gradient information at the simplex vertices is combined with the function values to achieve enhanced accuracy. Thus, an analysis at a vertex can be utilized to yield both response function values and, by way of sensitivity analysis, the response gradients at that vertex. The virtual load method employed in MSC.Marc for obtaining the response gradients is discussed under [“Design Sensitivity Analysis”](#). The response gradients can then easily be converted to constraint gradients for use in an optimization algorithm.

As a result, the results of an analysis at a vertex of the simplex can be summarized as the vector of constraint function values c_j , and the gradient, ∇c_j , of each constraint function (j) at that vertex.

For the case of a one variable problem, the results of analyses at the two vertices are symbolized in [Figure 5-22](#), for a hypothetical constraint c_j .

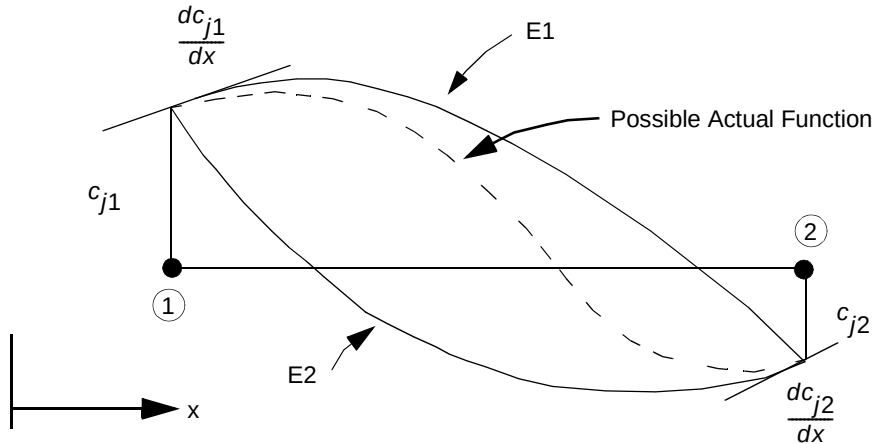


Figure 5-22 Vertex Results for One-dimensional Design Space

From [Figure 5-22](#), it appears almost natural to fit a cubic function to the four end conditions (two function values and two slopes) depicted in the figure. However, this approach is too rigid, and is not easily generalizable to higher dimensions. On the other hand, the use of two quadratics, which are then merged in a weighted manner gives higher flexibility and potential for increased accuracy.

It can be seen that the two equations, $E1$ and $E2$, are designed such that they both satisfy the function values at the two vertices, but $E1$ satisfies the slope at vertex 1 only, and $E2$ satisfies the slope at vertex 2 only. Finally, at any design point x , the response function c_j can be approximated as

$$c_j = (W1 \ E1 + W2 \ E2)/(W1 + W2) \quad (5-214)$$

where the weight functions $W1$ and $W2$ are normally functions of x and possibly some other parameters.

This type of approach has the further advantage that it is consistent with the use of the simplex for determining analysis points and approximating constraint functions in the higher dimensional cases. Therefore, for an n -dimensional problem, the simplex

having $n + 1$ vertices, each equation needs to have $(n + 1) + n = 2n + 1$ unknown parameters. The general quadratic polynomial without the cross-coupling terms satisfies this condition for all n .

Improvement of the Approximation

When approximations are used, the results of an optimization process need to be checked by means of a detailed analysis. As a result, the approximations can be adjusted and the optimization algorithm can be reapplied. Depending on how accurate the approximations prove to be and how many more detailed analyses are acceptable to you, this process can be applied for a number of cycles in order to improve upon the results.

The Optimization Algorithm

The optimization algorithm implemented in MSC.Marc is the Sequential Quadratic Programming method [Ref. 5]. This method is employed using the approximation described above. By obtaining response function and gradient values from the approximate equations, the need for full scale analyses is reduced. The method is summarized below.

The quadratic programming technique is based on the approximation of the objective function by a quadratic function. When nonlinear constraints exist, as is the case in most practical design optimization problems, the second order approximation concept is extended to the Lagrangian which is a linear combination of the objective function and the constraint functions. The solution method for a quadratic programming problem with nonlinear constraints can be summarized as the following:

At each step, modify the design variables vector x by adding a scaled vector

$$x \leftarrow x + qs \quad (5-215)$$

where s is a search direction and q is the scale factor along the search direction. If the search direction has been determined, the scale factor can be found by some type of line search along the search direction. The determination of the search direction constitutes the major undertaking in the quadratic programming method. If H is the Hessian of the Lagrangian and g is the gradient of the objective function, then the search direction s is that which minimizes the function

$$g^T s + (s^T H s)/2 \quad (5-216)$$

subject to the linearized constraints

$$J_s \geq -c \quad (5-217)$$

where J is the Jacobian matrix of the constraints and c is the vector of constraint functions at the current iteration. Due to lack of knowledge about which constraints will be active at the optimum, the Hessian of the Lagrangian is not always readily available. Thus, some iterations take the form of a gradient projection step.

The coefficients of the constraint functions in the Lagrangian are the Lagrange multipliers which are unknown before solution has started. At the optimum these multipliers are zero for inactive constraints. Normally, the above problem is solved using only those constraints which appear to be active at or close to the current design point, with the assumption that these constraints will be active at the optimum. The selection of these active constraints is done within the framework of an active set strategy, the set being modified appropriately with the progression of the iterations. Similarly, the arrays g , H , and J are also modified as the iterations proceed.

MSC.Marc User Interface for Sensitivity Analysis and Optimization

This is discussed below under input, output, and postprocessing.

Input

Input features related to design sensitivity and design optimization are similar. However, they differ in the parameter data blocks and in the optional specification of an objective function for the case of sensitivity analysis. Therefore, other than these two items, all discussion of input should be construed to refer to both procedures equally. All design sensitivity and design optimization related parameters and options in a MSC.Marc input file start with the word `DESIGN`.

All load cases and any eigenfrequency analysis request associated with sensitivity analysis or optimization should be input as load increments in the history definition section after the `END OPTION` line. The maximum number of nodes with point loads or the maximum number of distributed load cases should be defined in the `DIST LOADS` parameter.

The type of solution requested from MSC.Marc can simply be indicated by the parameter input `DESIGN SENSITIVITY` or `DESIGN OPTIMIZATION`. Only one of these lines can appear in the input. These parameters also let you change the sorting of constraints, the maximum number of constraints in the active set, and the maximum number of cycles of design optimization. The first two items are discussed in previous sections. A cycle of design optimization refers to a complete solution employing the sequential quadratic programming technique followed by a detailed finite element

analysis to accurately evaluate the new design point reached by way of approximate solution. If required, the approximation is improved using the results of the last finite element analysis, and a new cycle is started.

The specification of an objective function, being optional for design sensitivity, is made by use of the model definition option **DESIGN OBJECTIVE**. This allows you to choose from one of the available design objectives.

The **DESIGN VARIABLES** option allows you to specify the design variables associated with the finite element model. You have a choice of three major types of design variables:

1. Element cross-sectional properties as given by way of the **GEOMETRY** option.
2. Material properties as given by **ISOTROPIC** or **ORTHOTROPIC** options.
3. Composite element properties of layer thickness and ply angle, as given by the **COMPOSITE** option.

The properties which are supported are listed under the **DESIGN VARIABLES** option. The relevant elements, for which cross-sectional properties can be specified as design variables, each has a section in *Volume B: Element Library*, describing which properties, if any, can be posed as design variables for that element.

Similarly, the list of material properties (currently all related to linear behavior) that can be design variables is given under the **DESIGN VARIABLES** option. For composite groups, the layer thicknesses and ply angles can be given as design variables for composite groups.

Design variables can be linked across finite element entities such that a given design variable controls several entities. An example is the linking of the thicknesses of several plane stress elements by means of a single design variable. Thus, when this variable changes, the thicknesses of all linked elements reflect this change. On the other hand, for the unlinked case, all thicknesses are associated with separate design variables. This feature is controlled by the **LINKED** and **UNLINKED** commands.

Response quantities, or constraints on response quantities, are specified by means of the **DESIGN DISPLACEMENT CONSTRAINTS**, **DESIGN STRAIN CONSTRAINTS**, **DESIGN STRESS CONSTRAINTS**, and **DESIGN FREQUENCY CONSTRAINTS** options. There is no limit to the number of constraints. Displacement constraints are posed at nodes or groups of nodes, while the strain and stress constraints are posed over elements or groups of elements, and frequency constraints are posed for free vibration modes. A complete list can be found under the above mentioned options in *MSC.Marc Volume C: Program Input*.

When strain or stress constraints are prescribed, it is useful to know that the program evaluates such constraints at all integration points of all layers of an element and proceeds to consider the most critical integration point at the related layer for the element. Thus, a strain or stress constraint on an element normally refers to the most critical value the constraint can attain within the element. During optimization, the most critical location within the element may change and any necessary adjustment takes place internally.

For certain responses, the limiting values can be the same in absolute value for both the positive and negative values of the response. For constraints on such response functions, you have the choice of prescribing either separate constraints on the positive and negative values, or a combined constraint on the absolute value. The first approach is more accurate albeit at a higher computational cost.

Output

For sensitivity analysis, the output file contains the following information:

- Echo of input, any warnings or error messages.
- A of numbers and definitions for your prescribed design variables. It is important to note that the variable values are always output in the internal numbering sequence which is defined in this. Search for the words ‘Design variable definitions’ to reach this in the output file.
- Analysis results for your prescribed design.
- The value of the objective function and its gradient with respect to the design variables prescribed by you.
- Sensitivity analysis results for the response functions in the default or user-defined set, sorted in order of criticalness (if specified). It should be noted that although the responses are sorted across multiple load cases, the sensitivity results are output for each load case. The related output file consists of a check on the actual response value (obtained by sensitivity analysis) and the gradient of the response with respect to the design variables. For eigenfrequency results, the check values on the response can be somewhat more accurate than the results from eigenfrequency analysis since the latter is iterative, but the check uses the Rayleigh quotient on top of the iteration results. A constraint reference number allows you to track the sensitivity analysis results plots when postprocessing. Search for the word ‘Sensitivity’ to reach this output.

For design optimization, the following is written into the output file:

- Echo of input, any warnings or error messages.
- Certain indications that some analyses are being done, but no analysis results except for the ‘best’ design reached during optimization.
- The objective function values and the vector of design variable values:

- At each of the simplex vertices.
 - For the starting simplex vertex (from this point on, the information also includes whether the design is feasible or not).
 - At the end of each cycle of optimization.
 - For the ‘best’ design found.
- Analysis results for the ‘best’ design.

Postprocessing

This requires that you ask the program to create a post file. The following plots can be obtained by way of postprocessing.

For sensitivity analysis:

- Bar charts for gradients of response quantities with respect to the design variables.
- Contour plots of element contributions to response quantities. Thus, a finite element model contour plot gives the element contributions to a specific response quantity which was posed in the form of a constraint in the input file.

The increment number of the sensitivity analysis results is the highest increment number available in the post file. The information for each response quantity is written out as for a subincrement. The zeroth subincrement corresponds to the objective function information. Numbers for the other subincrements correspond to the constraint reference number(s) in the output file.

For design optimization:

- Path plots showing the variation of the objective function and of the design variables over the history of the optimization cycles. The best design (feasible or not) is not necessarily the last design point in the plot. The values at the starting vertex are considered as belonging to the zeroth subincrement. Each optimization cycle is then another subincrement. The increment number corresponding to these subincrements is taken as zero. The analysis results for the ‘best’ design start from increment one.
- Bar charts where each chart gives the values of the design variables at the optimization cycle corresponding to that bar chart.

Transfer Axisymmetric Analysis Data to 3-D Analysis

In many cases, it is possible to begin the numerical simulations as a two-dimensional axisymmetric problem even though the final problem is fully three-dimensional. This is advantageous because of the large computational savings. For this to be useful, the first stage of the problem should be truly axisymmetric. The second stage of the problem can be fully three-dimensional. The [AXITO3D](#) model definition option is used in the input file of the 3-D problem to transfer results from an axisymmetric analysis into a 3-D analysis. The data from the axisymmetric analysis is stored on the post file. In the 3-D analysis, the results from axisymmetric analysis are used as initial conditions.

There are three steps in performing an axisymmetric to 3-D analysis.

1. Run axisymmetric analysis.
2. Expand axisymmetric model to 3-D model and transfer data from axisymmetric model to 3-D model. (Axisymmetric model file is used for expansion of loads and boundary conditions besides the mesh. Exception is remeshing where the model file is not used.)
3. Run 3-D analysis.

Most quadrilateral axisymmetric elements (in 3-D case, hexahedral elements), including Herrmann elements, and most available materials, such as metal and rubber, can be used in the axisymmetric to 3-D analysis. For problems involving large deformation, either total or updated Lagrangian formulation must be used. Thermal and dynamic effects are considered.

For more detailed information on this feature, see the model definition option, [AXITO3D](#), in *MSC.Marc Volume C: Program Input*.

Load and Displacement Boundary Conditions Transfer from Axisymmetric Analysis to 3-D using Curve Shift

Because the externally applied loads and displacement boundary conditions are not fully available in the post file, it is, generally, not possible to complete an [AXITO3D](#) analysis based solely on the information in the post file. MSC.Marc Mentat is used to achieve the load and displacement boundary conditions data transfer from axisymmetric to 3-D analysis by means of shifting load curves.

The entire history of the load and the displacement boundary conditions for a complete axisymmetric to 3-D analysis must be defined using the curves. Once the axisymmetric part of the analysis is finished, MSC.Marc Mentat is used to expand the axisymmetric model to a 3-D model. During the model expansion, you are required to

define if the curves need to be shifted and how much the curves are to be shifted. For this purpose, either an increment number (converts internally to time) for the axisymmetric analysis or time value (which is the starting point of the 3-D analysis) has to be defined. If the curve shift is performed, the increment number or time value is defined under:

MESH GENERATION→EXPAND→AXISYMMETRIC MODEL TO 3D.

This definition is copied to:

INITIAL CONDITIONS→MECHANICAL→AXISYMMETRIC TO 3D

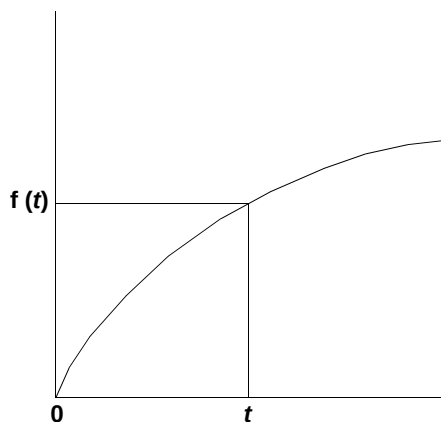
and no modification is allowed. If the curve shift is not performed, the increment number or time value is defined under:

INITIAL CONDITIONS→MECHANICAL→AXISYMMETRIC TO 3D

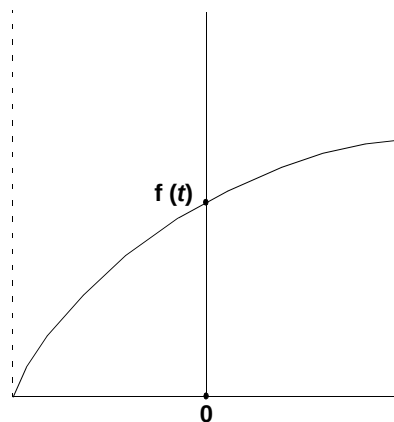
After the curve shift is completed, you can delete or modify existing curves or add new curves depending on the specific requirement.

The curves used with external loads are shifted differently from those associated with displacement boundary conditions. Examples are shown in [Figure 5-23](#). The curve is A is a typical curve used to define load or displacement boundary conditions, where t is the time at which the 3-D analysis starts. If the curve in A is associated with external loads, B shows the shifted curve, where the total load at the end of the axisymmetric analysis is applied at the start of the 3-D analysis. If the curve in A is associated with displacement boundary conditions, C shows the shifted curve, where the new curve for 3-D analysis starts at the zero.

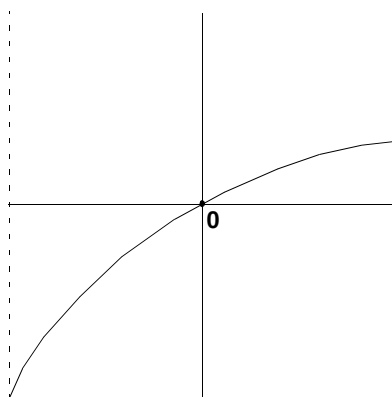
If r- or z-direction point load or prescribed displacement is specified at a node in the axisymmetric model, local coordinate systems are automatically defined by MSC.Marc Mentat for all nodes obtain from 3-D expansion of the node in the axisymmetric model. It is recommended that these local coordinate systems be checked after the expansion is completed to be certain they are properly defined and needed.



A. Typical Curve used to Define Either Load or Displacement Boundary Conditions in Axisymmetric Analysis



B. Shifted Curve in 3-D Analysis if Curve in **A** is associated with External Load



C. Shifted Curve in 3-D Analysis if Curve in **A** is associated with Displacement Boundary Conditions

Figure 5-23 Curve Shift in Axisymmetric to 3-D Analysis

Steady State Rolling Analysis

Rolling contact analysis of a cylindrical deformable body in a Lagrangian framework can be computationally expensive because it may require not only time-dependent transient process but also a fine mesh in the entire body to accurately capture contact characteristics. However, some problems involve only fully axisymmetric structures with constant moving/spinning velocities. These problems can be considered steady state if a reference configuration, which moves with the body but does not spin around the rolling axis, is used.

MSC.Marc provides the capability of steady state rolling analysis. The feature is characterized by a mixed Eulerian/Lagrangian formulation with inertia effects in spinning/cornering deformable bodies. Using a non-spinning reference frame attached to the wheel axel, the analysis becomes purely space dependent. It presents a better alternative to the unnecessary computational burden of arriving at a steady state condition through a transient analysis. Furthermore, a finer mesh only needs to be used in the contact region as opposed to the entire rolling surface.

Kinematics

We consider the axisymmetric body shown in [Figure 5-24](#). The body spins at an angular velocity ω_s around the axisymmetric axis T_s at point P_s and, simultaneously rotates with a cornering angular velocity ω_c around an axis T_c at point P_c . Assume that a particle in the body has a location P_0 at time $t = 0$. At time t , its motion contains three parts:

1. from P_0 to a location X because of the spinning
2. from X to Y because of a deformation $Y = D(X)$, where D is time independent function resulting from the steady state condition
3. from Y to Z because of the cornering.

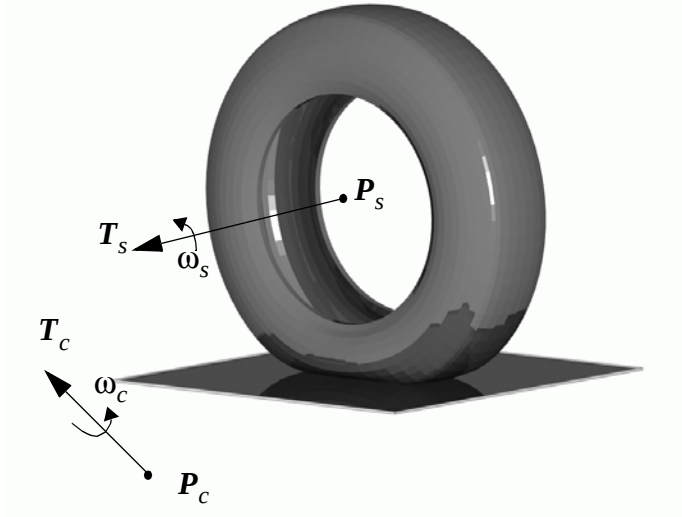


Figure 5-24 Kinematics

The three motions can be described as

$$X = R_s \cdot (P_o - P_s) + P_s \quad (5-218)$$

$$Y = D(X) \quad (5-219)$$

$$Z = R_c \cdot (Y - P_c) + P_c \quad (5-220)$$

with

$$R_s = \exp(\bar{\omega}_s t) \text{ and } R_c = \exp(\bar{\omega}_c t) \quad (5-221)$$

In Equation 5-221, ω_s and ω_c are the skew-symmetric tensors associated with the rotation vectors $\omega_s T_s$ and $\omega_c T_c$, respectively, with

$$\bar{\omega}_s \cdot r = \omega_s T_s \times r \text{ and } \bar{\omega}_c \cdot r = \omega_c T_c \times r \quad (5-222)$$

for any vector r .

Time derivative of Equation 5-221 gives

$$\dot{R}_s = \bar{\omega}_s \cdot R_s \text{ and } \dot{R}_c = \bar{\omega}_c \cdot R_c \quad (5-223)$$

Making use of Equations 5-218, 5-219, 5-222, and 5-223, the velocity of the particle can be obtained by the first time derivative of Equation 5-220 as

$$\dot{Z} = R_c \cdot \left[\omega_c T_c \times (Y - P_c) + \omega_s \frac{\partial D}{\partial \alpha} \right] \quad (5-224)$$

where $\alpha = \omega_s t$, is the spinning angle.

Similarly, the acceleration is obtained by the second derivative of Equation 5-220 with respect to time:

$$\ddot{Z} = R_c \cdot \left[\omega_c^2 (T_c \otimes T_c - 1) \cdot (Y - P_c) + 2\omega_s \omega_c T_c \times \frac{\partial D}{\partial \alpha} + \omega_s^2 \frac{\partial^2 D}{\partial \alpha^2} \right] \quad (5-225)$$

where $\otimes$ denotes the tensor product and 1 is the unit tensor.

Transformation of Equations 5-224 and 5-225 into the reference configuration defined by X by premultiplying R_c^T , gives

$$v = \omega_c T_c \times (Y - P_c) + \omega_s \frac{\partial D}{\partial \alpha} \quad (5-226)$$

$$a = \omega_c^2 (T_c \otimes T_c - 1) \cdot (Y - P_c) + 2\omega_s \omega_c T_c \times \frac{\partial D}{\partial \alpha} + \omega_s^2 \frac{\partial^2 D}{\partial \alpha^2} \quad (5-227)$$

where v and a are velocity and acceleration of the particle with respect to the reference frame.

Inertia Effect

The contribution of inertia effect into the right-hand-side of the system equation can be calculated using the weak form

$$\begin{aligned} \delta \pi &= \int_V \rho a \cdot \delta u dV \\ &= -\rho \omega_c^2 \int_V (T_c \otimes T_c - 1) \cdot (Y - P_c) \cdot \delta u dV - 2\rho \omega_s \omega_c \int_V T_c \times \frac{\partial D}{\partial \alpha} \cdot \delta u dV \\ &\quad + \rho \omega_s^2 \int_V \frac{\partial D}{\partial \alpha} \cdot \frac{\partial \delta u}{\partial \alpha} dV \end{aligned} \quad (5-228)$$

where ρ is the density, V is the volume, and u is the displacement.

Linearization of Equation 5-228 gives

$$\begin{aligned} \Delta\delta\pi = & -\rho\omega_c^2 \int_V \Delta u \cdot (T_c \otimes T_c - 1) \cdot \delta u dV - 2\rho\omega_x\omega_c \int_V T_c \times \frac{\partial\Delta u}{\partial\alpha} \cdot \delta u dV \\ & + \rho\omega_s^2 \int_V \frac{\partial\Delta u}{\partial\alpha} \cdot \frac{\partial\delta u}{\partial\alpha} dV \end{aligned} \quad (5-229)$$

which can be used to calculate the contribution of the inertia effect to stiffness matrix.

Rolling Contact

To take into account the rolling effect in contact, the velocity vector in Equation 5-226 is decomposed into a normal and a tangential component, with respect to the contact surface, for all nodes in contact with ground. The normal component is then forced to become zero because of the contact conditions. The relative slipping velocity used in friction calculation is the difference between the tangential component and the ground moving velocity.

Steady State Rolling with MSC.Marc

The capability of steady state rolling analysis in MSC.Marc has taken into account the effects of rolling frictions and the inertia effects resulting from both spinning and cornering. The deformable rolling body can contact with multiple, flat or nonflat rigid surfaces. A typical example is a tire model which is in contact with a rigid rim and a rigid road surface. The spinning effects are involved only for selected contact body pair (for example, tire/road) which is defined with the **SS-ROLLING** history model definition option. The steady state rolling analysis can follow a static stress analysis with various loadcases and can also combine with any steady state loads. The feature is available for 3-D analysis only.

The element types supported for steady state rolling analysis include 7, 9, 18, 21, 35, 57, 61, 84, 117, 120, 146, 147, and 148.

The friction type supported in contact for steady state rolling analysis, based on nodal force, is Coulomb friction for rolling.

Because the system matrix becomes nonsymmetric in steady state rolling analysis, it is recommended that the multifrontal direct sparse solver be used.

The feature requires the so-called streamline along the circumferential direction of the spinning body. Therefore, the 3-D mesh must be generated by revolving its corresponding axisymmetric mesh. For a 3-D brick element, the circumferential direction must be from the element face defined by nodes 1-2-3-4 to the element face defined by nodes 5-6-7-8.

Below is a list of options required in a steady state analysis, in addition to the others in a standard static stress analysis.

SS-ROLLING: Parameter activating the steady state rolling analysis

ROTATION A: Model definition option defining the spinning axis

CORNERING AXIS: Model definition option defining the cornering axis

SS-ROLLING: History definition option defining the spinning body, ground body, spinning body motion (spinning/cornering/moving relative to the ground) and others. All velocities/forces defined in the option are total values.

References

1. Key, S. W. and R. D. Krieg, 1982, "On the Numerical Implementation of Inelastic Time-Dependent, Finite Strain Constitutive Equations in Structural Mechanics", *Computer Methods in Applied Mechanics in Engineering*, V.33, pp. 439–452, 1982.
2. Bathe, K. J., E. Ramm, and E. L. Wilson. "Finite Element formulation for Large Deformation Dynamic Analyses", *International Journal for Numerical Methods in Engineering*, V. 9, pp. 353–386, 1975.
3. Montgomery, D. C., Design and Analysis of Experiments, (2nd ed.) John Wiley and Sons, 1984.
4. Spendley, W., Hext, G. R., Himsworth, F. R., "Sequential Application of Simplex Designs in Optimisation and Evolutionary Operation", *Technometrics*, Vol. 4, No. 4, pp. 441-461 (1962).
5. Gill, P. E., Murray, W., Saunders, M. A., Wright, M. H., "Sequential Quadratic Programming Methods for Nonlinear Programming", NATO-NSF-ARO, Advanced Study Inst. on Computer Aided Analysis and Optimization of Mechanical System Dynamics, Iowa City, IA, Aug. 1-2, 1983, pp. 679-700.
6. Bathe, K. J. *Finite Element Procedures* Prentice-Hall, 1996.
7. MARC Reference Manual Volume F.
8. Wilkinson, J. H. *The Algebraic Eigenvalue Problem*. Oxford: Clarendon Press, 1965.
9. Zienkiewicz, O. C., and Taylor, L. C. *The Finite Element Method*. Fourth Ed., Vol. 1 & 2. London: McGraw-Hill, 1989.
10. Barsoum, R. S. "On the Use of Isoparametric Finite Elements in Linear Fracture Mechanics." *Int. J. Num. Methods in Engr.* 10, 1976.
11. Dunham, R. S. and Nickell, R. E. "Finite Element Analysis of Axisymmetric Solids with Arbitrary Loadings." No. 67-6. Structural Engineering Laboratory, University of California at Berkeley, June, 1967.
12. Hibbitt, H. D., Marcal, P. V. and Rice, J. R. "A Finite Element Formulation for Problems of Large Strain and Large Displacement." *Int. J. Solids Structures* 6 1069-1086, 1970.

13. Houbolt, J. C. "A Recurrence Matrix Solution for the Dynamic Response of Elastic Aircraft." *J. Aero. Sci.* 17, 540-550, 1950.
14. McMeeking, R. M., and Rice, J. R. "Finite-Element Formulations for Problems of Large Elastic-Plastic Deformation." *Int. J. Solids Structures* 11, 601-616, 1975.
15. MARC Update. "The Inverse Power Sweep Method in MARC." U.S. Ed., vol. 3, no. 1, February, 1984.
16. Simo, J. C., Taylor, R. L., and Pister, K. S. "Variational and Projection Methods for the Volume Constraint in Finite Deformation Elasto-Plasticity," *Comp. Meth. in App. Mech. Engg.*, 51, 1985.
17. Simo, J. C. and Taylor, R. L. "A Return Mapping Algorithm for Plane Stress Elasto-Plasticity," *Int. J. of Num. Meth. Engg.*, V. 22, 1986.
18. Wriggers, P., Eberlein, R., and Reese, S. "A Comparison of Three-Dimensional Continuum and Shell Elements for Finite Plasticity," *Int. J. Solids & Structures*, V. 33, N. 20-22, 1996.
19. Simo, J. C. and Taylor, R. L. "Quasi-Incompressible Finite Elasticity in Principal Stretches, Continuum Basis and Numerical Algorithms," *Comp. Meth. App. Mech. Engg.*, 85, 1991.
20. Marcal, P. V. "Finite Element Analysis of Combined Problems of Nonlinear Material and Geometric Behavior." in *Proceedings of the ASME Computer Conference, Computational Approaches in Applied Mechanics*, Chicago, 1969.
21. Melosh, R. J., and Marcal, P. V. "An Energy Basis for Mesh Refinement of Structural Continua." *Int. J. Num. Meth. Eng.* 11, 1083-1091, 1971.
22. Morman, K. N., Jr., Kao, B. G., and Nagtegaal, J. C. "Finite Element Analysis of Viscoelastic Elastomeric Structures Vibrating about Nonlinear Statically Stressed Configurations." SAE Technical Papers Series 811309, presented at 4th Int. Conference on Vehicle Structural Mechanics, Detroit, November 18-20, 1981.
23. Morman, K. N., Jr., and Nagtegaal, J. C. "Finite Element Analysis of Small Amplitude Vibrations in Pre-Stressed Nonlinear Viscoelastic Solids." *Int. J. Num. Meth. Engng*, 1983.
24. Nagtegaal, J. C. "Introduction in Geometrically Nonlinear Analysis." Int. Seminar on New Developments in the Finite Element Method, Santa Marherita Ligure, Italy, 1980.

25. Nagtegaal, J. C., and de Jong, J. E. "Some Computational Aspects of Elastic-Plastic Large Strain Analysis." in *Computational Methods in Nonlinear Mechanics*, edited by J. T. Oden. North-Holland Publishing Company, 1980.
26. Newmark, N. M. "A Method of Computation for Structural Dynamics." *ASCE of Eng. Mech.* 85, 67-94, 1959.
27. Parks, D. M. "A Stiffness Derivative Finite Element Technique for Determination of Elastic Crack Tip Stress Intensity Factors." *International Journal of Fractures* 10, (4), 487-502, December 1974.
28. Timoshenko, S., Young, D. H., and Weaver, Jr., W. *Vibration Problems in Engineering*. John Wiley, New York: 1979.
29. Wilkinson, J. H. *The Algebraic Eigenvalue Problem*. Oxford: Clarendon Press, 1965.
30. Chung, J. and Hulbert, G.M., "A family of single-step Houbolt time integration algorithms for structural dynamics", *Comp. Meth. in App. Mech. Engg.*, 118, 1994.
31. Shih, C.F., Moran B. and Nakamura K., "Energy release rate along a three-dimensional crack front in a thermally stressed body", *International Journal of Fracture*, vol. 30, pp. 79–102, 1986.
32. Anderson, T.L., *Fracture Mechanics: Fundamentals and Applications* (2nd ed.) CRC Press, 1995.
33. Yoon, J.W., Yang, D.Y. and Chung, K., "Elasto-plastic finite element method based on incremental deformation theory and continuum based shell elements for planar anisotropic sheet materials", *Comp. Methods Appl. Mech. Eng.*, 174, 23 (1999).

CHAPTER

6

Nonstructural and Coupled Procedure Library

- Heat Transfer
- Hydrodynamic Bearing
- Electrostatic Analysis
- Magnetostatic Analysis
- Electromagnetic Analysis
- Piezoelectric Analysis
- Acoustic Analysis
- Fluid Mechanics
- Coupled Analyses
- References

This chapter describes the nonstructural analysis procedures available in MSC.Marc. These are comprised of several areas including heat transfer, hydrodynamic bearing, electrostatic, magnetostatic, electromagnetic, piezoelectric, fluid mechanics, and coupled analysis. This chapter provides the technical background information as well as usage information about these capabilities.

Heat Transfer

MSC.Marc contains a solid body heat transfer capability for one-, two-, and three-dimensional, steady-state and transient analyses. This capability allows you to obtain temperature distributions in a structure for linear and nonlinear heat transfer problems. The nonlinearities in the problem may include temperature-dependent properties, latent heat (phase change) effect, heat convection in the flow direction, and nonlinear boundary conditions (convection and radiation). The temperature distributions can, in turn, be used to generate thermal loads in a stress analysis.

MSC.Marc can be applied to solve the full range of two- and three-dimensional transient and steady-state heat conduction and heat convection problems. MSC.Marc provides heat transfer elements that are compatible with stress elements. Consequently, the same mesh can be used for both the heat transfer and stress analyses. Transient heat transfer is an initial- boundary value problem, so proper initial and boundary conditions must be prescribed to the problem in order to obtain a realistic solution. MSC.Marc accepts nonuniform nodal temperature distribution as the initial condition, and can handle temperature/time-dependent boundary conditions. Both the thermal conductivity and the specific heat in the problem can be dependent on temperature; however, the mass density remains constant at all times. The thermal conductivity can also be anisotropic. Latent heat effects (solid-to-solid, solid-to-liquid phase changes) can be included in the analysis. A time-stepping procedure is available for transient heat transfer analysis. Temperature histories can be stored on a post file and used directly as thermal loads in subsequent stress analysis. User subroutines are available for complex boundary conditions such as nonlinear heat flux, convection, and radiation.

A summary of MSC.Marc capabilities for transient and steady-state analysis is given below.

- Selection of the following elements that are compatible with stress analysis:
 - 1-D: three-dimensional link (2-node, 3-node)
 - 2-D: planar and axisymmetric element (3-, 4-, and 8-node)
 - 2-D: axisymmetric shells
 - 3-D: solid elements (4-, 8-, 10-, and 20-node)
 - 3-D: shell elements (4- and 8-node)
- Specification of temperature-dependent materials (including latent heat effects) is performed with the model definition options **ISOTROPIC**, **ORTHOTROPIC**, **TEMPERATURE EFFECTS**, and **ORTHO TEMP**.
- Selection of initial conditions is done using the **INITIAL TEMP** option.

- Selection of the temperature and time-dependent boundary conditions (prescribed temperature history, volumetric flux, surface flux, film coefficients, radiation, change of prescribed temperature boundary conditions during analyses) is done using the **FIXED TEMPERATURE**, **TEMP CHANGE**, **DIST FLUXES**, **POINT FLUX**, and **FILMS** options.
- Import of viewfactors calculated by MSC.Marc Mentat for radiation analyses.
- Selection of time steps using history definition option **TRANSIENT** or **AUTO STEP**.
- Application of a tying constraints on nodal temperatures using model definition option **TYING**.
- Generation of a thermal load (temperature) file using the **POST** option, which can be directly interfaced with the stress analysis using model definition option **CHANGE STATE**.
- Use of user subroutines **ANKOND** for anisotropic thermal conductivity, **FILM** for convective and radiative boundary conditions or **FLUX** for heat flux boundary conditions.
- Selection of nodal velocity vectors for heat convection is done using **VELOCITY** and **VELOCITY CHANGE** options.
- Use of user subroutines **UVELOC** for heat convection.

In addition, a number of thermal contact gap and fluid channel elements are available in MSC.Marc. These elements can be used for heat transfer problems involving thermal contact gap and fluid channel conditions.

The **CONRAD GAP** option, used in conjunction with 4- and 8-node 2-D continuum elements or 8- and 20-node 3-D continuum elements, provides a mechanism for perfect conduction or radiation/convection between surfaces, depending on the surface temperatures. The perfect conduction capability in the elements allows for the enforcement of equal temperatures at nodal pairs and the radiation/convection capability allows for nonlinear heat conduction between surfaces, depending on film coefficient and emissivity. The perfect conduction is simulated by applying a tying constraint on temperatures of the corresponding nodal points. An automatic tying procedure has been developed for such elements. The radiation/convection capabilities in the elements are modeled by one-dimensional heat transfer in the thickness direction of the elements with variable thermal conductivity.

For the purpose of cooling, the **CHANNEL** option allows coolant to flow through passage ways that often appear in the solid. The fluid channel elements are designed for the simulation of one-dimensional fluid/solid convection conditions based on the following assumptions:

1. Heat conduction in the flow direction can be neglected compared to heat convection;
2. The heat flux associated with transient effects in the fluid (changes in fluid temperature at a fixed point in space) can be neglected.

For high velocity air flow, these assumptions are reasonable and the following approach is used:

Each cooling channel is modeled using channel elements. On the sides of the channel elements, convection is applied automatically. The film coefficient is equal to the film coefficient between fluid and solid, whereas the sink temperature represents the temperature of the fluid. A two-step staggered solution procedure is used to solve for the weakly coupled fluid and solid temperatures.

Thermal Contact

The [CONTACT](#) and [THERMAL CONTACT](#) options may also be used to define the fluxes entering the surfaces. For more information, see [Chapter 8 Contact](#) in this volume.

Convergence Controls

Use model definition option [CONTROL](#) to input convergence controls for heat transfer analysis. These options are:

1. Maximum allowable nodal temperature change. This is used only for transient heat transfer analysis in conjunction with an automatic time stepping scheme.
2. Maximum allowable nodal temperature change before properties are re-evaluated and matrices reassembled. This is used only for transient heat transfer analysis in conjunction with an automatic time stepping scheme. For mildly nonlinear problems, if the time step remains constant, the operator matrix is not reassembled until this value is reached.
3. Maximum error in temperature estimate used for property evaluation. This control provides a recycling capability to improve accuracy in highly nonlinear heat transfer problems (for instance, for latent heat or radiation boundary conditions). When non-zero, this control is used for both steady state and transient heat transfer analysis. Also, for transient analysis, this control is used for both fixed as well as adaptive stepping procedures.

Steady State Analysis

For steady state problems, use the **STEADY STATE** history definition option. If the problem is nonlinear, use the tolerance for temperature estimate error on the **CONTROL** model definition set to obtain an accurate solution.

You must distinguish between two cases of nonlinearity in steady-state solutions: mild nonlinearities and severe nonlinearities. In the case of mild nonlinearities, variations are small in properties, film coefficients, etc., with respect to temperatures. The steady-state solution can be obtained by iteration. After a small number of iterations, the solution should converge.

The technique described above is not suitable for severe nonlinearities. Examples of severe nonlinearities are radiation boundary conditions and internal phase-change boundaries. In these cases, you must track the transient with a sufficiently small time step (Δt) to retain stability until the steady-state solution is reached. Clearly, the choice of Δt is dependent on the severity of the nonlinearity. The number of steps necessary to obtain the steady-state solution can often be reduced by judicious choice of initial conditions. The closer the initial temperatures are to the steady-state, the fewer the number of increments necessary to reach steady-state.

Transient Analysis

Both fixed stepping and adaptive stepping schemes are available for transient heat transfer analysis. The fixed stepping scheme is **TRANSIENT NON AUTO**. The adaptive stepping schemes are **TRANSIENT** and **AUTO STEP**.

In the fixed stepping scheme, the program is forced to step through the transient with a fixed time step that is user specified. Only the error in temperature estimate (convergence control 3) is used with the fixed stepping scheme and the first two convergence controls are not checked.

The scheme for the **TRANSIENT** option is as follows:

1. After the program obtains a solution for a step, it calculates the maximum temperature change in the step and checks this value against the allowable temperature change per step given on the **CONTROL** option (convergence control 1).
2. If the actual maximum change exceeds the specified value, the program repeats the step with a smaller time step and continues repeating this step until the maximum temperature change is smaller than the specified value or until the maximum number of recycles given on the control option is reached (in which case the program stops).

3. If the actual maximum temperature change is:
 - a. between 80 percent and 100 percent of the specified value, the program goes on to the next step, using the same time step.
 - b. between 65 percent and 80 percent of the specified value, the program tries the next step with a time step of 1.25 times the current step. If the actual maximum is below 65 percent of the specified value, the program tries the next step with a time step that is 1.5 times the current step.

The objective of the scheme is to increase the time step as the analysis proceeds.

The **TRANSIENT** option can be used for both heat transfer as well as thermo-mechanical coupled analysis. The time step for coupled analyses is only based on the convergence characteristics of the heat transfer pass.

The **AUTO STEP** option is a unified time-stepping scheme that is available for thermal, mechanical and thermo-mechanically coupled analysis. The time step for coupled analyses is based on the convergence characteristics of each of the heat transfer and structural passes. Details of the **AUTO STEP** scheme are provided in Chapter 11.

Technical Background

Let the temperature $T(x)$ within an element be interpolated from the nodal values T of the element through the interpolation functions $N(x)$,

$$T(x) = N(x)T \quad (6-1)$$

The governing equation of the heat transfer problem is

$$C(T)\dot{T} + K(T)T = Q \quad (6-2)$$

In **Equation 6-2**, $C(T)$ and $K(T)$ are the temperature-dependent heat capacity and thermal conductivity matrices, respectively, T is the nodal temperature vector, $\dot{T}$ is the time derivative of the temperature vector, and Q is the heat flux vector.

The selection of the backward difference scheme for the discretization of the time variable in **Equation 6-2** yields the following expression:

$$\left[\frac{1}{\Delta t} C(T) + K(T) \right] T_n = Q_n + \frac{1}{\Delta t} C(T) T_{n-1} \quad (6-3)$$

Equation 6-3 computes nodal temperatures for each time increment $\Delta(t)$.

For the evaluation of temperature-dependent matrices, the temperatures at two previous steps provide a linear (extrapolated) temperature description over the desired interval

$$T(\tau) = T(t - \Delta t) + \frac{\tau}{\Delta t} (T(t - \Delta t) - T(t - 2\Delta t)) \quad (6-4)$$

This temperature is then used to obtain an average property of the material f over the interval to be used in [Equation 6-3](#), such that

$$f = \frac{1}{\Delta t} \int_{t - \Delta t}^t f[T(\tau)] d\tau \quad (6-5)$$

During iteration, the average property is obtained based on the results of the previous iteration:

$$T(\tau) = T(t - \Delta t) + \frac{\tau}{\Delta t} (T^*(t) - T(t - \Delta t)) \quad (6-6)$$

$T^*(t)$ are the results of the previous iteration.

Temperature Effects

The thermal conductivity, specific heat, and emissivity in a heat transfer analysis can depend on temperatures; however, the mass density remains constant. Specify reference temperature values of thermal conductivity, specific heat, mass density, and emissivity with the **ISOTROPIC** option. Enter temperature variations of both the thermal conductivity and specific heat using the **TEMPERATURE EFFECTS** option. This option also allows input of latent heat information. The temperature-dependent data can be entered using either the slope-break point representation or the property versus temperature representation. During analysis, an extrapolated/interpolated averaging procedure is used for the evaluation of temperature-dependent properties.

Latent heat can be induced because of a phase change that can be characterized as solid-to-solid, solid-to-fluid, fluid-to-solid, or a combination of the above, depending on the nature of the process.

Phase change is a complex material behavior. Thus, a detailed modeling of this change of material characteristic is generally very difficult. The use of numerical models to simulate these important phenomena is possible; several major factors associated with phase change of certain materials have been studied numerically.

The basic assumption of the latent heat option in MSC.Marc is that the latent heat is uniformly released in a temperature range between solidus and liquidus temperatures of the materials (see [Figure 6-1](#)). MSC.Marc uses a modified specific heat to model the latent heat effect. If the experimental data is sufficient and available, a direct input of the temperature-dependent specific heat data (see [Figure 6-2](#)) can be used. Results of both approaches are comparable if the temperature increments are relatively small.

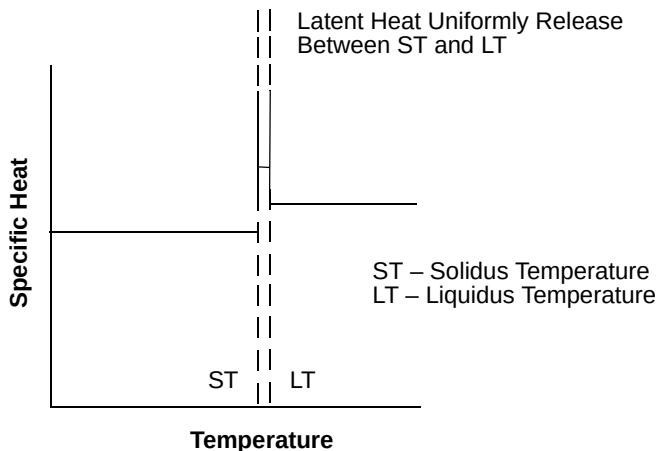


Figure 6-1 Modeling Phase Changes with the Latent Heat Option

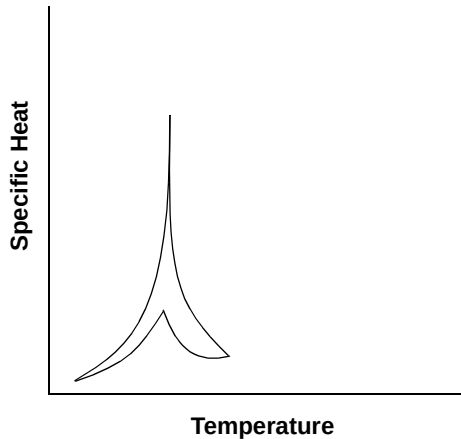


Figure 6-2 Modeling Phase Changes with the Specific Heat Option

Initial Conditions

In a transient heat transfer analysis, MSC.Marc accepts nonuniform nodal temperature distribution as the initial condition. Enter the initial condition through the model definition option **INITIAL TEMP** or through user subroutine **USINC**. Initial conditions are not required in steady-state heat transfer analysis, even though they can improve convergence when temperature-dependent properties are included.

Boundary Conditions

There are two types of boundary conditions in transient/steady-state heat transfer analysis: prescribed nodal temperatures and nodal/element heat fluxes. These boundary conditions are entered directly through input or through user subroutines.

Prescribe nodal temperatures at boundary nodes using the model definition option **FIXED TEMPERATURE**. Prescribe time-dependent nodal temperatures through user subroutine **FORCDT**. You can change prescribed nodal temperatures with the **TEMP CHANGE** option.

The model definition option **POINT FLUX** allows you to enter constant concentrated (nodal) heat fluxes such as heat source and heat sink. Use the user subroutine **FORCDT** for time and/or temperature-dependent nodal fluxes.

Use the model definition option **DIST FLUXES** and the user subroutine **FLUX** for constant and time/temperature-dependent distributed (surface or volumetric) heat fluxes.

Notes: In heat transfer analysis, you must always specify total values; for example, total temperature boundary conditions or total fluxes. This specification is to be used consistently for the heat transfer portion of analysis on coupled thermal-mechanical (thermal-solid), fluid-thermal, and fluid-thermal-solid.

The time variation used for thermal boundary conditions is a function of the stepping procedure that is used. If adaptive stepping (either **TRANSIENT** or **AUTO STEP**) is used, thermal boundary conditions are applied instantaneously. If fixed stepping (**TRANSIENT NON AUTO**) is used, the magnitudes of the thermal boundary conditions are applied in accordance with the tables used to define them.

Use the model definition option **FILMS** to input the constant film coefficient and the ambient temperature associated with the convective boundary conditions. Use the user subroutine **FILM** for time/temperature-dependent convective boundary conditions. The expression of the convective boundary condition is

$$q = H(T_s - T_\infty) \quad (6-7)$$

where q , H , T_s , and T_∞ are heat flux, film coefficient, unknown surface temperature, and ambient temperature, respectively.

The radiative boundary condition can be expressed as

$$q = \sigma \epsilon (T_s^4 - T_\infty^4) \quad (6-8)$$

where q is the heat flux, σ is the Stefan-Boltzmann coefficient, ϵ is emissivity, and T_s and T_∞ are unknown surface and ambient temperatures, respectively. The radiative boundary condition can be rewritten as

$$\begin{aligned} q &= \sigma \epsilon (T_s^3 + T_s^2 T_\infty + T_s T_\infty^2 + T_\infty^3) (T_s - T_\infty) \\ &= H(\sigma, \epsilon, T_s, T_\infty) (T_s - T_\infty) \end{aligned} \quad (6-9)$$

This shows that the radiative boundary condition is equivalent to a nonlinear convective boundary condition, in which the equivalent film coefficient $H(\sigma, \epsilon, T_s, T_\infty)$ depends on the unknown surface temperature T_s . This case requires the user subroutine **FILM**.

Use the model definition option **DIST FLUXES** and the user subroutine **FLUX** for constant and time/temperature-dependent distributed (surface or volumetric) heat fluxes.

If the heat flux defined in user subroutine **FLUX** or **FORCDT** is temperature dependent, then the convergence behavior of the solution process of the set of nonlinear equations can be improved by defining not only the current flux value, but also the derivative of the flux with respect to temperature. Based on this derivative, MSC.Marc adapts the conductivity matrix $K(T)$ as well as the heat flux vector Q . This is done by rewriting the heat flux using a Taylor series expansion. For a temperature dependent point flux on node i , the flux during iteration n is thus given by:

$$q_n^i = q_{n-1}^i + \left(\frac{dq^i}{dT} \right)_{n-1} (T_n^i - T_{n-1}^i) \quad (6-10)$$

Since all the terms in equation **Equation 6-10** except for T_n^i are known from iteration $n-1$, the corresponding entries in the conductivity matrix and heat vector are updated as:

$$K_n^{ii} \rightarrow K_n^{ii} - \left(\frac{dq^i}{dT} \right)_{n-1} \quad (6-11)$$

$$Q_n^i \rightarrow Q_n^i - \left(\frac{dq^i}{dT} \right)_{n-1} T_{n-1}^i \quad (6-12)$$

The temperature dependent distributed fluxes are treated in a similar way and require the user subroutine [FILM](#).

If a film coefficient is temperature dependent, then, similar to temperature dependent fluxes, defining the derivative of the film coefficient with respect to temperature may speed up the convergence of the iterative process. This derivative should be defined in user subroutine [FILM](#).

Radiation Viewfactors

One important element in radiation is the viewfactor calculations which are generally tedious and nontrivial. In many analyses, the radiative transfer of heat between surfaces plays a significant role. To properly model this effect, it is necessary to compute the proportion of one surface which is visible from a second surface known as the formfactor or viewfactor. The viewfactor, defined by a fourth order integral, presents many difficulties in its computation. Primary among these is the large amount of computing power needed, especially when shadowing effects are included.

The radiative flow of heat from surface 1 to surface 2 is given by:

$$q_{12} = \sigma F_{12}(T_1^4 - T_2^4) \quad (6-13)$$

in which, F_{12} is the viewfactor and is calculated as:

$$F_{12} = \frac{1}{A_1} \int_{A_1} \int_{A_2} \frac{\cos \phi_1 \cos \phi_2}{\pi r^2} dA_2 dA_1 \quad (6-14)$$

Computational Approach

There are several methods available for the viewfactor calculation, namely, the [Direct Adaptive Integration](#), [Adaptive Contour Integration](#), and the [Monte Carlo Method](#) method. A brief description of these methods is given below.

Direct Adaptive Integration

This approach computes viewfactors by directly computing a fourth order integral between every pair of surfaces. This means that there are N^2 integrations to be performed, a quadratic scaling.

Besides being quadratic, to achieve reasonable accuracy, this approach requires a huge number computations for surfaces which are close together; a situation which frequently occurs.

Using the direct integration approach, MSC.Marc calculates the viewfactors automatically for you in each cavity of an axisymmetric body involving radiative heat transfer. This capability is only available for axisymmetric bodies. The parameter **RADIATION** is used to activate this viewfactor calculation capability; the model definition option **RADIATING CAVITY** allows you to input outlines of each cavity in terms of nodal numbers.

You must subdivide the radiative boundary in this heat transfer problem into one or more unconnected cavities. For each cavity, you define the outline of the cavity in terms of an ordered sequence of nodes. Usually, the nodes coincide with the nodes of the finite element mesh. You can add extra nodes, provided you also give the appropriate boundary conditions.

The nodes must be given in counterclockwise order with respect to an axis orthogonal to the plane of the figure and pointing to the viewer (see [Figure 6-3](#)). If the cavity is not closed, the program adds the last side by connecting the last node with the first one. This side is treated as a black body as far as radiation is concerned; its temperature is taken as the average between the temperatures of the adjacent nodes.

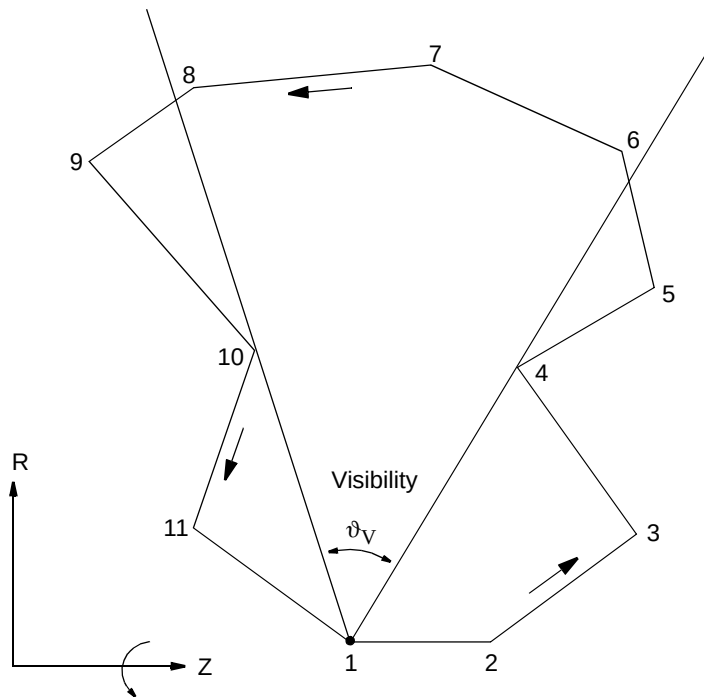


Figure 6-3 Radiating Cavity (Approach II: Valid for Axisymmetric Capability Only)

MSC.Marc internally computes the viewfactor between every side of the cavity and all other sides. The matrix with the viewfactors can be stored into a file, and read in again during a subsequent analysis, thus avoiding a new computation. During a transient heat transfer analysis, for every time step, the program estimates the temperature reached at the end of the step. From the estimated temperature, the emissivity (temperature dependent) is computed. In addition, the radiating heat fluxes are computed. The temperatures at the end of the step are computed by solving the finite element equations.

At every node, the difference between estimated and computed temperature is obtained. If the tolerance allowed by the model definition option **CONTROL** is exceeded, iterations within the time step take place. Otherwise, the computation of the step is concluded.

The cavity is defined by its boundary defined by a list of nodes ordered counterclockwise.

Insulated boundary condition (for example, symmetry boundaries) requires that the sum of the heat fluxes at a node be zero. This requirement is satisfied automatically. Therefore, no input is required for this type of boundary condition.

As previously mentioned, in a heat transfer analysis of axisymmetric body involving radiative boundary conditions, MSC.Marc automatically calculates viewfactors for radiation. A description of the viewfactor calculation follows.

The amount of radiation exchanged between two surfaces will depend upon what fraction of the radiation from each surface impinges the other surfaces. Referring to **Figure 6-3**, the radiation propagating from surface i to surface j will be:

$$q_{ij} = J_i A_i \left(\frac{\cos \phi_i \cos \phi_j A_j}{r^2} \right) = J_i A_i F_{ij} \quad (6-15)$$

It is noted that the viewfactor, F_{ij} , is solely geometrical in nature. From the definition of F_{ij} , we see that

$$A_j F_{ji} = A_i F_{ij} \quad (6-16)$$

We are now ready to derive the heat transfer radiation equation. Steady-state (equilibrium) energy conservation requires:

$$q_i = A_i (J_i - G_i) \quad (6-17)$$

Two independent expressions for G_i can be formed.

1. The incoming radiation on a surface must equal the radiation emitted by all other surfaces which strike this surface,

$$A_i G_i = \sum_{j=1}^N J_j A_j F_{ji} = \sum_{j=1}^N J_j A_i F_{ij}$$

(6-18)

$$G_i = \sum_{j=1}^N J_j F_{ij}$$

where N is the number of surfaces involved in the computations.

2. The other expression for G_i is

$$G_i = \frac{J_i - \varepsilon_i E_{n_i}}{\rho_i} = \frac{J_i - \varepsilon_i E_{n_i}}{1 - \varepsilon_i}$$

(6-19)

where

E_n is the emissive power

ε is the emissivity

ρ is the reflectance.

Substituting [Equation 6-19](#) into [Equation 6-17](#) and rearranging it gives

$$J_i = E_{n_i} - \frac{1 - \varepsilon_i}{A_i \varepsilon_i} q_i$$

(6-20)

This expression for J_i can be inserted into [Equation 6-18](#). After regrouping terms, you get the governing equation for gray body diffuse radiation problems

$$\left[\frac{A_i \delta_{ij} - (1 - \varepsilon_j) A_i F_{ij}}{A_j \varepsilon_j} \right] \{q_j\} = [A_i \delta_{ij} - A_i F_{ij}] \{E_{n_i}\}$$

(6-21)

For the problem of black bodies, that is, $\varepsilon = 1$, we have,

$$\{q_j\} = [A_i \delta_{ij} - A_i F_{ij}] \{E_{n_i}\}$$

(6-22)

This equation states the obvious; net radiation heat flow from a black surface is the difference between radiation given off and received; that is, there is no reflection.

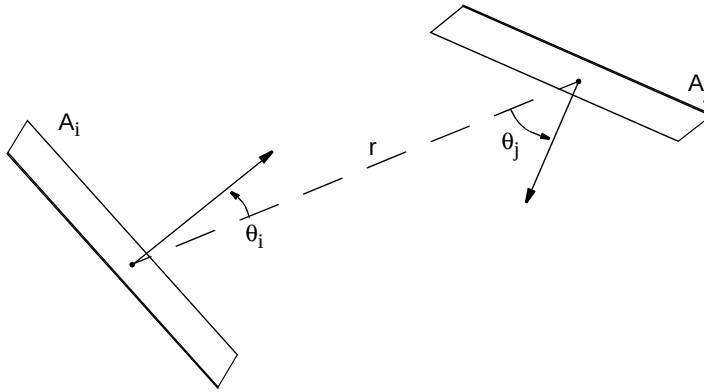


Figure 6-4 Viewfactor Definition

The heat flux radiating from A_i to A_j is computed as

$$q_{ij} = J_i(A_i \cos \theta_i) \left(\frac{A_j \cos \theta_j}{\pi r^2} \right) \quad (6-23)$$

where J_i is the power radiating from A_i , the first term within parentheses is the projection of A_j normal to the connecting line, and the second term is the solid angle under which A_j is seen from the center of A_i defining the viewfactor:

$$F_{ij} = \frac{\cos \theta_i \cos \theta_j}{\pi r^2} A_j \quad (6-24)$$

$$q_{ij} = J_i A_i F_{ij}$$

Adaptive Contour Integration

Although more efficient than direct integration, this method has very little chance to produce accurate shadowing effects. To do this would require the accurate construction of the contours caused by shadows; a step, which by itself, is extremely costly.

Monte Carlo Method

In this method, the idea is to randomly emit rays from the surface in question. The percentage of these rays which hit another surface is the formfactor between the surfaces. The Monte Carlo method computes N formfactors at one time, providing

linear scaling. In fact, the larger the number of surfaces, the faster the formfactors are computed compared to the Direct Adaptive Integration and Adaptive Contour Integration. Hence, this method is adopted for the viewfactor calculations.

Once the viewfactors are calculated by MSC.Marc Mentat; the resulting output file containing the viewfactors can be read in with the inclusion of the parameter **RADIATION** and the model definition option **VIEW FACTOR**.

Some of the features of the viewfactor calculation in MSC.Marc are:

1. You are not required to specify blocking elements. This is embedded into the algorithm completely and, hence, done automatically. This is specially useful in three-dimensional analysis since, for complex geometries, it is impractical to predict what surfaces are blocking other surfaces.
2. The cost of calculation is nearly linearly proportional to the number of elements which means that, for big problems, the cost does not increase significantly.
3. The methodology in MSC.Marc guarantees that the sum of viewfactors is always one. This is in contrast to the direct integration approach which has the requirement of normalization to avoid artificial heat gain and loss, because the sum of the viewfactors for any one surface is unlikely to add up to 1.0 due to errors in the computation of individual viewfactors.
4. Shadowing effects (due to two surfaces being hidden from one another by other surfaces) can be modeled.

For a surface to participate in the computations, it must participate in the following operations:

1. *Ray Emission:* The surface should be able to randomly emit a ray from its surface. The origin of the ray should be randomly distributed over the area of the surface (see [Figure 6-5](#)). The direction of the ray should be distributed according to the cosine of the angle between the ray and the normal to the surface at the origin of the ray, thereby emitting more rays normal to the surface than tangent to it.
2. *Ray Intersection:* Given an origin, direction, and length of a ray, the surface should be able to determine whether it is hit by that ray, and if hit, the length of the ray at the point of intersection.

Determination of viewfactors involves consideration of several desired properties like speed, accuracy, shadowing, translucence, absorption, nonuniform emission, reciprocity, and efficiency. In light of the properties listed above, the Monte Carlo approach is adapted in conjunction with the ray tracing and boxing algorithms.

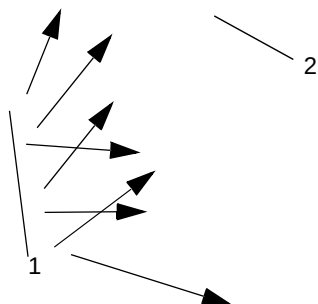


Figure 6-5 Random Ray Emission in Monte Carlo Method

[Figure 6-6](#) depicts the formation of shadows which essentially involves the computation of incident light by tracing rays from the light source to the point of incidence of the eye rays. These rays do not reflect or refract. Shadows are formed when light source rays are obstructed, either partially or fully, from reaching an object. Refraction and reflection of light from light sources is not computed.

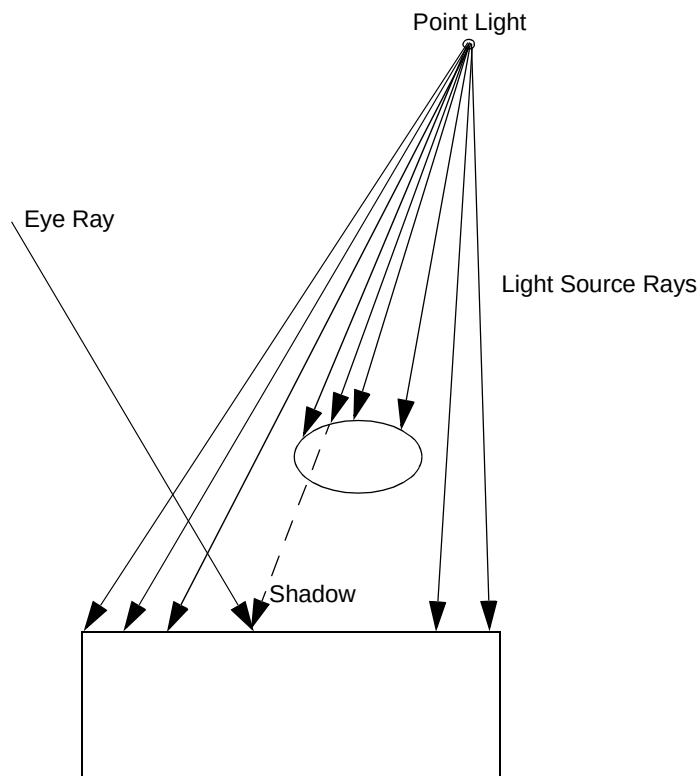


Figure 6-6 Shadowing Effects in Ray Tracing

Finally, to compute the viewfactors, it is not necessary to check the intersection of incident rays with all objects under consideration. Such a method would be prohibitively computationally expensive and preclude a large scale three dimensional analysis. An effective technique for fast calculation of intersections is employed.

The method relies on sorting the objects before any intersections are computed. The information computed is used to eliminate most of the intersection computations. The technique requires that each object to be sorted have a bounding box which entirely encloses it.

The sorting relies on creation of a binary tree of bounding boxes. Thus, a bounding box is computed for all objects. The objects are then sorted by the coordinate of the longest dimension of the bounding box. This list of objects is then divided into two sets, each having an equal number of objects. This process is repeated recursively until each set contains no more than a given maximum number of objects. This recursive sorting process is depicted in [Figure 6-7](#).

Ray intersections are performed by searching the binary bounding box tree which involves determining whether the ray hits the bounding box of all the objects. If the intersection with one bounding box is not found, then another bounding box is considered. In the event an intersection does exist, each object in the node is intersected with the ray if the intersection is a bottom node of the tree. Otherwise, the bounding boxes of both subtrees are intersected with the search in the closest subtree conducted first. The process continues until all rays are exhausted.

Conrad Gap

For the thermal contact gap element, in the *gap open* condition, two surface temperatures T_a and T_b at the centroid of the surfaces of a thermal contact element are obtained by interpolation from the nodal temperatures. These two surface temperatures are used for the computation of an equivalent conductivity for the radiation/convection link. The expression of the equivalent thermal conductivity k_1 is:

$$k_1 = \varepsilon \sigma L (T_{ka} + T_{kb}) (T_{ka}^2 + T_{kb}^2) + H L \quad (6-25)$$

where ε is the emissivity, σ is the Stefan-Boltzmann constant, L is the length of the element (distance between a and b), T_{ka} , T_{kb} are absolute temperatures at a and b converted from T_a and T_b ; and H is the constant film coefficient.

The equivalent thermal conductivity k_1 for the thermal contact element is assumed to be in the gap direction. The thermal conductivities in other two local directions are all set to zero. A coordinate transformation from the local to the global coordinate system allows the generation of the thermal conductivity matrix of the thermal contact element in the global system for assembly.

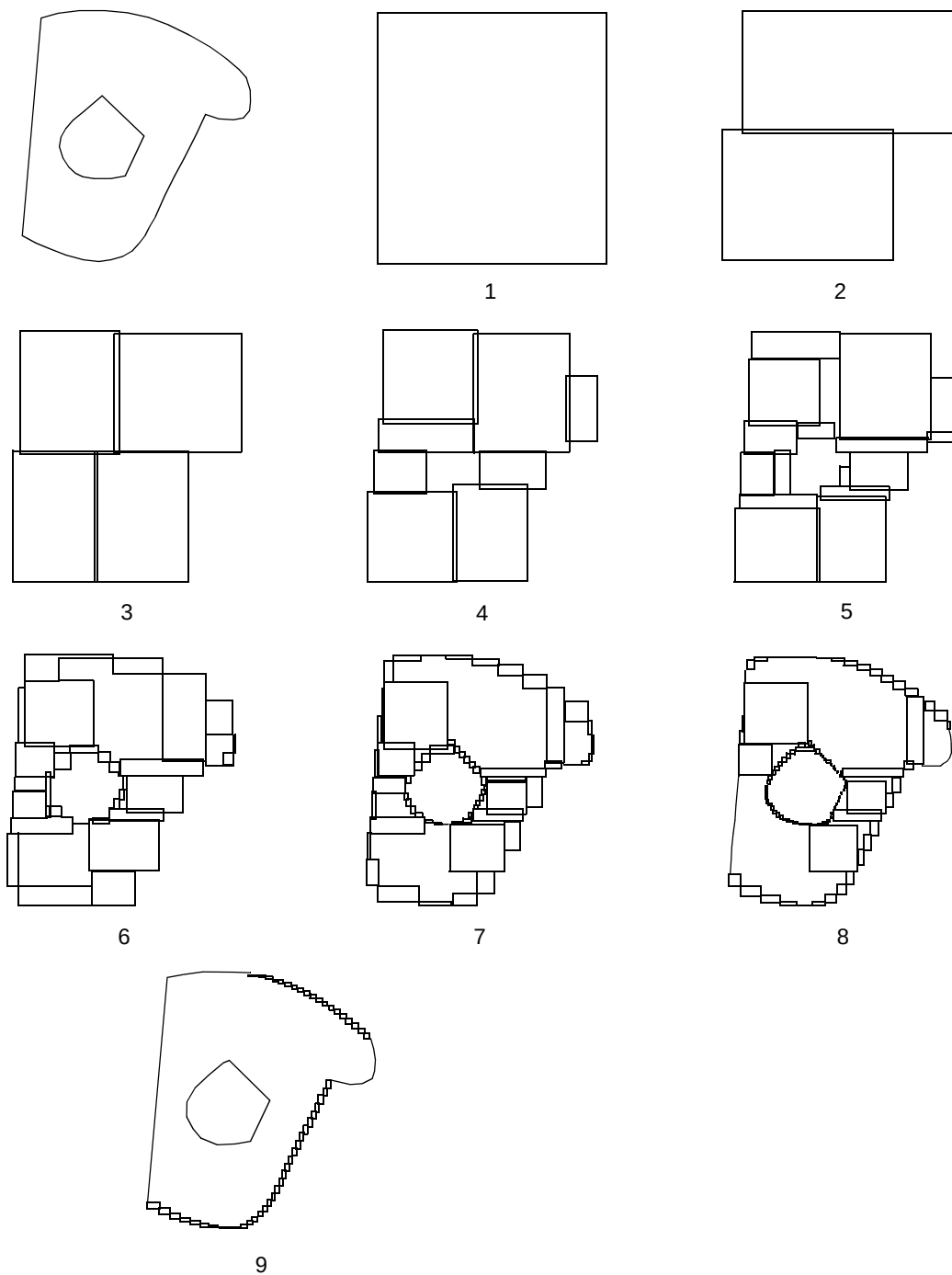


Figure 6-7 Fast Intersection Technique for Sorting Objects

Similarly, in the gap closed condition, tying constraints are automatically generated by the program for thermal contact elements. The constraint equation for each pair of nodes can be expressed as:

$$T_I + T_J \quad \text{if } (T_{gap} > T_{close}) \quad (6-26)$$

where

$$T_{gap} = \frac{1}{2}(T_I + T_J)$$

T_I, T_J = nodal temperatures at nodes I and J

T_{close} = gap closure temperature.

Channel

For the fluid channel element, the one-dimensional, steady-state, convective heat transfer in the fluid channel can be expressed as:

$$\begin{aligned} \dot{m}c \frac{\partial T_f}{\partial s} + \Gamma h(T_f - T_s) &= 0 \\ T_f(0) &= T_{inlet} \end{aligned} \quad (6-27)$$

where $\dot{m}$ is mass flow rate, c is specific heat, T_f is fluid temperature, T_s is solid temperature, s is streamline coordinate, Γ is circumference of channel, h is film coefficient, and T_{inlet} is inlet temperature.

Similarly, the conductive heat transfer in the solid region is governed by the following equation:

$$C\dot{T}_s + KT_s = Q \quad (6-28)$$

subjected to given initial condition and fixed temperature and/or flux boundary conditions. At the interface between the fluid and solid, the heat flux estimated from convective heat transfer is

$$q = h(T_s - T_f) \quad (6-29)$$

In [Equation 6-28](#), C is the heat capacity matrix, K is the conductivity matrix and Q is the heat flux vector. [Equation 6-28](#) and [Equation 6-29](#) are coupled equations. The coupling is due to the unknown solid temperature T_s appearing in [Equation 6-27](#) and unknown fluid temperature T_f in [Equation 6-29](#) for the solution of [Equation 6-28](#).

The solutions for Equation 6-27 and Equation 6-28 are obtained from the introduction of a backward difference for the discretization of time variable in Equation 6-28 and of streamline distance in Equation 6-27. Let

$$\dot{T}_s = [T_s^i - T_s^{i-1}]/(\Delta t) \quad (6-30)$$

we obtain

$$\left(\frac{1}{\Delta t}C + K\right)T_s^i = Q^i + \frac{1}{\Delta t}CT_s^{i-1} \quad (6-31)$$

where Δt = time-step in transient analysis. Similarly, let

$$\frac{dT_f}{ds} = [T_f^j - T_f^{j-1}]/(\Delta s) \quad (6-32)$$

we obtain

$$T_f^j = [\Delta s\beta + T_f^{j-1}]/(1 + \Delta s\alpha) \quad (6-33)$$

where Δs is the streamline increment,

$$\alpha = \Gamma h/(\dot{m}c); \quad \text{and} \quad \beta = \Gamma hT_s^{j-1}/(\dot{m}c) \quad (6-34)$$

Output

MSC.Marc prints out both the nodal temperatures and the temperatures at the element centroid when the **CENTROID** parameter is used, or at the integration points if the **ALL POINTS** parameter is invoked. You can also indicate on the **HEAT** parameter for the program to print out the temperature gradients and the resulting nodal fluxes.

To create a file of element and nodal point temperatures, use the **POST** model definition option. This file can be used as temperature input for performing a thermal stress analysis. This file is processed using the **CHANGE STATE** option in the subsequent thermal stress analysis. This post file can also interface with MSC.Marc Mentat or MSC.Patran plot temperature as a function of time.

Heat Transfer with Convection

MSC.Marc has the capability to perform heat transfer with convection if the velocity field is known. The numerical solutions of the convection-diffusion equation have been developed in recent years. The streamline-upwind Petro-Galerkin (SUPG) method has been implemented into the MSC.Marc heat transfer capability.

The elements which are available are described in [Table 6-1](#).

Table 6-1 Heat Transfer Convection Elements

Element Type	Description
36 , 65	2-, 3-node link
37 , 39 , 41 , 50 , 69 , 85 , 86	3-, 4-, 8-node planar
38 , 40 , 42 , 70 , 87 , 88 , 122	4-, 4-, 8-node axisymmetric
43 , 44 , 71 , 123 , 133	8-, 20-node hexahedron
135	4-node tetrahedral
133	10-node tetrahedron
131 , 132	6-node triangular

To activate the convection contribution, use the [HEAT](#) parameter and set the fifth field to 2. Due to the nonsymmetric nature of the convection term, the nonsymmetric solver is used automatically. Specify the nodal velocity vectors using the [VELOCITY](#) option. To change velocity, use [VELOCITY CHANGE](#). If nonuniform velocity vectors are required, user subroutine [UVELOC](#) is used. This capability can be used in conjunction with the [Rigid-Plastic Flow](#) on [page 90](#) to perform a coupled analysis, in which the velocity fields are obtained.

Technical Background

The general convection-diffusion equation is:

$$\rho * c \left(\frac{\partial T}{\partial t} + v \cdot \nabla T \right) = \nabla \cdot (\kappa \nabla T) + Q \tag{6-35}$$

The perturbation weighting functions are introduced as:

$$W = N + \alpha \left(\frac{h}{2|v|} (v \cdot \nabla N) \right) + \beta \left(\frac{h}{4|v|} \Delta t (v \cdot \nabla N) \right) \tag{6-36}$$

N is the standard interpolation function in [Equation 6-1](#). The upwinding parameter, α , is the weighting used to eliminate artificial diffusion of the solution; while the beta term, β , is to avoid numerical dispersion. $|v|$ is the magnitude of local velocity vectors. T is the temperature, κ is the diffusion tensor. Q is the source term and Δt is the time increment.

The optimal choice for α and β are:

$$\alpha = \coth(Peclet/2) - (2/Peclet)$$

$$\beta = C/3 - (2/Peclet) * (\alpha/C)$$

where *Peclet* is the local Peclet number in the local element and *C* is the local Courant number:

$$Peclet = \text{density} * \text{specific heat} * \text{characteristic length} * \text{magnitude of the fluid velocity} / \text{conductivity}$$

$$Peclet = \rho * c * h * |v| / k$$

and

$$C = |v| * (\Delta t) / h \text{ where } (\Delta t) \text{ is the time increment.}$$

The characteristic length *h* is defined in [Ref. 12] where $C \leq 1$ is required for numerical stability. When $C > 1$, the β is set to be zero and a large time step is recommended to avoid numerical dispersion.

Note: The interpolation function *N* is not the time-space functions defined in [Ref. 6], so that most MSC.Marc heat transfer elements can be used. The convection contribution of heat transfer shell elements is limited due to the definitions of the perturbation weighting function and the interpolation function.

Hydrodynamic Bearing

MSC.Marc has a hydrodynamic bearing analysis capability, which enables you to solve lubrication problems. This capability makes it possible to model a broad range of practical bearing geometries and to calculate various bearing characteristics such as load carrying capability, stiffness, and damping properties. It can also be used to analyze elasto-hydrodynamic problems.

The lubricant flow in hydrodynamic bearing is governed by the Reynolds equation. The bearing analysis capability has been implemented into MSC.Marc to determine the pressure distribution and mass flow in bearing systems.

MSC.Marc is capable of solving steady-state lubrication problems; the incremental procedure analyzes a sequence of different lubricant film profiles. MSC.Marc also can be used to solve coupled elasto-hydrodynamic problems. This analysis requires a step-by-step solution for both the lubrication and the stress problems using separate runs. Because the finite element meshes for each problem are different, the program does not contain an automated coupling feature.

Only one-dimensional or two-dimensional lubricant flow needs to be modeled, since no pressure gradient exists across the film height. This modeling is done with the available heat transfer elements. The library elements listed in [Table 6-2](#) can be used for this purpose.

Table 6-2 Hydrodynamic Bearing Elements

Element	Description
36	2-node, three dimensional link
37	3-node, planar triangle
39	4-node, bilinear quadrilateral
41	8-node, planar biquadratic quadrilateral
65	3-node, three-dimensional link
69	8-node, biquadratic quadrilateral with reduced integration
121	4-node bilinear quadrilateral with reduced integration
131	6-node triangle

MSC.Marc computes and prints the following elemental quantities: lubricant thickness, pressure, pressure gradient components, and mass flux components. Each of these is printed at the element integration point.

The nodal point data consists of pressures, equivalent nodal mass flux at fixed boundary points, or residuals at points where no boundary conditions are applied. In addition, the program automatically integrates the calculated pressure distribution over the entire region to obtain consistent equivalent nodal forces. This integration is only performed in regions where the pressure exceeds the cavitation pressure.

The output includes the load carrying capacity (the total force on the bearing). This capacity is calculated by a vectorial summation of the nodal reaction forces. In addition, the bearing moment components with respect to the origin of the finite element mesh can be calculated and printed.

To activate the bearing analysis option, use the **BEARING** parameter. If the analysis requires modeling of flow restrictors, also include the **RESTRICTOR** parameter.

The values of the viscosity, mass density, and cavitation pressure must be defined in the **ISOTROPIC** option. Specify temperature-dependent viscosity values via **TEMPERATURE EFFECTS**. If thermal effects are included, the **STATE VARS** parameter is also required. In hydrodynamic bearing analyses, temperature is the second state variable. Pressure is the first state variable.

The fluid thickness field can be strongly position-dependent. A flexible specification of the film profile is allowed by using either the nodal thickness or elemental thickness option. Define nodal thickness values in the **THICKNESS** option. You may also redefine the specified values via the user subroutine **UTHICK**. Elemental values of lubricant thicknesses can be defined in the **GEOMETRY** option.

MSC.Marc also enables the treatment of grooves. Constant groove depth magnitudes can be specified in the **GEOMETRY** option. If the groove depth is position-dependent, the contribution to the thickness field can be defined in user subroutine **UGROOV**.

The relative velocity of the moving surfaces is defined on a nodal basis in the **VELOCITY** option. In addition, you can redefine the specified nodal velocity components in user subroutine **UVELOC**.

Specify prescribed nodal pressure values in the **FIXED PRESSURE** option. Define restrictor type boundary conditions in the **RESTRICTOR** option. To specify nonuniform restrictor coefficients, use user subroutine **URESTR**.

Input nodal point mass fluxes using the **POINT FLUX** option. Specify distributed mass fluxes in the **DIST FLUXES** option. If nonuniform fluxes are necessary, apply this via user subroutine **FLUX**.

Define variations of the previously specified lubrication film thickness field through the **THICKNS CHANGE** option. The program adds this variation to the current thickness values and solves the lubrication problem.

Activate the calculation of bearing characteristics (that is, damping and stiffness properties) through the **DAMPING COMPONENTS** or **STIFFNS COMPONENTS** options. MSC.Marc evaluates these properties based on the specified change in film thickness. This evaluation requires the formation of a new right-hand-side, together with a matrix back substitution. This is performed within so-called subincrements. The bearing force components calculated within these subincrements represent the bearing characteristics (that is, the change in load carrying capacity for the specified thickness change or thickness rate). The previously specified total thickness is not updated within subincrements. The calculated bearing characteristics are passed through to user subroutine **UTHICK**. This allows you to define an incremental thickness change as a function of the previously calculated damping and/or stiffness properties. This procedure can be applied when analyzing the dynamic behavior of a bearing structure. Mechanical problems can often be represented by simple mass-damper-spring systems if the bearing structure is nondeformable. A thickness increment can be derived based on the current damping and stiffness properties by investigating the mechanical equilibrium at each point in time.

The bearing analysis capability deals with only steady-state solution and does not include the analysis of transient lubrication phenomena. Note that the incrementation procedure is only meant to analyze a sequence of film profiles. No nonlinearities are involved; each increment is solved in a single step without iteration.

To calculate the reaction forces that act on the bearing structure, MSC.Marc requires information about the spatial orientation of the lubricant. This information is not contained in the finite element model because of the planar representation of the lubricant. Therefore, it is necessary to define the direction cosines of the unit normal vector that is perpendicular to the lubricant on a nodal basis in user subroutine **UBEAR**. The resulting nodal reaction forces are printed.

MSC.Marc requires a step-by-step solution of both the lubrication problem and the stress problem in separate runs. The thickness changes need to be defined within the lubrication analysis based upon the displacements calculated in the stress analysis. The stress analysis post file and user subroutine **UTHICK** can be used for this purpose. The tractions to be applied in the stress analysis can be read from the bearing analysis post file in subroutine **FORCDT**.

Technical Background

The flow of a lubricant between two surfaces that move relative to each other is governed by the Reynolds equation

$$\nabla \cdot \left(\frac{\rho h^3}{12\eta} \nabla p \right) - \frac{\partial(\rho h)}{\partial t} - \frac{1}{2} \nabla(\rho h u) + M = 0 \quad (6-37)$$

where:

p is lubricant pressure

ρ is mass density

h is film thickness

η is viscosity

t is time

u is the relative velocity vector between moving surfaces

M is the mass flux per unit area added to the lubricant

The following assumptions are involved in the derivation of this equation:

- The lubricant is a Newtonian fluid; that is, the viscosity is constant.
- There is no pressure gradient across the film height.
- There is laminar flow.
- Inertial effects are negligible.
- The lubricant is incompressible; that is, mass density is constant.
- Thermal effects are absent.

By introducing the film constant

$$\lambda = \frac{\rho h^3}{12\eta} \quad (6-38)$$

Equation 6-38 can be written as

$$\nabla \cdot (\lambda \nabla p) + M^r = 0 \quad (6-39)$$

where M^r is the reduced mass flux given by

$$M^r = M - \frac{\partial(\rho h)}{\partial t} - \frac{1}{2} \nabla(\rho h u) \quad (6-40)$$

In case of a stationary bearing, the transient term in Equation 6-40 vanishes.

Three different kinds of boundary conditions can be distinguished for the lubrication problem: prescribed pressure on boundary, prescribed mass flux normal to the boundary, and mass flux proportional to pressure.

Prescribed pressure on boundary is specified as

$$p = \bar{p} \quad (6-41)$$

where $\bar{p}$ is the value of the prescribed pressure.

Prescribed mass flux normal to the boundary has the form

$$-\lambda \frac{\partial p}{\partial n} = \bar{m}_n - \frac{1}{2} \rho h \bar{u}_n = \bar{m}_n^r \quad (6-42)$$

where $\bar{m}_n^r$ is the reduced inward mass flux. Here, n refers to the inward normal on the boundary, and $\bar{m}_n$ and $\bar{u}_n$ are the inward components of total mass flux and relative velocity, respectively.

If a restrictor is used (as in hydrostatic bearings), the total mass flux is a linear function of the pressure on the boundary. This condition is specified as

$$m_n = -\lambda \frac{\partial p}{\partial n} + \frac{1}{2} \rho h u_n = c(\bar{p} - p) \quad (6-43)$$

or, written in a slightly different form

$$-\lambda \frac{\partial p}{\partial n} = c(p^r - p) \quad (6-44)$$

where c is the restriction coefficient and

$$\bar{p}^r = \bar{p} - \frac{\rho h \bar{u}_n}{2c} \quad (6-45)$$

is the reduced pressure.

The differential [Equation 6-37](#), together with the boundary conditions ([Equation 6-42](#), [Equation 6-43](#), and [Equation 6-45](#)) completely describe the lubrication problem. This is analogous to a heat conduction problem as shown in [Table 6-3](#).

Table 6-3 Comparison of Lubrication and Heat Conduction

Lubrication		Heat Conduction	
Pressure	p	Temperature	T
Film constant	λ	Conductivity	k
Reduced body mass flux	M^r	Body heat flux	Q
Reduced boundary mass flux	m_n	Boundary heat flux	q_n
Restriction coefficient	c	Film coefficient	h
Reduced reference pressure	p^r	Reference temperature	T_r

Electrostatic Analysis

MSC.Marc has the capability to perform electrostatic analysis. This allows the program to evaluate the electric fields in a body or media, where electrical charges are present. This can be solved for one-, two-, or three-dimensional fields. The semi-infinite elements can be used to represent an infinite domain. The electrostatic problem is governed by the Poisson equation for a scalar potential. This analysis is purely linear and has been implemented in MSC.Marc analogously to the steady state heat transfer problem. The available elements are described in [Table 6-4](#) below.

Table 6-4 Element Types for Electrostatic Analysis

Element Type	Description
37 , 39 , 131 , 41	3-, 4-, 6-, 8-node planar
69	8-node reduced integration planar
121	4-node reduced integration planar
101 , 103	6-, 9-node semi-infinite planar
38 , 40 , 132 , 42	3-, 4-, 6-, 8-node axisymmetric
70	8-node reduced integration axisymmetric
122	4-node reduced integration axisymmetric
43 , 44	8-, 20-node 3-dimensional brick
71	20-node reduced integration brick
105 , 106	12-, 27-node semi-infinite brick
123	8-node reduced integration brick
133	10-node tetrahedral
50 , 85 , 86	3-, 4-, 8-node shell
87 , 88	2-, 3-node axisymmetric shell
135	4-node tetrahedral

MSC.Marc computes and prints the following quantities: electric potential field vector (E) and electric displacement vector (D) at the element integration points. The nodal point data consists of the potential ϕ and the charge Q .

To activate the electrostatic option, use the **ELECTRO** parameter. The value of the isotropic permittivity constant is given in the **ISOTROPIC** option, orthotropic constants can be specified using the **ORTHOTROPIC** option. Optionally, user subroutine **UEPS** can be used.

Specify nodal constraints using the **FIXED POTENTIAL** option. Input nodal charges using the **POINT CHARGE** option. Specify distributed charges by using the **DIST CHARGES** option. If nonuniform charges are required, user subroutine **FLUX** can be used for distributed charges and user subroutine **FORCDT** for point charges.

The electrostatic capability deals with linear, steady-state problems only. The **STEADY STATE** option is used to begin the analysis. The resultant quantities can be stored on the post file for processing with MSC.Marc Mentat.

Technical Background

The Maxwell equations to govern electrostatics are written in terms of the electric displacement vector D and the electric field vector E such that

$$\nabla \cdot D = \rho \quad (6-46)$$

and

$$\nabla \times E = 0 \quad (6-47)$$

where ρ is a given volume charge density.

The constitutive law is typically given in a form as:

$$D = \epsilon E \quad (6-48)$$

where ϵ is the dielectric constant.

Introducing the scalar potential ϕ such that

$$E = -\nabla\phi \quad (6-49)$$

which satisfied the constraint **Equation 6-47** exactly.

Denoting the virtual scalar potential by ψ , the variational formulation is

$$\int_V \nabla\psi \cdot (\epsilon \nabla\phi) dV = \int_V \psi \rho dV + \int_A \psi \cdot (\epsilon \nabla\phi - n) da \quad (6-50)$$

The natural boundary condition is applied through the surface integral in terms of the normal electric displacement using

$$-\epsilon \nabla \phi \cdot \mathbf{n} = \mathbf{D} \cdot \mathbf{n} = D_n \quad (6-51)$$

Consider a material interface A_{12} , separating two materials 1 and 2, and as ϕ is continuous over the material interface, the tangential electric field constraint is automatically satisfied.

$$\mathbf{n} \times (\mathbf{E}_1 - \mathbf{E}_2) = 0 \quad \text{on } A_{12} \quad (6-52)$$

If charges are present on the interface, these are applied as distributed loads as follows:

$$\mathbf{n} \cdot (\mathbf{D}_1 - \mathbf{D}_2) = \rho_s \quad \text{on } A_{12} \quad (6-53)$$

Using the usual finite element interpolation functions N and their derivatives β , we obtain

$$\psi = N\Psi \quad \phi = N\Phi$$

$$\mathbf{K} = \int_V \beta^T \epsilon \beta dV \quad (6-54)$$

$$\mathbf{F} = \int_V N^T \rho N dV + \int_A N^T N D_n da + \int_{A_{12}} N^T N \rho_s da \quad (6-55)$$

and finally

$$\mathbf{K}\Phi = \mathbf{F} \quad (6-56)$$

Magnetostatic Analysis

MSC.Marc has the capability to perform magnetostatic analysis. This allows MSC.Marc to calculate the magnetic field in a media subjected to steady electrical currents. This can be solved for two- or three-dimensional fields. Semi-infinite elements can be used to represent an infinite domain.

The magnetostatic analysis for two-dimensional analysis is solved using a scalar potential, while for three-dimensional problems, a full vector potential is used. The magnetic permeability can be a function of the magnetic field, hence, creating a nonlinear problem. Only steady-state analyses are performed. The available elements which are described in [Table 6-5](#).

Table 6-5 Elements Types for Magnetostatic Analysis

Element Type	Description
37 , 39 , 131 , 41	3-, 4-, 6-, 8-node planar
69	8-node reduced integration planar
121	4-node reduced integration planar
101 , 103	6-, 9-node semi-infinite planar
102 , 104	6-, 9-node semi-infinite axisymmetric
38 , 40 , 132 , 42	3-, 4-, 6-, 8-node axisymmetric
70	8-node reduced integration axisymmetric
122	4-node axisymmetric reduced integration
109	8-node brick
110	12-node semi-infinite brick

MSC.Marc computes and prints magnetic induction (B), the magnetic field vector (H), and the vector potential at the element integration points. The nodal point data consist of the potential (A) and the current (J).

To activate the magnetostatic option, use the [MAGNETO](#) parameter. The value of the isotropic permeability (μ) is given on the [ISOTROPIC](#) option; orthotropic constants can be specified using the [ORTHOTROPIC](#) option. Optionally, user subroutine [UEPS](#) can be used. Often, it is easier to specify $(1/\mu)$, which is also available through these options. A nonlinear permeability can be defined using the B-H relation.

Specify nodal constraints using the **FIXED POTENTIAL** option. Input nodal currents using the **POINT CURRENT** option. Specify distributed currents by using the **DIST CURRENT** option. Permanent magnets can be introduced by using the **PERMANENT** option for isotropic materials, or by entering a remanence vector via the **B-H RELATION** option for orthotropic materials. In addition, user subroutine **FLUX** can be used for nonuniform distributed current and user subroutine **FORCDT** for fixed nodal potentials or point current.

The magnetostatic capability is linear unless a nonlinear B-H relation is defined. In such problems, convergence is reached when the residual current satisfies the tolerance defined in the **CONTROL** option. The **STEADY STATE** option is used to begin the analysis. The resultant quantities can be stored on the post file for processing with MSC.Marc Mentat.

Technical Background

The Maxwell equations for magnetostatics are written in terms of the magnetic flux density vector B such that

$$\nabla \times H = J \quad (6-57)$$

and

$$\nabla \cdot B = 0 \quad (6-58)$$

where J is the current density vectors.

For magnetic materials, the following relation between B , H , and M , the magnetization vector, holds:

$$B = \mu_0(H + M) \quad (6-59)$$

with μ_0 being the permeability of vacuum. Denoting the magnetic susceptibility by χ_m and the permanent magnet vector by M_0 , we have

$$M = \chi_m H + M_0 \quad (6-60)$$

which can be substituted into Equation 6-59 to yield

$$B = \mu H + B_r \quad (6-61)$$

in which μ is the permeability, given by

$$\mu = \mu_0(1 + \chi_m) \quad (6-62)$$

and B_r is the remanence, given by

$$B_r = \mu_0 M_0 \quad (6-63)$$

Notice that $(1 + \chi_m)$ is usually called the relative permeability μ_r .

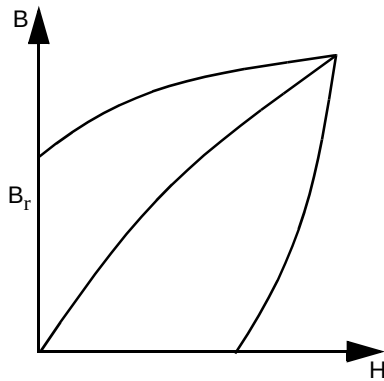


Figure 6-8 Nonlinear B-H Relationship

For isotropic linear material, χ_m and M_0 are scalar constants. If the material is orthotropic, χ_m and M change into tensors. For real ferromagnetic material, χ_m and μ are never constant. Instead, they depend on the strength of the magnetic field. Usually this type of material nonlinearity is characterized by a so-called magnetization curve or B-H relation specifying the magnitude of (a component of) B as a function of (a component of) H .

In MSC.Marc, the magnetization curve can be entered via the **B-H RELATION** option. For isotropic material, only one set of data points, representing the magnitude of the magnetic induction, $|B|$, as a function of $|H|$, the magnitude H , needs to be given. For orthotropic material, multiple curves are needed with each curve relating a component of B to the corresponding component of H .

From [Equation 6-61](#) and [Equation 6-63](#), it can be seen that for orthotropic materials, a permanent magnetization or remanence can be entered through the **B-H RELATION** option, by putting in a nonzero B value for $H = Q$. For isotropic material, this does not work since the direction of the remanence vector is still indeterminate. Therefore, in the isotropic case, the only possibility is to supply the magnetization vector through the **PERMANENT** option. Any offset of the $|B|(|H|)$ -curve, implying

$|B| \neq 0$ at $(|H| = 0)$ is disregarded in this case. For orthotropic material, it is not allowed to use the [PERMANENT](#) option. In this case, the magnetization can exclusively be specified through the [B-H RELATION](#) option.

It is emphasized that the magnetization curve specified under [B-H RELATION](#) must be monotonic and uniquely defined.

Introducing the vector potential A such that

$$\nabla \times A = B \quad (6-64)$$

which automatically satisfies the constraint, [Equation 6-58](#), and we then have the final form

$$\nabla \times \mu^{-1}(\nabla \times A) = J + \mu^{-1}B_r \quad (6-65)$$

The vector potential (A) is not uniquely defined by [Equation 6-64](#). In 2-D magnetostatic simulations, this indeterminacy is removed by the reduction of A to a scalar quantity. In 3-D situations, MSC.Marc uses the Coulomb gauge for this purpose:

$$\nabla \cdot A = 0 \quad (6-66)$$

Denote the virtual potential by W ; then, the variational formulation is

$$\int_V \mu^{-1}(\nabla \times W) \cdot (\nabla \times A) dV = \int_V W \cdot J dV + \int_V W \cdot (\nabla \times \mu^{-1}B_r) dV + \int_{\Gamma} W \cdot (H \times n) dA \quad (6-67)$$

where n is the outward normal to V at the boundary Γ .

In the three-dimensional case, the Coulomb gauge, [Equation 6-66](#), is enforced with a penalty formulation. The resulting term added to the variational formulation, [Equation 6-67](#) reads:

$$\begin{aligned} \int_V \mu^{-1}(\nabla \times W) \cdot (\nabla \times A) dV = & \int_V W \cdot J dV + \int_V W \cdot (\nabla \times \mu^{-1}B_r) dV + \int_{\Gamma} W \cdot (H \times n) dA \\ & + \int_V r(\nabla \cdot W)(\nabla \cdot A) dV \end{aligned} \quad (6-68)$$

The default value used for r is:

$$r = 10^{-4} |\mu^{-1}| \quad (6-69)$$

Using the usual finite element interpolation functions N , the discrete curl operator G , and the weighting function $W = N$.

$$G = \begin{bmatrix} \partial N / \partial y \\ \partial N / \partial x \end{bmatrix} \quad \text{for two-dimensional problems} \quad (6-70)$$

$$G = \begin{bmatrix} \partial N / \partial y - \partial N / \partial z \\ \partial N / \partial z - \partial N / \partial x \\ \partial N / \partial x - \partial N / \partial y \end{bmatrix} \quad \text{for three-dimensional problems}$$

Leads to the resulting system of algebraic equations

$$KA = F \quad (6-71)$$

where

$$K = \int_V G^T \mu^{-1} G dV \quad (6-72)$$

$$F = \int_V N^T N(J) dV + \int_V G^T M^{-1} B_r dV + \int_\Gamma N^T N(H \times n) da \quad (6-73)$$

The Coulomb gauge is based upon the principle of conservation of electric charge which in its steady state form reads:

$$\nabla \cdot J = 0 \quad (6-74)$$

In MSC.Marc, it is up to you to specify the current distribution. When doing so, it is recommended to ensure that this distribution satisfies [Equation 6-73](#). Otherwise, condition [Equation 6-66](#) could be violated. As a consequence, the results could become less reliable.

From the third term on the right-hand side of [Equation 6-73](#), it becomes clear that the natural boundary condition in this magnetostatic formulation is $H \times n$, the tangential component of the magnetic field intensity. Consequently, if no other condition is specified by you, by default a zero tangential magnetic field intensity at the boundaries is assumed.

Besides, when there are no currents present on a Γ_{12} material interface separating two materials 1 and 2, the tangential magnetic field intensity is assumed to be continuous:

$$n \times (H_1 - H_2) = 0 \text{ on } \Gamma_{12} \quad (6-75)$$

With H_1 and H_2 , the magnetic field intensities in material 1 and 2, respectively. A discontinuous tangential magnetic field intensity can be modeled by assigning an appropriate distributed “shear” current density to the interface. This (surface) current density J is related to H_1 and H_2 by:

$$n \times (H_1 - H_2) = J \text{ on } \Gamma_{12} \quad (6-76)$$

Electromagnetic Analysis

MSC.Marc has the capability to perform both transient (dynamic) and harmonic fully coupled electromagnetic analysis. This allows MSC.Marc to calculate the electrical and magnetic fields subjected to external excitation. This can be solved for both two- or three-dimensional fields. A vector potential for the magnetic field is augmented with a scalar potential for the electrical field. If a transient analysis is performed, the magnetic permeability can be a function of the magnetic field; hence, a nonlinear problem. The elements available for electromagnetic analysis are described in [Table 6-6](#).

Table 6-6 Element Types for Electromagnetic Analysis

Element Type	Description
111	4-node planar
112	4-node axisymmetric
113	8-node brick

MSC.Marc prints the magnetic flux density (B), the magnetic field vector (H), electric flux density (D), and the electrical field intensity at the integration points. In a harmonic analysis, these have real and imaginary components. The nodal point data consists of the vector potential A , the scalar potential V , the charge Q , and current I .

To activate the electromagnetic option, use the [EL-MA](#) parameter. The values of the isotropic permittivity (ϵ), permeability (μ), and conductivity (σ) are given in the [ISOTROPIC](#) option. Orthotropic constants can be specified using the [ORTHOTROPIC](#) option. User subroutines [UEPS](#), [UMU](#), and [USIGMA](#) can be optionally used. A nonlinear permeability can be defined using the B-H relation.

Specify nodal constraints using the [FIXED POTENTIAL](#) option. Input nodal currents and charge using the [POINT CURRENT](#) option. Specify distributed currents by using the [DIST CURRENT](#) option and distributed charges by using the [DIST CHARGE](#) option. Nonuniform distributed currents and charges can also be specified by user subroutine [FORCEM](#).

The electromagnetic capability is linear, unless a nonlinear B-H relation is defined. In such problems, convergence is reached when the residual satisfies the tolerance defined in the [CONTROL](#) option. The transient capability is only available with a fixed time step; use the [DYNAMIC CHANGE](#) option to activate this option. The resultant quantities can be stored on the post file for processing with MSC.Marc Mentat.

In electromagnetic analysis, you can enter the current and/or the charge. In a harmonic analysis, you can enter both the real and imaginary components.

Table 6-7 Input Options for Electromagnetic Analysis

Load Description	Input Options		
	Model Definition	History Definition	User Subroutine
Nodal Current Nodal Charge	POINT CURRENT	POINT CURRENT	FORCDT
Distributed Current	DIST CURRENT	DIST CURRENT	FORCEM
Distributed Charge	DIST CHARGE	DIST CHARGE	FORCEM

Technical Background

Technical Formulation

Electromagnetic analysis is based upon the well-known Maxwell's equations. This has been implemented in MSC.Marc for both transient and harmonic analyses.

Transient Formulation

The Maxwell's equations are:

$$\nabla \times E + \dot{B} = 0 \quad (6-77)$$

$$\nabla \times H - \epsilon \dot{E} - \sigma E = 0 \quad (6-78)$$

$$\nabla \cdot (\epsilon \dot{E} + \sigma E) = 0 \quad (6-79)$$

$$\nabla \cdot B = 0 \quad (6-80)$$

where the constitutive relations are

$$D = \epsilon E$$

$$B = \mu_0(H + M)$$

$$J = \sigma E$$

and

E is the electric field intensity
 D is the electric flux density
 H is the magnetic field intensity
 B is the magnetic flux density
 J is the current density
 M is the magnetization

and

ϵ is the permittivity
 μ is the permeability
 σ is the conductivity
 μ_0 is the permeability of free space.

Additionally, we have the conservation of charge: $\dot{\rho} + \nabla \cdot J = 0$

where ρ is the charge density. We assume that the magnetization vector is given by

$$M = \chi H + M_0 \quad (6-81)$$

where M_0 is the strength of the permanent magnet and χ is the susceptibility. The magnetic field can be defined as

$$B = \mu H + B_r \quad (6-82)$$

in which μ is the permeability, given by

$$\mu = \mu_0(1 + \chi_m) \quad (6-83)$$

and B_r is the remanence, given by

$$B_r = \mu_0 M_0 \quad (6-84)$$

Notice that $(1 + \chi_m)$ is usually called the relative permeability μ_r .

A vector magnetic potential A and a scalar potential V are introduced, such that

$$B = \nabla \times A \quad (6-85)$$

$$E = -(\nabla V + \dot{A}) \quad (6-86)$$

Note that since only the curl of A is required, an arbitrary specification of the divergence can be made. The Coulomb gauge is then introduced.

$$\nabla \cdot A = 0 \quad (6-87)$$

This is implemented using a penalty condition. It is important to note that E depends not only on the scalar potential, but also upon the vector potential. Hence, interpretation of V as the usual voltage can lead to erroneous results.

Substituting into Maxwell's equations results in:

$$\nabla \times [\mu^{-1}(\nabla \times A - \mu_0 M_0)] + \varepsilon(\nabla \dot{V} + \ddot{A}) + \sigma(\nabla V + \dot{A}) = 0 \quad (6-88)$$

$$-\nabla \cdot [\varepsilon(\nabla \dot{V} + \ddot{A})] + \sigma(\nabla V + \dot{A}) = 0 \quad (6-89)$$

It has been assumed that $\dot{\varepsilon} = 0$; in that, the permittivity has a zero time derivative.

Due to the hyperbolic nature of the above system (similar to dynamics), a Newmark-beta algorithm is employed in the discretization.

The general form is

$$A^{n+1} = A^n + \Delta t \dot{A}^n + \left(\frac{1}{2} - \beta\right) \Delta t^2 \ddot{A}^n + \beta \Delta t^2 A^{n+1} \quad (6-90)$$

$$\dot{A}^{n+1} = \dot{A}^n + (1 - \gamma) \Delta t \ddot{A}^n + \gamma \Delta t \ddot{A}^{n+1} \quad (6-91)$$

The particular form of the dynamic equations corresponding to the trapezoidal rule:

$$\gamma = \frac{1}{2} \quad \beta = \frac{1}{4} \quad (6-92)$$

results in a symmetric formulation, which is unconditionally stable for linear systems.

In the current formulation, a fixed time step procedure must be used. The time step is defined through the **DYNAMIC CHANGE** option.

Harmonic Formulation

In harmonic analysis, it is assumed that the excitation is a sinusoidal function, and the resultant also has a sinusoidal behavior. This results in the solution of a complex system of equations. In this case, Maxwell's equations become

$$\nabla \times E + i\omega B = 0 \quad (6-93)$$

$$\nabla \times H - i\varepsilon\omega E - \sigma E = 0 \quad (6-94)$$

$$\nabla \cdot D - \rho = 0 \quad (6-95)$$

$$\nabla \cdot B = 0 \quad (6-96)$$

where ω is the excitation angular frequency and $i = \sqrt{-1}$.

Additionally, we have the conservation of charge

$$i\omega\rho + \nabla \cdot J = 0 \quad (6-97)$$

where all symbols are the same as in the discussion above regarding transient behavior. Again, a vector potential A and a scalar potential V are introduced. In this case, these are complex potentials. Substituting into the Maxwell's equations results in

$$\nabla \times [\mu^{-1}(\nabla \times A - \mu_o M_o)] + \tilde{\sigma}(\nabla V + i\omega A) = 0 \quad (6-98)$$

$$-\nabla \cdot [\tilde{\sigma}(\nabla V + i\omega A)] = 0 \quad (6-99)$$

where $\tilde{\sigma} = \sigma + i\omega\epsilon$

With a little manipulation, a symmetric complex formulation may be obtained. The excitation frequency is prescribed using the [HARMONIC](#) option.

Note that the capability to extract the natural frequencies of a complex system by modal analysis does not exist in MSC.Marc. The harmonic formulation is assumed to be linear; therefore, you should not include the [B-H RELATION](#) option.

Piezoelectric Analysis

The piezoelectric effect is the coupling of stress and electric field in a material. An electric field in the material causes the material to strain and vice versa. MSC.Marc has a fully coupled implementation of piezoelectric analysis, thus simultaneously solving for the nodal displacements and electric potential. The elements available for piezoelectric analysis are described in [Table 6-8](#). They can be used in static, transient dynamic, harmonic, and eigenvalue analysis. The analysis can be geometrically nonlinear but is materially linear. The piezoelectric elements have their equivalent heat transfer elements, so that they can also be used in a coupled thermal-piezoelectric analysis. Such a coupled analysis is weakly coupled, and solved using a staggered approach. When piezoelectric elements are used in a contact analysis with a node of the piezoelectric element touching a segment of another piezoelectric element, then a multipoint constraint relation is set up for the nodal displacements as well as for the electric potential.

Table 6-8 Element Types for Piezoelectric Analysis

Element Type	Description
160	4-node plane stress
161	4-node plane strain
162	4-node axisymmetric
163	8-node brick
164	4-node tetrahedron

MSC.Marc prints the stresses (σ), the strains (ϵ), the electric displacement (D) and the electric field intensity (E) at the integration points. The nodal data points consists of the displacements u , forces f , the potential ϕ , and the charge Q .

To perform a piezoelectric analysis, use the [PIEZO](#) parameter. The values of the isotropic, orthotropic and anisotropic mechanical properties are given in the [ISOTROPIC](#), [ORTHOTROPIC](#), and [ANISOTROPIC](#) model definition option, respectively. The electric constants, and the constants defining the coupling between the mechanical and electric part can be specified with the [PIEZOELECTRIC](#) model definition option.

Specify nodal constraints using the [FIXED DISP](#) option or [FIXED POTENTIAL](#) option. Input nodal loads using the [POINT LOAD](#) option or nodal charges using the [POINT CHARGE](#) option. Specify distributed loads with the [DIST LOADS](#) option and distributed charges with the [DIST CHARGE](#) option. Fixed nodal displacements and potentials, or nodal forces and charges can also be specified by user subroutine [FORCDT](#), nonuniform distributed loads can be specified by user subroutine [FORCEM](#), and nonuniform distributed charges can be specified by user subroutine [FLUX](#).

Technical Background

The mechanical equilibrium equation for the piezoelectric effect is

$$\int_V \sigma : \delta \epsilon dV = \int_S t \cdot \delta u dS + \int_V f \cdot \delta u dV \quad (6-100)$$

and the electrostatic equilibrium equation is (see also [Equation 6-50](#))

$$\int_V D \cdot \delta E dV = \int_S \delta \phi D \cdot n dS + \int_V \rho_V \delta \phi dV \quad (6-101)$$

where

σ is the stress tensor

ϵ is the strain tensor

t is the traction at a point on the surface

u is the displacement

f is the body force per unit volume

D is the electric displacement vector

E is the electric field vector

ρ is the volume charge density

$\delta E = -\frac{\partial}{\partial x} \delta \phi$ is the virtual electric field corresponding to the virtual potential $\delta \phi$.

The constitutive equations to govern piezoelectricity are written for the mechanical behavior as

$$\sigma = L^E : \epsilon - e \cdot E \quad (6-102)$$

and for the electrostatic behavior as

$$D = e^T : \varepsilon + \xi^\varepsilon \cdot E \quad (6-103)$$

where

L is the elastic stiffness

e is the piezoelectric matrix (stress based)

ξ is the permittivity

The superscripts E and ε represent coefficients measured at constant electric field, and constant strain, respectively. The term with e gives the electro-mechanical coupling in the two constitutive Equations 6-102 and 6-103.

We approximate the displacements and electrical potential within a finite element as

$$u = NU$$

and

$$\phi = N\Phi$$

where N contains the shape functions and U and Φ contain the nodal degrees of freedom. The body forces and charges, as well as the distributed loads and distributed charges, are interpolated in a similar manner.

The strains and the electric field are given as

$$\varepsilon = B_u U \quad (6-104)$$

and

$$E = B_\phi \Phi \quad (6-105)$$

where B_u and B_ϕ contain the gradients of N .

Denoting the virtual displacement by δU and the virtual potential by $\delta \Phi$ the variational formulations can be obtained by substituting Equations 6-102 and 6-104 into Equation 6-100:

$$\begin{aligned} \int_V \delta \varepsilon^T \sigma dV &= \int_V \delta U^T B_u^T \sigma dV = \int_S t \delta U dS + \int_V f \delta U dV \\ \int_V \delta U^T B_u^T L B_u U dV + \int_V \delta U^T B_u^T e B_\phi \Phi dV &= \int_S t \delta U dS + \int_V f \delta U dV \\ \delta U^T K_{uu} U + \delta U^T K_{u\phi} \Phi &= \delta U^T F_u \end{aligned} \quad (6-106)$$

and similarly by substituting Equations 6-103 and 6-105 into Equation 6-101:

$$\begin{aligned} \int_V \delta E^T D dV &= \int_V -\delta \Phi^T B_\phi^T D dV = \int_S \delta \Phi D n dS + \int_V \rho_V \delta \Phi dV \\ -\int_V \delta \Phi^T B_\phi^T e B_u U dV + \int_V \delta \Phi^T B_\phi^T \xi B_\phi \Phi dV &= \int_S \delta \Phi D n dS + \int_V \rho_V \delta \Phi dV \\ -\delta \Phi^T K_{\phi u} U + \delta \Phi^T K_{\phi\phi} \Phi &= \delta \Phi^T \rho_\phi \end{aligned} \quad (6-107)$$

The final set of equations in matrix form is then

$$\begin{bmatrix} K_{uu} & K_{u\phi} \\ -K_{\phi u} & K_{\phi\phi} \end{bmatrix} \begin{bmatrix} U \\ \Phi \end{bmatrix} = \begin{bmatrix} F_u \\ \rho_\phi \end{bmatrix} \quad (6-108)$$

Strain Based Piezoelectric Coupling

It is also possible to apply strain based coefficients for the piezoelectric coupling matrix. Then the constitutive equations are for the mechanical behavior

$$\varepsilon = C:\sigma + d \cdot E \quad (6-109)$$

and for the electrostatic behavior

$$D = d:\sigma + \xi^* \cdot E \quad (6-110)$$

where

C is the elastic compliance

d is the piezoelectric matrix (strain based)

ξ^* is the permittivity (strain based)

When d and ξ^* are given, MSC.Marc converts this into stress based properties,

where $e = L:d$ and $\xi = \xi^* - e^T \cdot d$.

Acoustic Analysis

MSC.Marc has the capability to perform acoustic analysis in a rigid as well as a deformable cavity. This allows the program to calculate the fundamental frequencies of the cavity, as well as the pressure distribution in the cavity. This can be solved for two- or three-dimensional fields.

Rigid Cavity Acoustic Analysis

The acoustic problem with rigid reflecting boundaries is a purely linear problem analogous to dynamic analysis, but using the heat transfer elements.

The elements which are available for acoustic analysis of a rigid cavity are described in [Table 6-9](#).

Table 6-9 Element Types for Rigid Cavity Acoustic Analysis

Element Type	Description
37 , 39 , 131 , 41	3-, 4-, 6-, 8-node planar
69	8-node reduced integration planar
121	4-node reduced integration planar
101 , 103	6-, 9-node semi-infinite planar
38 , 40 , 132 , 42	3-, 4-, 6-, 8-node axisymmetric
70	8-node reduced integration axisymmetric
122	4-node reduced integration axisymmetric
133	10-node tetrahedral
135	4-node tetrahedral

MSC.Marc computes and prints the following quantities: pressure and pressure gradient at the integration points. The nodal point data consists of the pressure and the source.

Technical Background

The wave equation in an inviscid fluid can be expressed in terms of the pressure p as

$$c^{\frac{1}{2}} \frac{\partial^2 p}{\partial t^2} = \nabla^2 p \tag{6-111}$$

where c is the sonic velocity

$$c = \sqrt{K/\rho} \quad (6-112)$$

where K is the bulk modulus and ρ is the density.

Equation 6-111 can be rewritten as

$$K\nabla^2 p - \rho \frac{\partial^2 p}{\partial t^2} = 0 \quad (6-113)$$

Where the source terms are neglected, note that this is analogous to the dynamic equation of motion.

The modeling of rigid reflecting boundaries can be done as follows.

Mathematically, a reflecting boundary is described by:

$$\frac{\partial p}{\partial n} = 0 \quad (6-114)$$

where $\frac{\partial p}{\partial n}$ is the pressure gradient normal to the reflecting surface.

This is analogous to an insulated boundary in heat transfer. Hence, a reflecting boundary can be modeled by a set of nodes at the outer surface of the area which are not connected to another part of the medium. A reflecting plate in the middle of an acoustic medium can be modeled by double nodes at the same location

Note: Where there are no boundary conditions applied, there is a zero-valued eigenvalue, corresponding to a constant pressure mode. Hence, you need to have a nonzero initial shift point.

To activate the acoustic option, use the **ACOUSTIC** parameter. The number of modes to be extracted should also be included on this parameter. The bulk modulus and the density of the fluid are given in the **ISOTROPIC** option.

Specify nodal constraints using the **FIXED PRESSURE** option. Input nodal sources using the **POINT SOURCE** option. Specify distributed sources using the **DIST SOURCES** option. If nonuniform sources are required, apply these via user subroutine **FLUX**.

To obtain the fundamental frequencies, use the **MODAL SHAPE** option after the **END OPTION**. The nodes can be used in a transient analysis by invoking the **DYNAMIC CHANGE** option. The point and distributed sources could be a function of time.

Coupled Acoustic-Structural Analysis

In a coupled acoustic-structural analysis, both the acoustic medium and the structure are modeled. In this way, the effect of the acoustic medium on the dynamic response of the structure and of the structure on the dynamic response of the acoustic medium can be taken into account. Such a coupled analysis is especially important when the natural frequencies of the acoustic medium and the structure are in the same range. In MSC.Marc, only a harmonic coupled acoustic-structural analysis can be performed. Since the interface between the acoustic medium and the structure is determined automatically by MSC.Marc based on the **CONTACT** option, setting up the finite element model is relatively easy since the meshes do not need to be identical at the interface.

Technical Background

The acoustic medium will be called the fluid, although it might also be a gas, and is considered to be inviscid and compressible. The equilibrium equation is given by:

$$\nabla p + \rho_f \ddot{u} = 0 \quad (6-115)$$

in which p is the pressure, ρ_f is the fluid density, u is the displacement vector and a superposed dot indicates differentiation with respect to time.

Using the bulk modulus of the fluid K_f , the constitutive behavior can be written as:

$$p = -K_f \nabla u \quad (6-116)$$

The Equations (6-115) and (6-116) can be combined to:

$$\frac{1}{K_f} \ddot{p} - \frac{1}{\rho_f} \nabla^2 p = 0 \quad (6-117)$$

It is also possible to add a damping term $r\dot{u}$ to Equation (6-115), with r the resistivity or fluid drag:

$$\nabla p + r\dot{u} + \rho_f \ddot{u} = 0 \quad (6-118)$$

Since we restrict the discussion to harmonic analyses, we can write $\dot{u} = i\omega u$, with ω the excitation frequency and i the imaginary unit, so that Equation (6-118) reduces to:

$$\frac{1}{K_f} \ddot{p} - \frac{1}{\bar{\rho}_f} \nabla^2 p = 0 \quad (6-119)$$

in which the complex density $\bar{\rho}_f$ is given by:

$$\bar{\rho}_f = \left(\rho_f - \frac{ir}{\omega} \right) \quad (6-120)$$

The weak form of Equation (6-119) is obtained in a standard way by introducing the variational field δp and integrating over the fluid volume V :

$$\int_V \delta p \left(\frac{1}{K_f} \ddot{p} - \frac{1}{\bar{\rho}_f} \nabla^2 p \right) dV = 0 \quad (6-121)$$

Applying the divergence theorem of Green to this expression yields:

$$\int_V \left(\delta p \frac{1}{K_f} \ddot{p} + \frac{1}{\bar{\rho}_f} \nabla \delta p \nabla p \right) dV + \int_{\Gamma} \frac{1}{\bar{\rho}_f} \delta p \frac{\partial p}{\partial n} d\Gamma = 0 \quad (6-122)$$

in which Γ represents the boundary of the fluid with an inward normal n .

Along the boundary, various conditions can occur:

- The pressure p is prescribed on the boundary Γ_p .
- The normal acceleration is prescribed to be a_n on the accelerating boundary Γ_a .
- The normal acceleration of the fluid equals the normal acceleration of the solid, $\ddot{u}_{fn} = \ddot{u}_{sn}$, on the fluid-structure interface Γ_{fs} .
- Nonreflecting or reactive boundary conditions are introduced using a spring-dashpot analogy on Γ_{fn} as $\frac{1}{\bar{\rho}_f} \frac{\partial p}{\partial n} = \dot{p}/c_I + \ddot{p}/k_I$. In this way, a spring and a dashpot are placed in series between the acoustic medium and its boundary, with k_I the spring and c_I the dashpot parameter, both per unit area. Some classical conditions can be modeled by this expression:

1. Free surfaces waves in a gravity field: $\frac{\partial p}{\partial n} = \ddot{p}/g$, with g the gravity acceleration;

2. The plane wave radiation boundary condition: $\frac{\partial p}{\partial n} = \dot{p}/c$, with c the wave velocity;
3. Using the complex admittance $1/z(\omega)$ of the boundary, with $z(\omega)$ the impedance, the normal velocity can be related to the pressure by $\dot{u}_{fn} = p/z(\omega) = (1/c_I + i\omega/k_I)p$, so that a given $z(\omega)$ can be modeled by setting $1/c_I$ and $1/k_I$.

Upon discretizing the fluid and the solid using finite elements, the fluid pressure and the displacements of the solid can be approximated in a standard way, so that Equation (6-122) yields:

$$\begin{aligned} \int_V \left\{ \delta p^T N^p \frac{1}{K_f} N^{pT} \ddot{p} + \delta p^T (\nabla N^p) \frac{1}{\rho_f} (\nabla N^{pT}) p \right\} dV - \int_{\Gamma_a} \delta p^T N^p a_n d\Gamma - \int_{\Gamma_{fs}} \delta p^T N^p N^{uT} G \ddot{u} d\Gamma \\ + \int_{\Gamma_{fn}} \delta p^T \left(N^p \frac{1}{c_I} N^{pT} \dot{p} + N^p \frac{1}{k_I} N^{pT} \ddot{p} \right) d\Gamma = 0 \end{aligned} \quad (6-123)$$

where p and u contain the pressure and displacement degrees of freedom, N contains the interpolation functions and G is a transformation matrix to relate the displacements in normal direction to the global displacement components. Since Equation (6-123) must be valid for arbitrary admissible values of δp , evaluating the various integrals and using $\dot{p} = i\omega p$, $\ddot{p} = -\omega^2 p$ and $\ddot{u} = -\omega^2 u$, results in:

$$\{K_f + i\omega C_f - \omega^2 M_f\} p + \omega^2 S_{fs} u = F_f \quad (6-124)$$

In order to end up with a set of coupled equations, we need to consider the solid as well. The effect of the fluid on the solid originates from the fluid pressure at the fluid-solid interface. This can be easily included by evaluating the virtual work corresponding to the fluid pressure:

$$\delta W^p = \int_{\Gamma_{fs}} -\delta u_{ns} p d\Gamma = \int_{\Gamma_{fs}} -\delta u^T G^T N^u N^{pT} p d\Gamma = -\delta u^T S_{fs}^T p \quad (6-125)$$

Notice that the minus sign has been used to express that u_{ns} is positive in the outward direction of the solid. Including equation Equation 6-125, the solid behavior for a harmonic analysis is governed by:

$$\{K_s + i\omega C_s - \omega^2 M_s\}u + S_{fs}^T p = F_s \quad (6-126)$$

with K_s the stiffness matrix, which can include stress-stiffening, C_s is the damping matrix, M_s the mass matrix, and F_s the external load vector, except for the fluid pressure.

Premultiplying Equation 6-124 by ω^{-2} and combining the result with equation Equation 6-126 gives the desired coupled complex equation system:

$$\begin{bmatrix} \omega^{-2} A_f & S_{fs} \\ S_{fs}^T & A_s \end{bmatrix} \begin{bmatrix} p \\ u \end{bmatrix} = \begin{bmatrix} \omega^{-2} F_f \\ F_s \end{bmatrix} \quad (6-127)$$

in which $A_f = K_f + i\omega C_f$ and $A_s = K_s + i\omega C_s$.

The procedure to perform a coupled acoustic-structural is as follows. The acoustic medium and the structure are modeled separately; the acoustic structure using heat transfer elements with acoustic material properties and the structure using conventional stress elements. The elements representing the acoustic medium are assigned to an *acoustic contact body* and the elements representing the solid to a *deformable contact body*. If the acoustic medium is not surrounded completely by a deformable structure and one wants to, for example, model radiation boundary conditions along this part of the acoustic medium, then one can define *rigid contact bodies* in this area. Like deformable bodies, rigid bodies can be used to define the spring-dashpot analogy defined before. A harmonic analysis can be performed based on initial contact, so using the undeformed configuration of the contact bodies, but also after a preload of the deformable contact bodies. In the latter case, it might be necessary to remesh the acoustic and/or deformable bodies before the harmonic analysis can be performed, since during the static loading of the deformable bodies the deformation of the acoustic bodies is not taken into account.

The elements which are available for coupled-structural acoustic analysis are described in Table 6-10.

Table 6-10 Element Types for Coupled-Structural Acoustic Analysis

Element Type	Description
37, 39	3-, 4-node planar
121	4-node reduced integration planar
38, 40	3-, 4-node axisymmetric
122	4-node reduced integration axisymmetric
135	4-node tetrahedral

MSC.Marc computes and prints the following quantities: pressure and pressure gradient at the integration points. The nodal point data consists of the pressure and the source.

The **ACOUSTIC** parameter is used to indicate that a coupled acoustic-structural analysis is performed. In addition to the **CONTACT** option, the model definition options **ACOUSTIC** and **REGION** are used to define the material properties of the acoustic medium and to set which elements correspond to the solid and the fluid region. Typical boundary conditions for the acoustic medium can be entered using the **FIXED PRESSURE** model definition and **PRESS CHANGE** history definition option. The **DIST SOURCES** and **POINT SOURCE** model definition and **DIST SOURCES** and **POINT SOURCE** history definition options are used to define the load on the acoustic medium. The **HARMONIC** history definition option is used to define the excitation frequencies.

Fluid Mechanics

MSC.Marc has the capability to perform fluid flow analysis. MSC.Marc solves the Navier-Stokes equations in the fluid under the restrictions that the fluid is considered to be nonreactive, incompressible, single phases, and laminar. The capabilities in MSC.Marc can be applied to four different problems listed below:

- Fluid behavior only
- Fluid-thermal coupled behavior
- Fluid-solid coupled behavior
- Fluid-thermal-solid coupled behavior

Mass Conservation

The principle of mass conservation for a single-phase fluid can be expressed in differential form as:

$$\frac{D\rho}{Dt} + \rho \nabla \cdot \mathbf{v} = 0 \quad (6-128)$$

where ρ is the mass density, $\mathbf{v}$ is the Eulerian fluid velocity, and t is the time.

Momentum Conservation

The principle of conservation of linear momentum results in:

$$\rho \left(\frac{\partial \mathbf{v}}{\partial t} + \mathbf{v} \cdot \nabla \mathbf{v} \right) = \rho \mathbf{f} + \nabla \cdot \boldsymbol{\sigma} \quad (6-129)$$

where $\boldsymbol{\sigma}$ is the stress tensor and $\mathbf{f}$ is the body force per unit mass.

In general, the stress tensor can be written as the sum of the hydrostatic stress $\hat{\sigma}$ and the deviatoric stress σ' .

$$\hat{\sigma} = \frac{1}{3} \sigma_{ij} \delta_{ij} \quad (6-130)$$

$$\sigma' = \sigma_{ij} - \frac{1}{3} \sigma_{ij} \delta_{ij} = \sigma_{ij} - \hat{\sigma} \delta_{ij} \quad (6-131)$$

In fluid mechanics, the fluid pressure is introduced as the negative hydrostatic pressure:

$$p = -\hat{\sigma} \quad (6-132)$$

Energy Conservation

For incompressible fluids, the principle of conservation of thermal energy is expressed by:

$$\rho C_p \left(\frac{\partial T}{\partial t} + v_i \frac{\partial T}{\partial x_i} \right) = - \frac{\partial q_i}{\partial x_i} + H \quad (6-133)$$

where T is the temperature, C_p is the specific heat at constant pressure, q_i is the thermal flux, and H is the internal heat generated.

Equation of State

The equation of state for a homogeneous, single-phase gas is:

$$pV = \frac{m\bar{R}T}{M} \quad (6-134)$$

m is the mass of the fluid, M is the molecular weight of the gas, $\bar{R}$ is the universal gas constant, V is the volume that the gas occupies.

The gas constant is often expressed as $R = \bar{R}/M$ for each substance. The equation of state is then written as:

$$\rho = \frac{P}{RT} \quad (6-135)$$

In MSC.Marc, it is assumed that the fluid is incompressible. In such cases, the density is constant $\rho = \rho_o$

Constitutive Relations

The shear strain rate tensor is defined:

$$\epsilon_{ij} = \frac{1}{2} \left(\frac{\partial v_i}{\partial x_j} + \frac{\partial v_j}{\partial x_i} \right) \quad (6-136)$$

For viscous incompressible fluids, it is assumed that there is an expression:

$$\sigma_{ij}' = 2\mu\dot{\epsilon}_{ij} = \mu\dot{\gamma}_{ij} \quad (6-137)$$

where μ is the dynamic viscosity.

If the viscosity is not a function of the strain rate, the material is considered Newtonian. Viscous fluid flow is usually characterized by the Reynolds number Re :

$$Re = \frac{\rho L v}{\mu} \quad (6-138)$$

where v is a typical velocity and L is the characteristic length.

Viscosity for common fluids like air and water is fairly constant over a broad range of temperatures. However, many materials have viscosity which is strongly dependent on the shear strain rate. These materials include glass, concrete, oil, paint, and food products. There are several non-Newtonian fluid models in MSC.Marc to describe the viscosity as described below.

Piecewise Non-Newtonian Flow

$$\dot{\epsilon} \text{ is the equivalent strain rate} = \dot{\epsilon} = \left(\frac{2}{3} \dot{\epsilon}_{ij} \dot{\epsilon}_{ij} \right)^{1/2} \quad (6-139)$$

$\mu = \mu(\dot{\epsilon})$, where μ is a piecewise linear function.

Bingham Fluid

A Bingham fluid behaves as a “rigid” fluid if the stress is below a certain level, labeled the yield stress, and behaves in a nonlinear manner at higher stress levels.

$$\sigma_{ij}' = \mu_o \dot{\gamma}_{ij} + g \dot{\gamma}_{ij} / \dot{\gamma} \quad \text{if } \bar{\sigma} \geq g \quad (6-140)$$

$$\dot{\gamma}_{ij} = 0 \quad \text{if } \bar{\sigma} < g \quad (6-141)$$

The effective viscosity is $\mu = \mu_o + g / \dot{\gamma}$ and $\dot{\gamma} = \left(\frac{1}{2} \dot{\gamma}_{ij} \dot{\gamma}_{ij} \right)^{1/2}$

The materials that can be simulated with this model are cement, slurries, and pastes.

Power Law Fluid

The fluid is represented as:

$$\sigma_{ij}' = \mu_o K (\dot{\gamma})^{n-1} \dot{\gamma}_{ij} \quad (6-142)$$

where K is a nondimensional constant.

This model is useful for simulating flow of rubber solutions, adhesives, and biological fluids.

Carreau Model

This model alleviates the difficulties associated with the power law model and accounts for the lower and upper limiting viscosities for an extreme value of the equivalent shear strain rate.

$$\mu = \mu_{\infty} + (\mu_o - \mu_{\infty}) (1 + \tau^2 \dot{\gamma}^2)^{(n-1)/2} \quad (6-143)$$

where μ_o and μ_{∞} are the viscosity at time equals zero and infinity, respectively.

The thermal flux is governed by the Fourier law.

$$q_i = k_{ij} \frac{\partial T}{\partial x_j} \quad (6-144)$$

Finite Element Formulation

In the procedures developed, the fluid-thermal coupling can be treated as weak or strong. In the weak formulation, the solution of the fluid and thermal equilibrium equations are solved in a staggered manner. In the strongly coupled approach, a simultaneous solution is obtained.

This section cannot cover all of the details of the finite element formulation. It does discuss the final form of the linearized set of equations and some of the consequences. Beginning with degrees of freedom in a system – namely, v , p , and T for the mixed method – we utilize the traditional finite element interpolation functions to relate the values within the element to the nodal values.

Because the pressure stabilizing Petrov-Galerkin (PSPG) method is employed in MSC.Marc, equal order interpolation functions can be used for the velocity and the pressure. The method of weighted residuals is used to solve the coupled Navier-Stokes equations. Based upon the conservation laws of momentum (mass and energy), we obtain first order differential equations in the form:

$$M\dot{v} + A(v)v + K(T, v)v - Cp + B(T)T = F(t) \quad (6-145)$$

$$C^T v = 0 \quad (6-146)$$

$$N\dot{T} + D(v)T + L(T)T = Q(v, t) \quad (6-147)$$

where the first equation is obtained from momentum, the second from mass, and the third from energy conservation, respectively. This is expressed in matrix form as:

$$\begin{bmatrix} M & 0 & 0 \\ 0 & N & 0 \\ 0 & 0 & 0 \end{bmatrix} \begin{Bmatrix} \dot{v} \\ \dot{T} \\ \dot{p} \end{Bmatrix} + \begin{bmatrix} A + K & B & -C \\ 0 & D + L & 0 \\ -C^T & 0 & 0 \end{bmatrix} \begin{Bmatrix} v \\ T \\ p \end{Bmatrix} = \begin{Bmatrix} F \\ Q \\ 0 \end{Bmatrix} \quad (6-148)$$

if the penalty method is employed

$$p = \chi M_p^{-1} C^T v \quad (6-149)$$

resulting in

$$\begin{bmatrix} M & 0 \\ 0 & N \end{bmatrix} \begin{Bmatrix} \dot{v} \\ \dot{T} \end{Bmatrix} + \begin{bmatrix} A + K + \chi C M_p^{-1} C^T & B \\ 0 & D + L \end{bmatrix} \begin{Bmatrix} v \\ T \end{Bmatrix} = \begin{Bmatrix} F \\ Q \end{Bmatrix} \quad (6-150)$$

The introduction of the streamline upwinding technique (SUPG) developed by Brooks and Hughes is a major improvement for the stability of the fluid equations. This procedure controls the velocity oscillations induced by the advection terms. Effectively, this procedure adds artificial viscosity to the true viscosity. The second stabilization method, PSPG, allows equal order velocity and pressure interpolation functions to be used without inducing oscillations in the pressure field. The PSPG term is not included when using the penalty formulation. The details of these

stabilization terms are not provided here, but note that the magnitude of the contributions are dependent upon the element size, viscosity, and the time step. Including these contributions, the semi-discrete set of equations takes the form:

$$\begin{bmatrix} M + M_\delta & 0 & 0 \\ M_\epsilon & N & 0 \\ 0 & 0 & 0 \end{bmatrix} \begin{Bmatrix} \dot{v} \\ \dot{T} \\ \dot{p} \end{Bmatrix} + \begin{bmatrix} A + K + K_\delta & B & -(C + C_\delta) \\ K_\epsilon & D + L & 0 \\ -C^T & 0 & C_\epsilon \end{bmatrix} \begin{Bmatrix} v \\ T \\ p \end{Bmatrix} = \begin{Bmatrix} F + F_\delta \\ Q + Q_\epsilon \\ 0 \end{Bmatrix} \quad (6-151)$$

The terms K_δ is included for SUPG and C_ϵ and G_ϵ for PSPG. The other terms are neglected, leading to:

$$\begin{bmatrix} M & 0 & 0 \\ 0 & N & 0 \\ 0 & 0 & 0 \end{bmatrix} \begin{Bmatrix} \dot{v} \\ \dot{T} \\ \dot{p} \end{Bmatrix} + \begin{bmatrix} A + K + K_\delta & B & -C \\ 0 & D + L & 0 \\ -C^T & 0 & C_\epsilon \end{bmatrix} \begin{Bmatrix} v \\ T \\ p \end{Bmatrix} = \begin{Bmatrix} F \\ Q \\ 0 \end{Bmatrix} \quad (6-152)$$

These submatrices can be interpreted as follows:

- A represents advection of momentum
- D represents advection of energy
- K represents diffusion of momentum or viscosity matrix
- L represents diffusion of energy or conductance matrix
- M represents mass
- N represents heat capacitance
- B represents buoyancy
- K_δ represents SPUG stabilization matrix
- C represents gradient matrix
- C_T represents divergence matrix
- C_ϵ represents PSPG stabilization matrix
- F represents externally applied forces
- Q represents externally applied fluxes

Note that typically D , K , B , L , Q , and F are dependent upon the temperature. If a non-Newtonian fluid is used, A , K , D , and G are dependent upon the rate of strain. If the fluid is subjected to large motions, F can also depend on the total displacement.

In terms of physical parameter, excluding geometry, observe:

$$A = A(\rho_o)$$

$$D = D(\rho_o, C_p) = D(\rho_o, C_p, T)$$

$$K = K(\mu) = K(\mu, T)$$

$$L = L(K) = L(K, T)$$

$$M = M(\rho_o)$$

$$N = N(\rho_o, C_p)$$

$$B = B(\rho_o, g, B)$$

where:

ρ_o is the initial density

C_p is the specific heat

K is the thermal conductivity

g is the gravity

B is the coefficient of thermal expansion

Penalty Method

The penalty method is an alternative method to satisfy the incompressibility constraints. The objective is to add another term to the operator matrix (viscosity matrix) such that incompressibility is satisfied. Effectively, K is replaced with K_c where:

$$K_c = K + \chi K_p \quad (6-153)$$

where χ is a large number and K_p the penalty matrix. K_p can be written as:

$$K_p = CM_p^{-1}C^T \quad (6-154)$$

where C and M_p are functions of the geometry and shape functions only. The value of χ is typically between 10^5 to 10^9 .

The penalty method should not be used for three-node planar elements, or four-node tetrahedral elements.

Steady State Analysis

It is possible to simplify the equations governing fluid flow by assuming that the time derivatives of the velocity and temperature are zero. This is achieved by using the **STEADY STATE** option. For thermally coupled, non-Newtonian flows, you still have a highly nonlinear system and multiple iterations are required.

Transient Analysis

The equations discussed above are still differential equations because of the presence of time derivatives of velocity and temperature. The conversion of the equations to fully discrete linear equations requires an assumption of the behavior during the increment. The approximations inevitably result in problems with accuracy, artificial damping, and/or stability problems.

MSC.Marc uses the first order backward Euler procedure such that:

$$\dot{v}_{n+1} = (v_{n+1} - v_n)/\Delta t \quad (6-155)$$

This is substituted into the equations to yield:

$$\begin{bmatrix} A + K + K_\delta + M/\Delta t & B & -C \\ 0 & D + L + N/\Delta t & 0 \\ -C^T & 0 & C_\epsilon \end{bmatrix} \begin{Bmatrix} v \\ T \\ p \end{Bmatrix} = \begin{Bmatrix} F - M/\Delta t \cdot v_n \\ Q + Q_\epsilon - N/\Delta t \cdot T_n \\ 0 \end{Bmatrix} \quad (6-156)$$

MSC.Marc allows either fixed time step or adaptive time step procedure.

The equations of fluid flow are highly nonlinear even for Newtonian behavior because of the inclusion of the advection terms. The Newton-Raphson and direct substitution procedures are available to solve these problems.

Solid Analysis

Solid can be modeled in two ways:

- The first is to model them as a fluid with a very large viscosity. In such cases, an Eulerian procedure is used throughout the model. The constitutive laws are limited to the fluid relationships.
- The second is to model them as true solids. In such cases, a Lagrangian or updated Lagrangian approach is used in the solid.

The complete MSC.Marc constitutive routines are available for representing the solid behavior.

An important consideration is the interface between the fluid and solid. In the case of large motion at the interface, it might be necessary to remesh the fluid region. This is not supported in the K7 release. For small changes in the structural deformation, the nodes at the interface are updated. The pressure is transmitted between the fluid and the solid.

Solution of Coupled Problems in Fluids

The coupled fluid-thermal problem can be solved using either a tightly coupled procedure as shown above or in a staggered manner. Problems such as free convection are inherently coupled and are best solved using the tightly coupled procedure.

The fluid-solid interaction problem is solved in a weakly coupled manner. Every attempt has been made to maximize the computational efficiency. It should be noted that, while the staggered procedure can result in more global iterations, there are several motivating factors for solving the system in this manner.

- In solving fluid problems, especially in three dimensions, very large systems of equations are obtained. Any procedure that reduces the number of equations (for example, excluding the solid) is potentially beneficial.
- The fluid flow solution always requires the solution of a nonsymmetric system. In a weakly staggered analysis, the structural problem can still be solved with a symmetric solver.
- When using an iterative solver, the solution of the strongly coupled system will result in an ill-conditioned system, which results in poor convergence. This is because the terms associated with the fluid, thermal, and structural operator are of several different orders of magnitude.

The problem can be divided into two regions (fluid and solid) to accommodate these requirements. In each region, a different solver can be invoked; different nodal optimizers can be used; and a different memory allocation can be performed. Often, the fluid uses an out-of-core nonsymmetric solver while the structure uses an in-core symmetric solver.

Degrees of Freedom

The degrees of freedom in a fluid analysis are the velocities or, for the mixed method, the velocity and the pressure. When the pressure is not included explicitly as a degree of freedom, the incompressibility constraint is imposed using a penalty approach.

If a strongly coupled fluid-thermal problem is solved, the degrees of freedom are either the velocities and temperatures or the velocities, pressure, and temperature. For a three dimensional problem, the number of degrees of freedom could be 5.

The degrees of freedom associated with the solid are the conventional displacements or the displacements and temperatures.

Element Types

The fluid region can be represented using the conventional displacement elements in MSC.Marc. When using the mixed method, the pressure has the same order and interpolation functions as the velocity. In a coupled fluid-thermal analysis, the temperature also has the same order as the velocity. Hence, the element types available are:

Planar

		<u>Type</u>
3-node	linear	3
4-node	isoparametric bilinear	11
4-node	isoparametric bilinear reduced integration	115
6-node	isoparametric triangle	125
8-node	isoparametric biquadratic	27
8-node	isoparametric biquadratic reduced integration	54

Axisymmetric

		<u>Type</u>
3-node	linear	2
4-node	isoparametric bilinear	10
4-node	isoparametric bilinear reduced integration	116
6-node	isoparametric triangle	126
8-node	isoparametric biquadratic	28
8-node	isoparametric biquadratic reduced integration	55

Three Dimensional

		<u>Type</u>
4-node	tetrahedron	134
8-node	trilinear brick	7
8-node	trilinear brick with reduced integration	117
20-node	brick	21
20-node	brick with reduced integration	57

The three-node triangular elements or the four-node tetrahedral element give poor results when used with the penalty formulation.

The shape functions used are identical to those used for structural analyses and can be found in any finite element textbook.

All of the mesh generation capabilities in MSC.Marc Mentat can be used to generate the fluid mesh. Furthermore, MSC.Marc Mentat can be used to visualize the results in a manner consistent with structural analyses.

MSC.Marc's implementation of the Navier-Stokes equations utilizes the natural boundary condition.

At nodal points, you can prescribe the time dependent values of the velocity and temperature. This can be achieved by using the [FIXED VELOCITY](#), [FIXED TEMPERATURE](#), [VELOCITY CHANGE](#), and/or [TEMP CHANGE](#) options. User subroutine [FORCDT](#) can be used for complex time-dependent behavior.

The excitation can be prescribed at the nodes as either time-dependent force or flux. This can be achieved using [POINT FLUX](#). Again, user subroutine [FORCDT](#) can be used for complex time-dependent behavior. Additionally, for two-dimensional problems, edge pressures or edge fluxes can be defined; for three-dimensional problems, surface fluxes or surface pressures can be defined. These quantities are numerically integrated using the shape functions.

Gravity and centrifugal loads can also be applied.

For coupled fluid-solid interaction problems, the pressure of the fluid is automatically applied to the structure. The resultant deformation of the structure is applied to the fluid boundaries. In the current release, only small deformations of the solid are permitted.

The viscosity, mass density, conductivity, and specific head are defined through the [ISOTROPIC](#) option. The [STRAIN RATE](#) option is used to define non-Newtonian viscous behavior. The penalty parameter can be entered through the [PARAMETERS](#) option.

Note: In fluid analysis, data for [POINT LOAD](#) and [DIST LOADS](#) should be prescribed as total rather than incremental quantity (as used in mechanical analysis). Similarly, [POINT FLUX](#) and [DIST FLUXES](#) for heat transfer analysis are also given as total quantity. This specification is to be used consistently for the fluid and/or heat transfer portion of analysis in coupled fluid-solid, fluid-thermal, and fluid-thermal-solid.

Coupled Analyses

The definition of coupled systems includes the multiple domains and independent or dependent variables describing different physical systems. In the situation with multiple domains, the solution for both domains is obtained simultaneously. Similarly, the dependent variables cannot be condensed out of the equilibrium equations explicitly.

Coupled systems can be classified into two categories:

1. **Interface variables coupling:** In this class of problems, the coupling occurs through the interfaces of the domain. The domains can be physically different (for example, fluid-solid interaction) or physically the same but with different discretization (for example, mesh partition with explicit/implicit procedures in different domains).

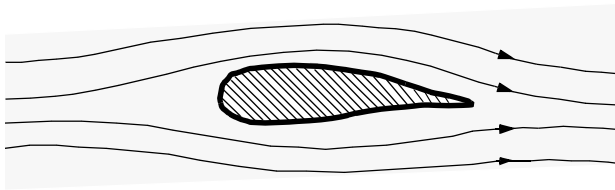


Figure 6-9 Fluid-structure Interaction (Physically Different Domains)

2. **Field variables coupling:** In this, the domain can be the same or different. The coupling occurs through the governing differential equations describing different physical phenomenon; for example, coupled thermo-mechanical problems.

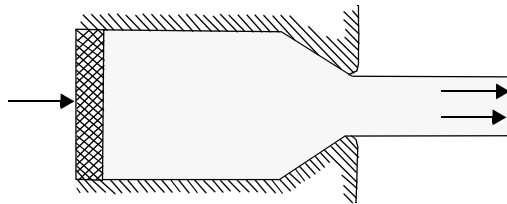


Figure 6-10 Metal Extrusion with Plastic Flow Coupled with Thermal Field

MSC.Marc can solve the following types of coupled problems: fluid-solid, fluid-thermal, fluid-solid-thermal interaction, piezoelectric, thermo-electrical (Joule heating), thermo-mechanical, electrical-thermal-mechanical (Joule Mechanical), fluid-soil (pore pressure), and electromagnetic. The type of coupling is summarized in [Table 6-11](#).

Table 6-11 Summary of Coupled Procedures

Interaction	Category	Coupling
Thermal- Mechanical	Field	Weak
Fluid-Solid added mass approach general approach	Interface Interface	Weak Weak
Fluid-Thermal channel approach general approach	Interface Field	Weak Strong or Weak
Fluid-Thermal-Solid	Field/Interface	Strong or Weak/Weak
Piezoelectric	Field	Strong
Thermal-Electric (Joule)	Field	Weak
Electrical-Thermal- Mechanical	Field	Weak
Fluid-Pore	Field	Strong
Electromagnetic	Field	Strong

There are two approaches for solving fluid-solid interaction problems. In the first approach, the fluid is assumed to be inviscid and incompressible. The effect of the fluid is to augment the mass matrix of a structure. Modal shapes can be obtained for a fluid/solid system; the modal superposition procedure predicts the dynamic behavior of the coupled fluid/solid system. This prediction is based on extracted modal shapes. This method is discussed below.

The general fluid-solid capability models nonlinear transient behavior and is discussed in the fluid mechanics section.

There are two approaches for solving fluid-thermal interaction problems. The first is for a fluid constricted to move through thin areas and is implemented using the [CHANNEL](#) option. This is discussed earlier in this chapter. The second approach solves the coupled Navier-Stokes equations and is discussed in the fluid mechanics section.

The fluid-thermal-mechanical capability is described in more detail in the fluid mechanics section.

In the coupled thermo-electrical problem, the coupling takes place through the temperature-dependent electrical conductivity in the electrical problem and the internal heat generation caused by electrical flow in the thermal problem. The program solves for the voltage and temperature distribution.

Similarly, the coupling between the thermal and mechanical problems takes place through the temperature-dependent material properties in the mechanical (stress) problem and the internal heat generation in the mechanical problem caused by plastic work, which serves as input for the heat transfer problem. The temperature distribution and displacements are obtained.

In each of the coupled problems described above, two analyses are performed in each load/time increment. Iterations can also be carried out within each increment to improve the convergence of the coupled thermo-electrical and thermo-mechanical solutions.

In the coupled fluid-soil model, the fluid is assumed to be inviscid and incompressible. The effect of the fluid is to augment the stress in the soil material to satisfy equilibrium, and to influence the soil's material behavior.

In the electromagnetic analysis, the fully coupled Maxwell's equations are solved.

In the latter two analyses, the equations (fluid flow/structural or electrical/magnetic) are solved simultaneously.

Thermal Mechanically Coupled Analysis

Many operations performed in the metal forming industry (such as casting, extrusion, sheet rolling, and stamping) can require a coupled thermo-mechanical analysis. The observed physical phenomena must be modeled by a coupled analysis if the following conditions pertain:

- The body undergoes large deformations such that there is a change in the boundary conditions associated with the heat transfer problem.
- Deformation converts mechanical work into heat through an irreversible process which is large relative to other heat sources.

In either case, a change in the temperature distribution contributes to the deformation of the body through thermal strains and influences the material properties.

MSC.Marc has a capability that allows you to perform mechanically coupled analysis. This capability is available for all stress elements and for small displacement, total Lagrangian, updated Lagrangian, or rigid plastic analysis.

The **COUPLE** parameter is used to invoke this option. When defining the mesh, if you specify the element as a stress type through the **CONNECTIVITY** option, MSC.Marc generates an associated heat transfer element, if possible. The region having an associated heat transfer element has coupled behavior. If you specify the element as a heat transfer type through the **CONNECTIVITY** option, that region is considered rigid. Only heat transfer is performed in that region.

MSC.Marc uses a staggered solution procedure in coupled thermo-mechanical analysis. It first performs a heat transfer analysis, then a stress analysis. Use the **CONTROL** option to enter the control tolerances used in the analysis.

Depending on the type of stress analysis performed, MSC.Marc can perform three different types of coupled analysis:

- Quasi-static coupled analysis: This comprises of a transient heat transfer pass and a static mechanical pass. Fixed stepping can be specified using **TRANSIENT NON AUTO** and adaptive stepping can use **TRANSIENT** or **AUTO STEP**. **AUTO STEP** is preferred over the **TRANSIENT** option.
- Creep coupled analysis: This comprises of a transient heat transfer pass and a creep mechanical pass. Fixed stepping can be specified using **CREEP INCREMENT** and adaptive stepping can use **AUTO CREEP** or **AUTO STEP**.
- Dynamic coupled analysis: This comprises of a transient heat transfer pass and a dynamic mechanical pass. Fixed stepping can be specified using **DYNAMIC CHANGE** and adaptive stepping can use **AUTO STEP**.

Load control and time step control can be specified in either of two ways:

- A fixed time step/load size can be specified by using the **TRANSIENT NON AUTO** option, the **CREEP INCREMENT** option, or the **DYNAMIC CHANGE** option. In these cases, mechanical loads and deformations are incremental quantities that are applied to each step. Fluxes are total quantities. Time variations for the mechanical and thermal loads can be specified using appropriate **s**.
- An adaptive time step/load size can be specified by using the **AUTO CREEP** option or the **AUTO STEP** option. In this case, mechanical loads and deformations are entered as the total quantities that are applied over the load set. Fluxes are total quantities. By default, mechanical loads are linearly increased while thermal loads are applied instantaneously.
- Exercise caution when applying boundary conditions. Use the **FIXED DISP** or **FIXED TEMPERATURE** options as well as **DISP CHANGE** or **TEMP CHANGE** for mechanical or thermal boundary conditions, respectively.

There are two primary causes of coupling. First, coupling occurs when deformations result in a change in the associated heat transfer problem. Such a change can be due to either large deformation or contact. Large deformation effects are coupled into

the heat transfer problem only if the **UPDATE** parameter is invoked. The gap element in MSC.Marc (Type 12) has been modified so that if no contact occurs, the gap element acts as a perfect insulator. When contact does occur, the gap element acts as a perfect conductor.

The second cause of coupling is heat generated due to inelastic deformation. The irreversibility of plastic flow causes an increase in the amount of entropy in the body. This can be expressed as

$$T\dot{S} = f\dot{W}^p \quad (6-157)$$

where $f\dot{W}^p$ is the fraction of the rate of plastic work dissipated into heat. Farren and Taylor measure f as approximately 0.9 for most metals. Using the mechanical equivalent of heat (M), the rate of specific volumetric flux is

$$R = Mf\dot{W}^p/\rho \quad (6-158)$$

Use the **CONVERT** model definition option to define Mf .

Of course, all mechanical and thermal material properties can be temperature dependent. The governing matrix equations can be expressed as

$$M\ddot{u} + D\dot{u} + K(T,u,t)u = F \quad (6-159)$$

$$C^T(T)\dot{T} + K^T(T)T = Q + Q^I + Q^F \quad (6-160)$$

In Equation 6-160, Q^I represents the amount of heat generated due to plastic work and Q^F represents the amount of heat generated due to friction. The specific heat matrix C^T and conductivity matrix K^T can be evaluated in the current configuration if the updated Lagrange option is used.

Note: All terms except M can be temperature dependent.

Fluid/Solid Interaction – Added Mass Approach

The fluid/solid interaction procedure investigates structures that are either immersed in, or contain a fluid. Examples of problems that make use of this feature are vibration of dams, ship hulls, and tanks containing liquids.

MSC.Marc is capable of predicting the dynamic behavior of a structural system that is subject to the pressure loading of fluid. The fluid is assumed to be inviscid and incompressible; for example, water.

In MSC.Marc, the fluid is modeled with heat transfer elements (potential theory) and the structure is modeled with normal stress or displacement elements. The element choice must ensure that the interface between the structural and fluid models has compatible interpolation; that is, both solid and fluid elements are either first order or second order. The **TYING** option can be used to achieve compatibility if necessary. To identify the interfaces between the fluid and the structure, the **FLUID SOLID** model definition set is necessary.

During increment zero, MSC.Marc calculates the stiffness matrix for the structure and the mass matrix for the structure augmented by the fluid effect. MSC.Marc then extracts the eigenvalues of the coupled system using the **MODAL SHAPE** option. The modal superposition procedure can then be used to predict the time response of the coupled system. **DYNAMIC CHANGE** can be used to perform modal superposition.

To input properties of solids and the mass density of the fluid elements, use the model definition option **ISOTROPIC**.

The calculation of the structural mass augmentation requires triangularization of the fluid potential matrix: this matrix is singular, unless the fluid pressure is fixed at least at one point with the **FIXED DISP** option.

Technical Background

In a fluid/solid interaction problem, the equations of motion can be expressed as

$$M_s a + K u = \frac{1}{\rho_f} S^T p \quad (6-161)$$

The pressure vector p can be calculated from

$$-S a = H p \quad (6-162)$$

The matrices in Equation 6-161 and Equation 6-162 are defined as

$$M_s = \int_V N_i^\beta N_i^\alpha \rho_s dV \quad (6-163)$$

$$K = \int_V \beta_{ij}^\beta D_{ijkl} \beta_{kl}^\alpha dV \quad (6-164)$$

$$S = \int_{A^T} R^\beta \rho_f n_i N_i^\alpha dA \quad (6-165)$$

$$H = \int_{V^f} \frac{\partial R^\beta}{\partial x_i} \frac{\partial R^\alpha}{\partial x_i} dV \quad (6-166)$$

where

ρ_f and ρ_s	are mass densities of the fluid and solid, respectively
a	is the acceleration vector
u	is the displacement vector
β_{ij}	is the strain displacement relation
D_{ijkl}	is the material constitutive relation
N_i	is the displacement interpolation function
R	is the pressure interpolation function
n_i	is the outward normal to the surface with fluid pressure p
A^T	is the surface on which the fluid pressure acts

In the present case, the fluid is assumed to be incompressible and inviscid. Only infinitesimal displacements are considered during the fluid vibration, so that the Eulerian and material coordinates coincide.

Substituting [Equation 6-162](#) into [Equation 6-161](#), we obtain

$$\begin{aligned} (M_s + S^T H^{-1} S / \rho_f) a + K u &= 0 \\ \text{or} \\ \bar{M} a + K u &= 0 \end{aligned} \quad (6-167)$$

This equation now allows the modes and frequencies of the solid structure immersed in the fluid to be obtained by conventional eigenvalue methods.

Coupled Thermal-Electrical Analysis (Joule Heating)

The coupled thermal-electrical analysis procedure can be used to analyze electric heating problems. The coupling between the electrical problem and the thermal problem in a Joule heating analysis is due to the fact that the resistance in the electric problem is dependent on temperatures, and the internal heat generation in the thermal problem is a function of the electrical flow.

MSC.Marc analyzes coupled thermo-electrical (Joule heating) problems. Use the parameter **JOULE** to initiate the coupled thermo-electrical analysis. This capability includes the analysis of the electrical problem, the associated thermal problem, and the coupling between these two problems.

The electrical problem is a steady-state analysis and can involve current and/or voltage boundary conditions as well as temperature-dependent resistivity.

The thermal analysis is generally a transient analysis with temperature-dependent thermal properties and time/temperature-dependent boundary conditions.

Use the model definition options **ISOTROPIC**, **ORTHOTROPIC**, **TEMPERATURE EFFECTS**, and **ORTHO TEMP** to input reference values of thermal conductivity, specific heat, mass density, and electrical resistivity, as well as their variations with temperatures. The mass density must remain constant throughout the analysis.

Use the model definition option **VOLTAGE** for nodal voltage boundary conditions, and the model definition options **POINT CURRENT** and/or **DIST CURRENT** for current boundary conditions. No initial condition is required for the electrical problem, since a steady-state solution is obtained.

In the thermal problem, you can use the model definition options **INITIAL TEMP**, **FILMS**, **POINT FLUX**, **DIST FLUXES**, and **FIXED TEMPERATURE** to prescribe the initial conditions and boundary conditions. Use the user subroutines **FILM** and **FLUX** for complex convective and flux boundary conditions. To enter the unit conversion factor between the electrical and thermal problems, use the model definition option **JOULE**. MSC.Marc uses this conversion factor to compute heat flux generated from the current flow in the structure.

Use the history definition options **STEADY STATE**, **TRANSIENT**, **POINT CURRENT**, **DIST CURRENT**, **VOLTAGE CHANGE**, **POINT FLUX**, **DIST FLUXES**, and **TEMP CHANGE** for the incrementation and change of boundary conditions.

A weak coupling between the electrical and thermal problems is assumed in the coupled thermo-electrical analysis, such that the distributions of the voltages and the temperatures of the structure can be solved separately within a time increment. A steady-state solution of the electrical problem (in terms of nodal voltages) is calculated first within each time step.

The heat generation due to electrical flow is included in the thermal analysis as an additional heat input.

The temperature distribution of the structure (obtained from the thermal analysis) is used to evaluate the temperature-dependent resistivity, which in turn is used for the electrical analysis in the next time increment.

Technical Background

In the coupled thermo-electrical analysis, the matrix equation of the electrical problem can be expressed as

$$K^E(T)V = I \quad (6-168)$$

and the governing equation of the thermal problem is

$$C^T(T)\dot{T} + K^T(T)T = Q + Q^E \quad (6-169)$$

In [Equation 6-168](#) and [Equation 6-169](#):

K^E	is the temperature-dependent electrical conductivity matrix
I	is the nodal current vector
V	is the nodal voltage vector
$C^T(T)$ and $K^T(T)$	are the temperature-dependent heat capacity and thermal conductivity matrices, respectively
T	is the nodal temperature vector
$\dot{T}$	is the time derivative of the temperature vector
Q	is the heat flux vector
Q^E	is the internal heat generation vector caused by the current flow.

The coupling between the electrical and thermal problems is through terms K^E and Q^E in [Equation 6-168](#) and [Equation 6-169](#).

The selection of the backward difference scheme for the discretization of the time variable in [Equation 6-169](#) yields the following expression:

$$\frac{1}{\Delta t}[C^T(T)] + K^T(T)T_n = Q_n + Q_n^E + \frac{1}{\Delta t}C^T(T)T_{n-1} \quad (6-170)$$

[Equation 6-170](#) is used for the computation of nodal temperatures in each time increment Δt .

The internal heat generation vector Q^E is computed from

$$Q^E = \int_V B^T q^E dV \quad (6-171)$$

$$q^E = i^2 R \quad (6-172)$$

where i and R are the current and electrical resistance, respectively.

The controls for the heat transfer option allow input of parameters that govern the convergence solution and accuracy of heat transfer analysis.

Coupled Electrical-Thermal-Mechanical Analysis

Coupled electrical-thermal-mechanical analysis (Joule-mechanical) basically combines electrical-thermal analysis (Joule heating) with thermal-mechanical analysis. Coupled electrical-thermal-mechanical analysis is handled using a staggered solution procedure. Using this approach, the electrical problem is solved first for the nodal voltages. Next, the thermal problem is solved to obtain the nodal temperatures. The mechanical problem is solved last for the nodal displacements.

Coupling between the electrical and thermal problems is mainly because of heat generation due to electrical flow Q^E (Joule heating). The thermal and mechanical problems are coupled through thermal strain loads F^T and heat generation due to inelastic deformation Q^I and friction Q^F . Additional coupling may be introduced in case of temperature-dependant electrical conductivity K^E and mechanical stiffness K^M . Nonlinearities may arise in the thermal problem due to convection, radiation, and temperature-dependant thermal conductivity and specific heat. The mechanical problem may involve geometric and material nonlinearities. Contact is another source of nonlinearity. If contact occurs between deformable bodies or deformable and rigid bodies in the mechanical problem, boundary conditions of the electrical and thermal problems are updated to reflect the new contact conditions.



The matrix equations governing the electrical, thermal and mechanical problems can be expressed as:

$$K^E(T)V = I$$

$$C^T(T)\dot{T} + K^T(T)T = Q + Q^E + Q^I + Q^F$$

$$M\ddot{u} + D\dot{u} + K^M(T,u,t)u = F + F^T$$

where

V	is the nodal voltage vector
T	is the nodal temperature vector
u	is the nodal displacement vector
$K^E(T)$	is the temperature-dependent electrical conductivity matrix
I	is the nodal current vector
$C^T(T)$	is the temperature-dependent heat capacity matrix
$K^T(T)$	is the temperature-dependent thermal conductivity matrix
Q	is the heat flux vector
Q^E	is the heat generation due to electrical flow vector
Q^I	is the heat generation due to inelastic deformation vector
Q^F	is the heat generation due to friction vector
M	is the mass matrix
D	is the damping matrix
$K^M(T,u,t)$	is the temperature, deformation and time-dependent stiffness matrix
F	is the externally applied force vector
F^T	is the force due to thermal strain vector

The **JOULE** and **COUPLE** parameters are needed to initiate the coupled electrical-thermal-mechanical analysis. When defining the mesh through the **CONNECTIVITY** model definition option, specify the elements as stress type. MSC.Marc internally switches to the associated heat transfer element in the electrical and thermal passes.

The **VOLTAGE** model definition option can be used for nodal voltage boundary conditions, and **POINT CURRENT** and/or **DIST CURRENT** model definition options for current boundary conditions. No initial condition is required for the electrical problem, since a steady-state solution is obtained.

In the thermal problem, **INITIAL TEMP**, **FILMS**, **POINT FLUX**, **DIST FLUXES**, and **FIXED TEMPERATURE** model definition options can be used to prescribe the initial conditions and boundary conditions. The user subroutines **FILM** and **FLUX** can be used for complex convective and flux boundary conditions. To enter the unit conversion factor between the electrical and thermal problems, use the **JOULE** model definition option. MSC.Marc uses this conversion factor to compute heat flux generated from the current flow in the structure.

In the mechanical problem, the usual structural model definition options such as **FIXED DISP**, **POINT LOAD**, and **DIST LOADS** are used to prescribe the boundary conditions.

The history definition options (**VOLTAGE CHANGE**, **POINT CURRENT**, **DIST CURRENT**, **TEMP CHANGE**, **POINT FLUX**, **DIST FLUXES**, **DISP CHANGE**, **POINT LOAD**, and **DIST LOADS**) for the incrementation and change of boundary conditions.

JOULE and **CONVERT** model definition options are used to enter the conversion factor for the electrical-thermal and the thermal-mechanical problems.

References

1. T. J. R. Hughes, L. P. France, and M. Becestra, "A New Finite Element Formulation for Computational Fluid Dynamics, V. Circumventing the Babuska-Brezzi Conduction, A S Petrov-Galerkin Formulation of the Stokes Problem Accommodating Equal-Order Interpolations", *Comp. Meth. in Applied Mech. and Eng.*, p. 85-90, Vol. 59, 1986.
2. G. Hauke and T. J. R. Hughes, "A unified approach to compressible and incompressible flow", *Comput. Methods Appl. Mech. Eng.*, p. 389-395, 113, (1994).
3. T. E. Tezduyar, S. Mittal, S. E. Ray, and R. Shih, "Incompressible Flow computations with stabilized bilinear and linear equal order-interpolation velocity-pressure elements", *Comput. Methods. Appl. Mech. Eng.*, p. 221-242, 95, (1992).
4. B. Ramaswaymy, T. C. Jue, "Some Recent Trends and Developments in Finite Element Analysis for Incompressible Thermal Flows", *Int. J. Num. Meth. Eng.*, p. 671-707, 3, (1992).
5. A. N. Brooks and T. J. R. Hughes, "Streamline upwind/Petrov-Galerkin formulations for convection dominated flows with particular emphasis on the incompressible Navier-Stokes equations", *Comp. Meth. Appl. Eng.*, 30, (1982).
6. Codina, R. "Finite element formulation for the numerical solution of the convection-diffusion equation" 1993.
7. Cornfield, G. C., and Johnson, R. H. "Theoretical Predictions of Plastic Flow in Hot Rolling Including the Effect of Various Temperature Distributions." *Journal of Iron and Steel Institute* 211, pp. 567-573, 1973.
8. Hsu, M. B. "Modeling of Coupled Thermo-Electrical Problems by the Finite Element Method." Third International Symposium on Numerical Methods for Engineering, Paris, March, 1983.
9. MARC Reference manual Volume F.
10. Peeters, F. J. H. "Finite Element Analysis of Elasto-Hydrodynamic Lubrication Problems." in *Proceedings of the XIth Int. Finite Element Kongress*, edited by IKOSS GmbH. Baden- Baden, Germany, Nov. 15-16, 1982.

11. Yu, C C. and Heinrich, J. C. “Petro-Galerkin methods for the time-dependent convective transport equation.” *Int. J. Numer. Meth. Engrg.*, Vol. 23 (1986), 883-901.
12. Yu, C C. and Heinrich, J. C. “Petro-Galerkin methods for multidimensional time-dependent convective transport equation.” *Int. J. Numer. Meth. Engrg.*, Vol. 24 (1987), 2201-2215.
13. Zienkiewicz, O. C. *The Finite Element Method in Engineering Science*. Third Ed. London: McGraw-Hill, 1978.
14. Zienkiewicz, O. C., and Godbole, P. N. “A Penalty Function Approach to Problems of Plastic Flow of Metals with Large Surface Deformations.” *Journal of Strain Analysis* 10, 180-183, 1975.
15. Zienkiewicz, O. C., and Godbole, P. N. “Flow of Plastic and viscoPlastic Solids with Special Reference to Extrusion and Forming Processes.” *Int. Num. Methods in Eng.* 8, 1974.
16. Zienkiewicz, O. C., Loehner, R., Morgan, K., and Nakazawa, S. *Finite Elements in Fluid Mechanics – A Decade of Progress*, John Wiley & Sons Limited, 1984.

CHAPTER 7

Material Library

- Linear Elastic Material
- Composite Material
- Gasket
- Nonlinear Hypoelastic Material
- Thermo-Mechanical Shape Memory Model
- Elastomer
- Time-independent Inelastic Behavior
- Time-dependent Inelastic Behavior
- Temperature Effects and Coefficient of Thermal Expansion
- Time-Temperature-Transformation
- Low Tension Material
- Soil Model
- Damage Models
- Nonstructural Materials
- References

This chapter describes the material models available in MSC.Marc. The models range from simple linear elastic materials to complex time- and temperature-dependent materials. This chapter provides basic information on the behavior of various types of engineering materials and specifies the data required by the program for each material. For example, to characterize the behavior of an isotropic linear elastic material at constant temperatures, you need only specify Young's modulus and Poisson's ratio. However, much more data is required to simulate the behavior of material that has either temperature or rate effects. References to more detailed information are cited in this chapter.

The material models included in this chapter are:

- [Linear Elastic Material](#)
- [Composite Material](#)
- [Gasket](#)
- [Nonlinear Hypoelastic Material](#)
- [Thermo-Mechanical Shape Memory Model](#)
- [Elastomer](#)
- [Time-independent Inelastic Behavior](#)
- [Time-dependent Inelastic Behavior](#)
- [Temperature Effects and Coefficient of Thermal Expansion](#)
- [Time-Temperature-Transformation](#)
- [Low Tension Material](#)
- [Soil Model](#)
- [Damage Models](#)
- [Nonstructural Materials](#)

Data for the materials is entered into MSC.Marc either directly through the input file or by user subroutines. Each section of this chapter discusses various options for organizing material data for input. Each section also discusses the constitutive (stress-strain) relation and graphic representation of the models and includes recommendations and cautions concerning the use of the models.

Linear Elastic Material

MSC.Marc is capable of handling problems with either isotropic linear elastic material behavior or anisotropic linear elastic material behavior.

The linear elastic model is the model most commonly used to represent engineering materials. This model, which has a linear relationship between stresses and strains, is represented by Hooke's Law. Figure 7-1 shows that stress is proportional to strain in a uniaxial tension test. The ratio of stress to strain is the familiar definition of modulus of elasticity (Young's modulus) of the material.

$$E \text{ (modulus of elasticity)} = (\text{axial stress})/(\text{axial strain}) \quad (7-1)$$

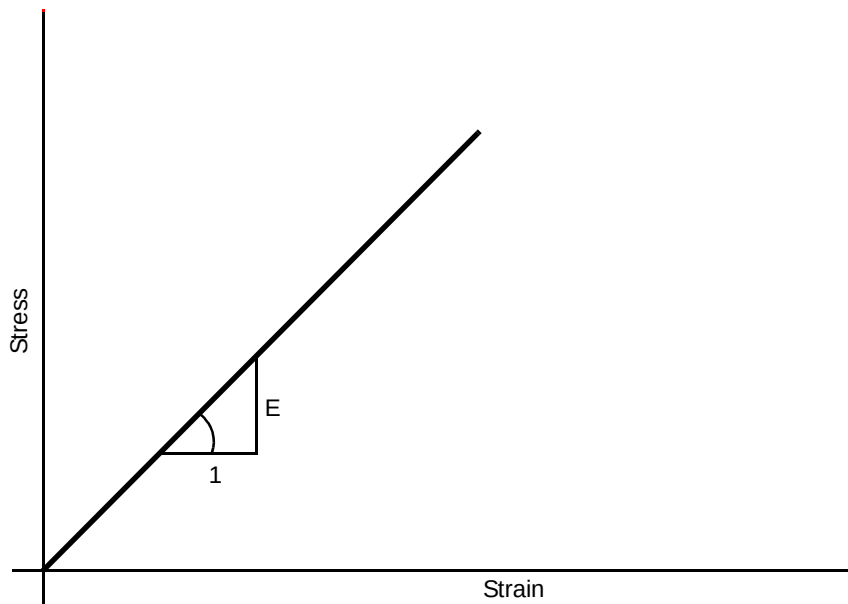


Figure 7-1 Uniaxial Stress-Strain Relation of Linear Elastic Material

Experiments show that axial elongation is always accompanied by lateral contraction of the bar. The ratio for a linear elastic material is:

$$\nu = (\text{lateral contraction})/(\text{axial elongation}) \quad (7-2)$$

This is known as Poisson's ratio. Similarly, the shear modulus (modulus of rigidity) is defined as:

$$G \text{ (shear modulus)} = (\text{shear stress})/(\text{shear strain}) \quad (7-3)$$

It can be shown that for an isotropic material

$$G = E/(2(1 + \nu)) \quad (7-4)$$

The shear modulus G can be easily calculated if the modulus of elasticity E and Poisson's ratio ν are known.

Most linear elastic materials are assumed to be isotropic (their elastic properties are the same in all directions). Anisotropic material exhibits different elastic properties in different directions. The significant directions of the material are labeled as preferred directions, and it is easiest to express the material behavior with respect to these directions.

The stress-strain relationship for an isotropic linear elastic method is expressed as

$$\sigma_{ij} = \lambda \delta_{ij} \epsilon_{kk} + 2G \epsilon_{ij} \quad (7-5)$$

where λ is the Lamé constant and G (the shear modulus) is expressed as

$$\begin{aligned} \lambda &= \nu E / ((1 + \nu)(1 - 2\nu)) \quad \text{and} \\ G &= E / (2(1 + \nu)) \end{aligned} \quad (7-6)$$

The material behavior can be completely defined by the two material constants E and ν .

Use the model definition option **ISOTROPIC** for the input of isotropic linear elastic material constants E (Young's modulus) and ν (Poisson's ratio).

The effects of these parameters on the design can be determined by using the **DESIGN SENSITIVITY** parameter. The optimal value of the elastic properties for linear elastic analysis can be determined using the **DESIGN OPTIMIZATION** parameter.

The stress-strain relationship for an anisotropic linear elastic material can be expressed as

$$\sigma_{ij} = C_{ijkl} \epsilon_{kl} \quad (7-7)$$

The values of C_{ijkl} (the stress-strain relation) and the preferred directions (if necessary) must be defined for an anisotropic material. For example, the orthotropic stress-strain relationship for a plane stress element is

$$C = \frac{1}{(1 - \nu_{12}\nu_{21})} \begin{bmatrix} E_1 & \nu_{21}E_1 & 0 \\ \nu_{12}E_2 & E_2 & 0 \\ 0 & 0 & (1 - \nu_{12}\nu_{21})G \end{bmatrix} \quad (7-8)$$

There are only four independent constants in [Equation 7-8](#).

To input anisotropic stress-strain relations, use the model definition options [ORTHOTROPIC](#) or [ANISOTROPIC](#) and user subroutine [ANELAS](#) or [HOOKLW](#). The [ORTHOTROPIC](#) option allows as many as 9 elastic constants to be defined. The [ANISOTROPIC](#) option allows as many as 21 elastic constants to be defined. If the anisotropic material has a preferred direction, use the model definition option [ORIENTATION](#) or the user subroutine [ORIENT](#) to input a transformation matrix. Refer to [MSC.Marc Volume D: User Subroutines and Special Routines](#) for information on user subroutines.

A Poisson's ratio of 0.5, which would be appropriate for an incompressible material, can be used for the following elements: Herrmann, plane stress, shell, truss, or beam. A Poisson's ratio which is close (but not equal) to 0.5 can be used for constant dilation elements and reduced integration elements in situations which do not include other severe kinematic constraints. Using a Poisson's ratio close to 0.5 for all other elements usually leads to behavior that is too stiff. A Poisson's ratio of 0.5 can also be used with the updated Lagrangian formulation in the multiplicative decomposition framework using the standard displacement elements. In these elements, the treatment for incompressibility is transparent to you.

Composite Material

Composite materials are composed of layers of different materials (or layers of the same anisotropic material) with various layer thicknesses and different orientations. The material in each layer may be either linear or nonlinear. Tightly bonded layers (layered materials) are often stacked in the thickness direction of beam, plate, shell structures, or solids.

Figure 7-2 identifies the locations of integration points through the thickness of beam and shell elements with/without the **COMPOSITE** option.

Note that when the **COMPOSITE** option is used, as shown on the left, the layer points are positioned midway through each layer. When the **COMPOSITE** option is not used, the layer points are equidistantly spaced between the top and bottom surfaces. MSC.Marc forms a stress-strain law by performing numerical integration through the thickness. If the **COMPOSITE** option is used, the trapezoidal method is employed; otherwise, Simpson's rule is used.

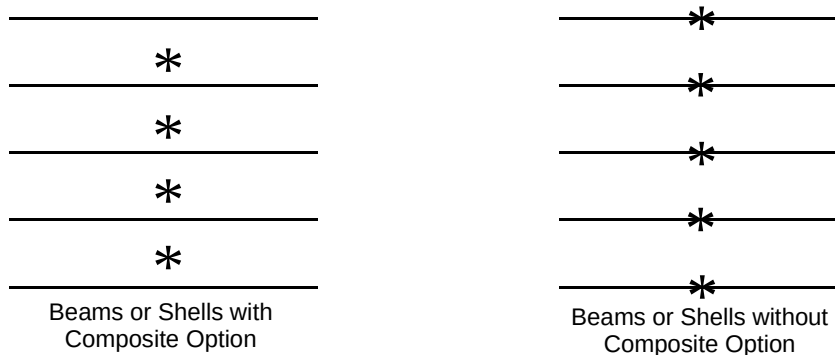


Figure 7-2 Integration Points through the Thickness of Beam and Shell Elements

Figure 7-3 shows the location of integration points through the thickness of composite continuum elements. MSC.Marc forms the element stiffness matrix by performing numerical integration based on the standard isoparametric concept.



Figure 7-3 Integration Points through the Thickness of Continuum Composite Elements

Layered Materials

To model layered materials including plates, shells, beams, and solids with MSC.Marc, use the **COMPOSITE** option. In this option, three quantities are specified on a layer-by-layer basis: material identification number, layer thickness, and ply angle. The entire set of data (a “composite group”) is then associated with a list of elements. For each individual layer, various constitutive laws can be used. The layer thickness can be constant or variable (in the case of variable total thickness elements), and the ply angle can change from one layer to the next. The orientation of the 0° ply angle within each element is defined in the **ORIENTATION** option. The ply thickness and the ply angle can be used as design variables in a design sensitivity analysis. The optimal values can be determined using the design optimization capability for linear elastic analysis.

The material identification number specified in the **COMPOSITE** option, is cross-referenced with the material identification number supplied in the **ISOTROPIC**, **ORTHOTROPIC**, **ANISOTROPIC**, **TEMPERATURE EFFECTS**, **ORTHO TEMP**, **WORK HARD**, and **STRAIN RATE** options. The **ISOTROPIC**, **ORTHOTROPIC**, and **ANISOTROPIC** model definition options allow you to input material constants such as Young’s modulus, Poisson’s ratio, shear modulus, etc. The **TEMPERATURE EFFECTS** and **ORTHO TEMP** options allow for input of temperature dependency of these material constants. Material constants for a typical layer are as follows:

t_i	thickness of the i^{th} layer
Young’s moduli	E_{xx}, E_{yy}, E_{zz}
Poisson’s ratios	$\nu_{xy}, \nu_{yz}, \nu_{zx}$
Shear moduli	G_{xy}, G_{yz}, G_{zx}
ρ	density

α_{xx}, α_{yy}	coefficients of thermal expansion
σ_y	yield stress
Mat	material identifier associated with temperature-dependent properties and workhardening data

User subroutines [ANELAS](#), [HOOKLW](#), [ANEXP](#), and [ANPLAS](#) can be used for the anisotropic behavior of elastic constants, coefficient of thermal expansion, and yield condition.

There are seven given classes of strain-stress relations. The class of a particular element depends on the number of direct (NDI) and shear (NSHEAR) components of stress. [Table 7-1](#) lists the eight classes of elements.

Table 7-1 Classes of Stress-Strain Relations

Class 1	NDI = 1, NSHEAR = 0 Beam Elements 5, 8, 13, 16, 23, 46, 47, 48, 52, 64, 77, 79 and Rebar Elements $\{\epsilon\} = [1/E_{xx}] \{\sigma\}$
Class 2	NDI = 2, NSHEAR = 0 Axisymmetric Shells 15 and 17 $\begin{Bmatrix} \epsilon_{xx} \\ \epsilon_{yy} \end{Bmatrix} = \begin{bmatrix} 1/E_{xx} & -\nu_{yx}/E_{yy} \\ -\nu_{xy}/E_{xx} & 1/E_{yy} \end{bmatrix} \begin{Bmatrix} \sigma_{xx} \\ \sigma_{yy} \end{Bmatrix}$ $\nu_{yx} = \nu_{xy} E_{yy} / E_{xx}$
Class 3	NDI = 1, NSHEAR = 1 Beam Elements 14, 45, 76, 78 $\begin{Bmatrix} \epsilon \\ \gamma \end{Bmatrix} = \begin{bmatrix} 1/E_{xx} & 0 \\ 0 & 1/G_{xy} \end{bmatrix} \begin{Bmatrix} \sigma \\ \tau \end{Bmatrix}$
Class 4	NDI = 2, NSHEAR = 1 Plane Stress, Plates and Thin Shells 49 and 72 $\begin{Bmatrix} \epsilon_{xx} \\ \epsilon_{yy} \\ \gamma_{xy} \end{Bmatrix} = \begin{bmatrix} 1/E_{xx} & -\nu_{yx}/E_{yy} & 0 \\ -\nu_{xy}/E_{xx} & 1/E_{yy} & 0 \\ 0 & 0 & 1/G_{xy} \end{bmatrix} \begin{Bmatrix} \sigma_{xx} \\ \sigma_{yy} \\ \tau_{xy} \end{Bmatrix}$ $\nu_{yx} = \nu_{xy} (E_{yy} / E_{xx})$

Table 7-1 Classes of Stress-Strain Relations (Continued)

Class 5	NDI = 2, NSHEAR = 1 Thick Axisymmetric Shells 1 and 89 $\begin{Bmatrix} \epsilon_{mm} \\ \epsilon_{\theta\theta} \\ \gamma_T \end{Bmatrix} = \begin{bmatrix} 1/E_{mm} & -\nu_{\theta m}/E_{\theta\theta} & 0 \\ -\nu_{m\theta}/E_{mm} & 1/E_{\theta\theta} & 0 \\ 0 & 0 & 1/G_{m\theta} \end{bmatrix} \begin{Bmatrix} \sigma_{mm} \\ \sigma_{\theta\theta} \\ \tau_T \end{Bmatrix}$
Class 6	NDI = 3, NSHEAR = 1 Plane Strain, Axisymmetric with No Twist, Elements 151-154. $\begin{Bmatrix} \epsilon_{xx} \\ \epsilon_{yy} \\ \epsilon_{zz} \\ \gamma_{xy} \end{Bmatrix} = \begin{bmatrix} 1/E_{xx} & -\nu_{yx}/E_{yy} & -\nu_{zx}/E_{zz} & 0 \\ -\nu_{xy}/E_{xx} & 1/E_{yy} & -\nu_{zy}/E_{zz} & 0 \\ -\nu_{xz}/E_{xx} & -\nu_{yz}/E_{yy} & 1/E_{zz} & 0 \\ 0 & 0 & 0 & 1/G_{xy} \end{bmatrix} \begin{Bmatrix} \sigma_{xx} \\ \sigma_{yy} \\ \sigma_{zz} \\ \tau_{xy} \end{Bmatrix}$ $\nu_{yx} = \nu_{xy}E_{yy}/E_{xx} \quad \nu_{zy} = \nu_{yz}E_{zz}/E_{yy} \quad \nu_{xz} = \nu_{zx}E_{xx}/E_{zz}$
Class 7	NDI = 2, NSHEAR = 3 Thick Shell, Elements 22, 75, and 140 $\begin{Bmatrix} \epsilon_{xx} \\ \epsilon_{yy} \\ \gamma_{xy} \\ \gamma_{yz} \\ \gamma_{zx} \end{Bmatrix} = \begin{bmatrix} 1/E_{xx} & -\nu_{yx}/E_{yy} & 0 & 0 & 0 \\ -\nu_{xy}/E_{xx} & 1/E_{yy} & 0 & 0 & 0 \\ 0 & 0 & 1/G_{xy} & 0 & 0 \\ 0 & 0 & 0 & 1/G_{yz} & 0 \\ 0 & 0 & 0 & 0 & 1/G_{zx} \end{bmatrix} \begin{Bmatrix} \sigma_{xx} \\ \sigma_{yy} \\ \tau_{xy} \\ \tau_{yz} \\ \tau_{zx} \end{Bmatrix}$ $\nu_{yx} = \nu_{xy}E_{yy}/E_{xx}$

Table 7-1 Classes of Stress-Strain Relations (Continued)

Class 8	NDI = 3, NSHEAR = 3 Three-Dimensional Brick Elements, Elements 149, 150 $ \begin{Bmatrix} \epsilon_{xx} \\ \epsilon_{yy} \\ \epsilon_{zz} \\ \gamma_{xy} \\ \gamma_{yz} \\ \gamma_{zx} \end{Bmatrix} = \begin{bmatrix} 1/E_{xx} & -\nu_{yx}/E_{yy} & -\nu_{zx}/E_{zz} & 0 & 0 & 0 \\ -\nu_{xy}/E_{xx} & 1/E_{yy} & -\nu_{zy}/E_{zz} & 0 & 0 & 0 \\ -\nu_{xz}/E_{xx} & -\nu_{yz}/E_{yy} & 1/E_{zz} & 0 & 0 & 0 \\ 0 & 0 & 0 & 1/G_{xy} & 0 & 0 \\ 0 & 0 & 0 & 0 & 1/G_{yz} & 0 \\ 0 & 0 & 0 & 0 & 0 & 1/G_{zx} \end{bmatrix} \begin{Bmatrix} \sigma_{xx} \\ \sigma_{yy} \\ \sigma_{zz} \\ \tau_{xy} \\ \tau_{yz} \\ \tau_{zx} \end{Bmatrix} $

Material Preferred Direction

Every element type in MSC.Marc has a default orientation (that is, a default coordinate system) within which element stress-strain calculations take place. This system is also assumed to be the coordinate system of material symmetry. This is especially important for non-isotropic materials (orthotropic, anisotropic, or composite materials). With the [ORIENTATION](#) option, you specify the orientation of the material axes of symmetry (relationship between the element coordinate system and the global coordinate system, or the 0° ply angle line, if composite) in one of four different ways:

1. as a specific angle offset from an element edge,
2. as a specific angle offset from the line created by two intersecting planes,
3. as a particular coordinate system specified by user-supplied unit vectors, or
4. as specified by user subroutine [ORIENT](#). This is accomplished by the specification of an orientation type, an orientation angle, or one or two user-defined vectors.

For the first option (EDGE I-J orientation type), the intersecting plane is defined by the surface normal vector and a vector parallel to the vector pointing from element node I to element node J. The intersection of this plane with the surface tangent plane defines the 0° orientation axis. (See [Figure 7-4](#).) The orientation angle is measured in the tangent plane positive about the surface normal.

For the second option (global plane orientation type), the intersecting plane is the chosen global coordinate plane. The intersection of this plane with the surface tangent plane defines the 0° orientation axis. (See [Figure 7-5](#).)

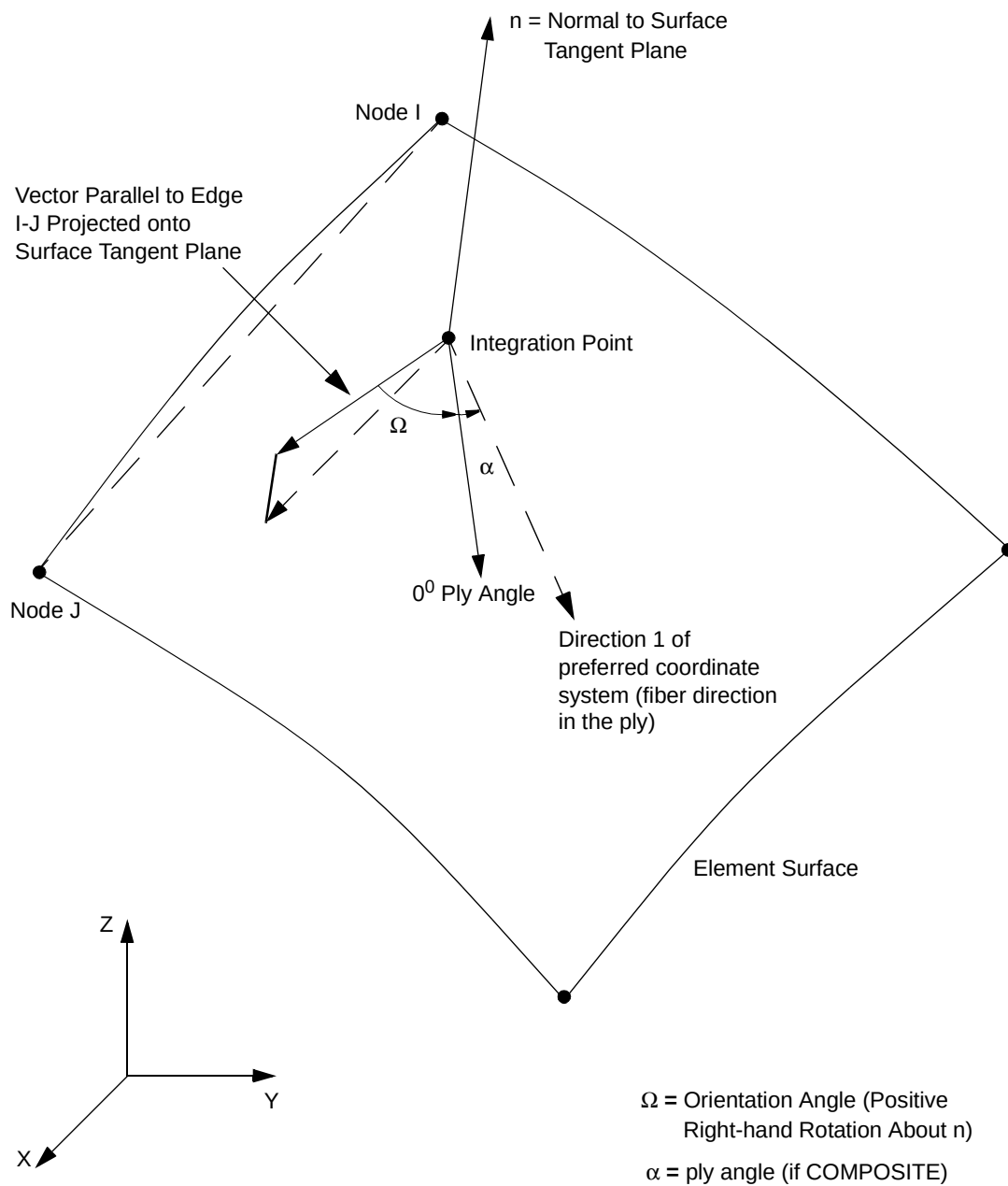


Figure 7-4 Edge I-J Orientation Type

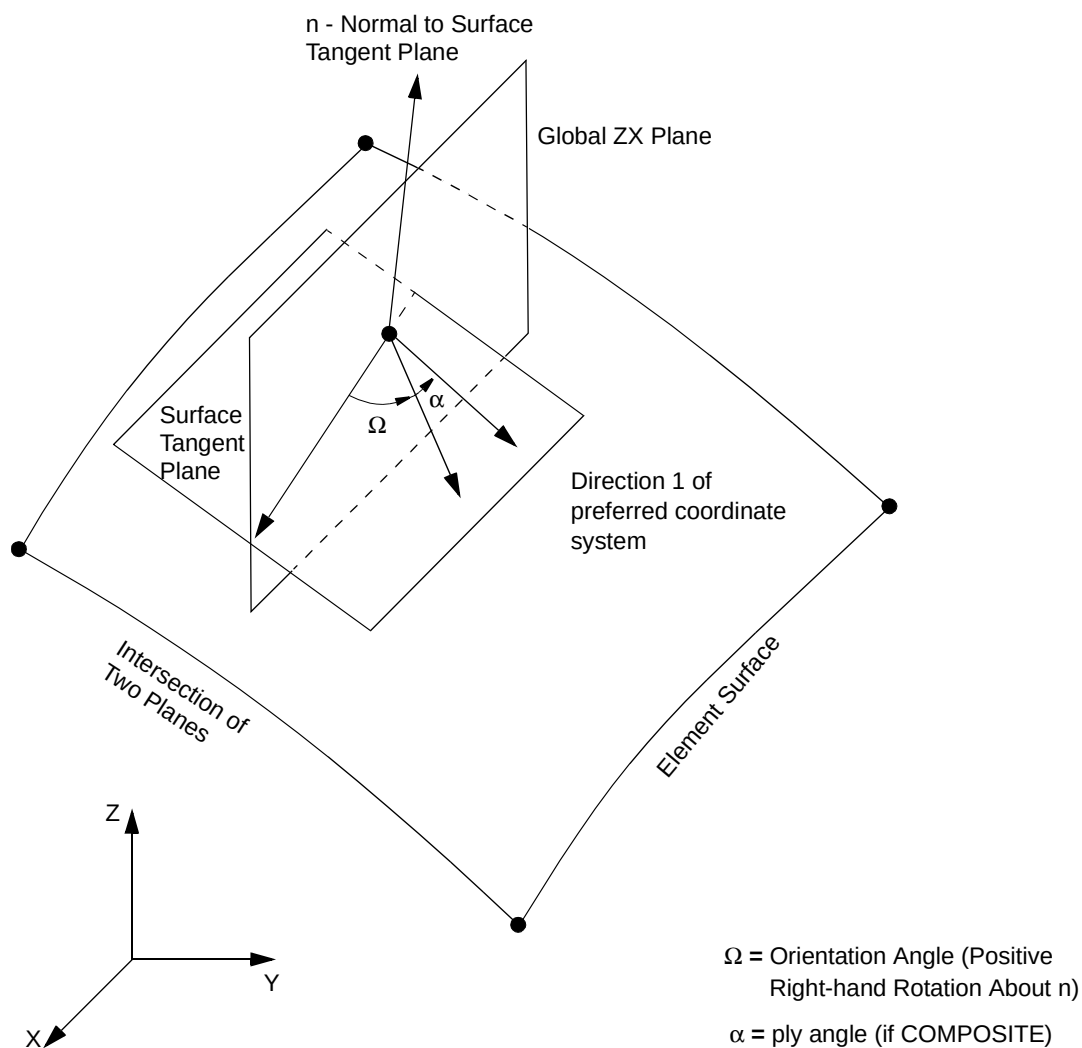


Figure 7-5 Global ZX Plane Orientation Type

The third option (user-defined plane orientation type) makes use of one or two user-defined vectors to define the intersecting plane. Using a single vector, the intersecting plane is that plane which contains the user vector and the chosen coordinate axis. Using two user vectors, the intersecting plane is that plane which contains both of them. (See [Figure 7-6](#).)

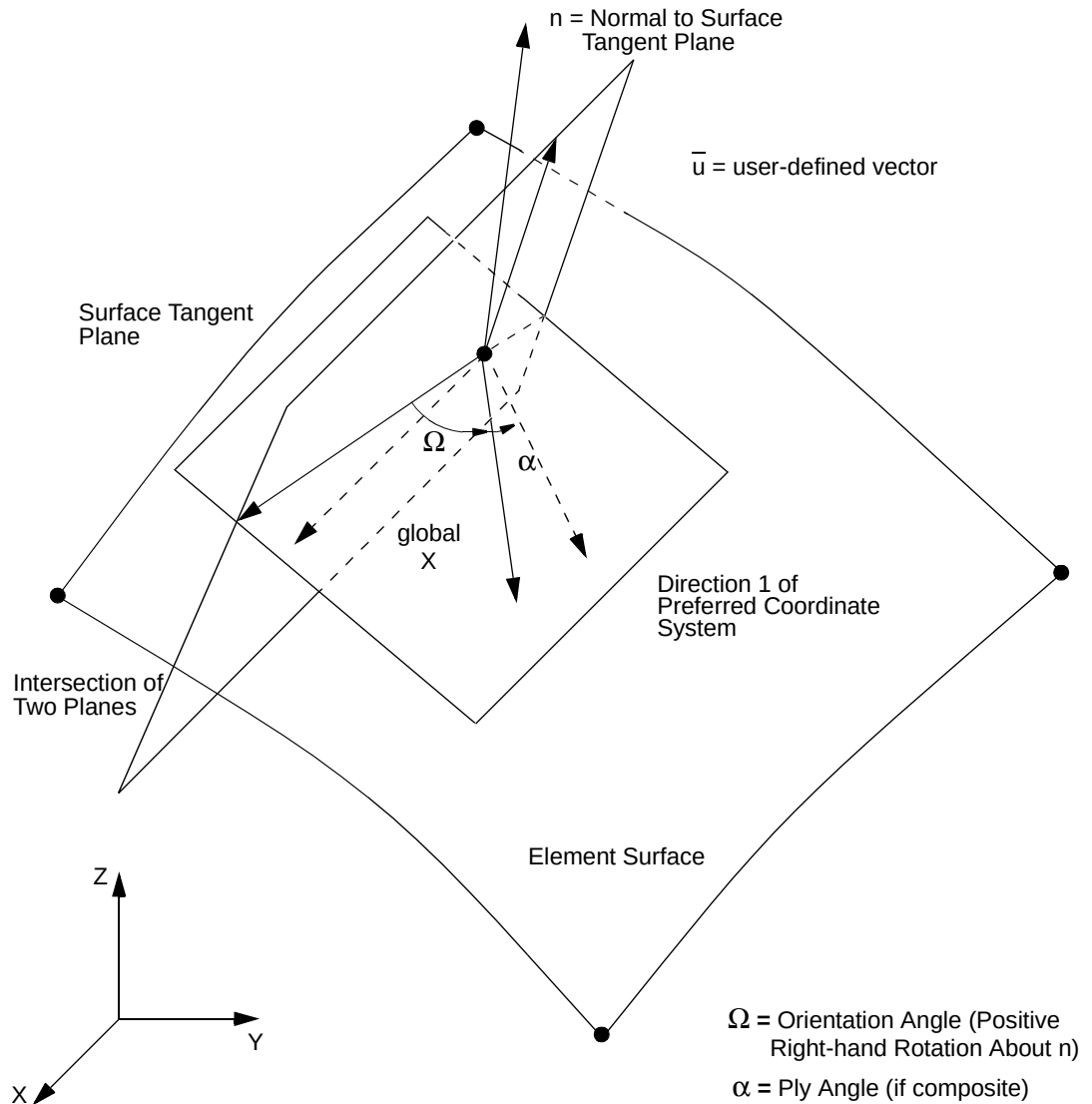


Figure 7-6 User Defined XU Plane Orientation Type

Orientation type 3-D ANISO also makes use of two user-defined vectors, but in this case, the first vector defines the first (1) principal direction and the second vector defines the second (2) principal direction. (See [Figure 7-7](#).)

Finally, in the fourth option, the user subroutine [ORIENT](#) must be used for the definition of the orientation of the material axis of symmetry.

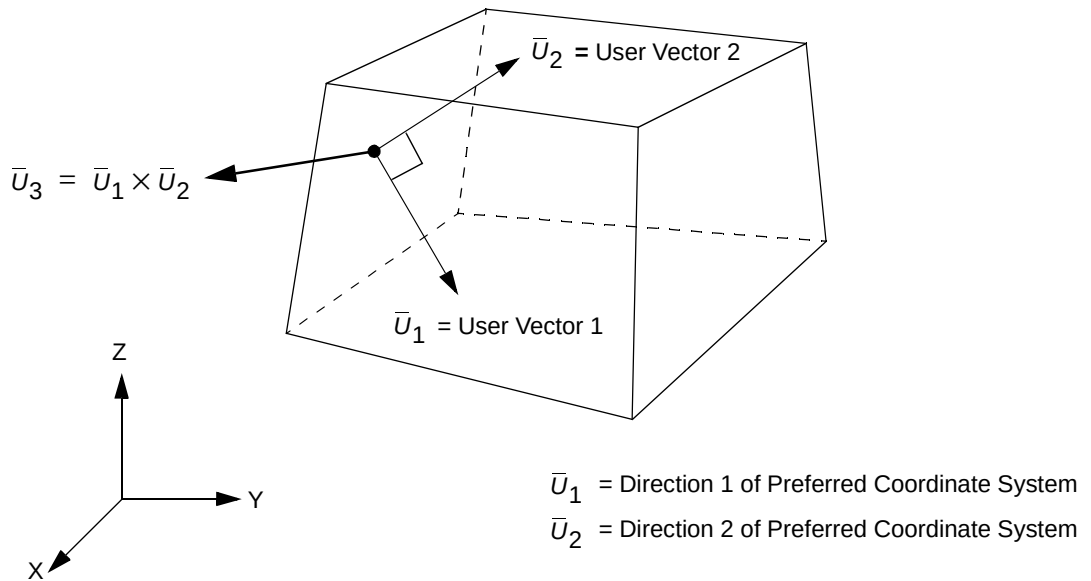


Figure 7-7 3-D ANISO Orientation Type

Material Dependent Failure Criteria

Calculations of user specified failure criteria on a layer by layer basis are available in MSC.Marc. They are maximum stress (MX STRESS), maximum strain (MX STRAIN), TSAI-WU, HOFFMAN, HILL or user subroutine [UFAIL](#). During each analysis, up to three failure criteria can be selected; failure indices are calculated and printed for every integration point. The model definition option [FAIL DATA](#) is used for the input of failure criteria data.

A simple description of these failure criteria is given below:

1. Maximum Stress Criterion

At each integration point, MSC.Marc calculates six quantities:

$$1. \left\{ \begin{array}{ll} \left(\frac{\sigma_1}{X_t} \right) / F & \text{if } \sigma_1 > 0 \\ \left(\frac{-\sigma_1}{X_c} \right) / F & \text{if } \sigma_1 < 0 \end{array} \right. \quad (7-9)$$

$$2. \left\{ \begin{array}{ll} \left(\frac{\sigma_2}{Y_t} \right) / F & \text{if } \sigma_2 > 0 \\ \left(-\frac{\sigma_2}{Y_c} \right) / F & \text{if } \sigma_2 < 0 \end{array} \right. \quad (7-10)$$

$$3. \left\{ \begin{array}{ll} \left(\frac{\sigma_3}{Z_t} \right) / F & \text{if } \sigma_3 > 0 \\ \left(-\frac{\sigma_3}{Z_c} \right) / F & \text{if } \sigma_3 < 0 \end{array} \right. \quad (7-11)$$

$$4. \left(\left| \frac{\sigma_{12}}{S_{12}} \right| \right) / F \quad (7-12)$$

$$5. \left(\left| \frac{\sigma_{23}}{S_{23}} \right| \right) / F \quad (7-13)$$

$$6. \left(\left| \frac{\sigma_{31}}{S_{31}} \right| \right) / F \quad (7-14)$$

where

F is the failure index ($F = 1.0$).

X_t, X_c are the maximum allowable stresses in the 1-direction in tension and compression.

Y_t, Y_c are maximum allowable stresses in the 2-direction in tension and compression.

Z_t, Z_c are maximum allowed stresses in the 3-direction in tension and compression.

S_{12} maximum allowable in-plane shear stress.

S_{23} maximum allowable 23 shear stress.

S_{31} maximum allowable 31 shear stress.

2. Maximum Strain Failure Criterion

At each integration point, calculates six quantities:

$$1. \left\{ \begin{array}{ll} \left(\frac{\epsilon_1}{e_{1t}} \right) / F & \text{if } \epsilon_1 > 0 \\ \left(-\frac{\epsilon_1}{e_{1c}} \right) / F & \text{if } \epsilon_1 < 0 \end{array} \right. \quad (7-15)$$

$$2. \left\{ \begin{array}{ll} \left(\frac{\epsilon_2}{e_{2t}} \right) / F & \text{if } \epsilon_2 > 0 \\ \left(-\frac{\epsilon_2}{e_{2c}} \right) / F & \text{if } \epsilon_2 < 0 \end{array} \right. \quad (7-16)$$

$$3. \left\{ \begin{array}{ll} \left(\frac{\epsilon_3}{e_{3t}} \right) / F & \text{if } \epsilon_3 > 0 \\ \left(-\frac{\epsilon_3}{e_{3c}} \right) / F & \text{if } \epsilon_3 < 0 \end{array} \right. \quad (7-17)$$

$$4. \left(\frac{\gamma_{12}}{g_{12}} \right) / F \quad (7-18)$$

$$5. \left(\frac{\gamma_{23}}{g_{23}} \right) / F \quad (7-19)$$

$$6. \left(\frac{\gamma_{31}}{g_{31}} \right) / F \quad (7-20)$$

where

F is the failure index ($F=1.0$).

e_{1t}, e_{1c} are the maximum allowable strains in the 1 direction in tension and compression.

e_{2t}, e_{2c} are the maximum allowable strains in the 2 direction in tension and compression.

e_{3t}, e_{3c} are the maximum allowable strains in the 3 direction in tension and compression.

g_{12} is the maximum allowable shear strain in the 12 plane.

g_{23} is the maximum allowable shear strain in the 23 plane.

g_{31} is the maximum allowable shear strain in the 31 plane.

3. Hill Failure Criterion

Assumptions:

- a. Orthotropic materials only
- b. Incompressibility during plastic deformation
- c. Tensile and compressive behavior are identical

At each integration point, MSC.Marc calculates:

$$\left[\frac{\sigma_1^2}{X^2} + \frac{\sigma_2^2}{Y^2} + \frac{\sigma_3^2}{Z^2} - \left(\frac{1}{X^2} + \frac{1}{Y^2} - \frac{1}{Z^2} \right) \sigma_1 \sigma_2 - \left(\frac{1}{X^2} + \frac{1}{Z^2} - \frac{1}{Y^2} \right) \sigma_1 \sigma_3 - \left(\frac{1}{Y^2} + \frac{1}{Z^2} - \frac{1}{X^2} \right) \sigma_2 \sigma_3 + \frac{\sigma_{12}^2}{S_{12}^2} + \frac{\sigma_{13}^2}{S_{13}^2} + \frac{\sigma_{23}^2}{S_{23}^2} \right] / F \quad (7-21)$$

For plane stress condition, it becomes

$$\left(\frac{\sigma_1^2}{X^2} - \frac{\sigma_1 \sigma_2}{X^2} + \frac{\sigma_2^2}{Y^2} + \frac{\sigma_{12}^2}{S_{12}^2} \right) / F \quad (7-22)$$

where

X is the maximum allowable stress in the 1 direction

Y is the maximum allowable stress in the 2 direction

Z is the maximum allowable stress in the 3 direction

$S_{12}, S_{23}, S_{31}, F$ are as before

4. Hoffman Failure Criterion

Note: Hoffman criterion is essentially Hill criterion modified to allow unequal maximum allowable stresses in tension and compression.

At each integration point, MSC.Marc calculates:

$$[C_1(\sigma_2 - \sigma_3)^2 + C_2(\sigma_3 - \sigma_1)^2 + C_3(\sigma_1 - \sigma_2)^2 + C_4\sigma_1 + C_5\sigma_2 + C_6\sigma_3 + C_7\sigma_{23}^2 + C_8\sigma_{13}^2 + C_9\sigma_{12}^2]/F \quad (7-23)$$

with

$$C_1 = \frac{1}{2} \left(\frac{1}{Z_t Z_c} + \frac{1}{Y_t Y_c} - \frac{1}{X_t X_c} \right)$$

$$C_2 = \frac{1}{2} \left(\frac{1}{X_t X_c} + \frac{1}{Z_t Z_c} - \frac{1}{Y_t Y_c} \right)$$

$$C_3 = \frac{1}{2} \left(\frac{1}{X_t X_c} + \frac{1}{Y_t Y_c} - \frac{1}{Z_t Z_c} \right)$$

$$C_4 = \frac{1}{X_t} - \frac{1}{X_c}$$

$$C_5 = \frac{1}{Y_t} - \frac{1}{Y_c}$$

$$C_6 = \frac{1}{Z_t} - \frac{1}{Z_c}$$

$$C_7 = \frac{1}{S_{23}^2}$$

$$C_8 = \frac{1}{S_{13}^2}$$

$$C_9 = \frac{1}{S_{12}^2}$$

For plane stress condition, it becomes

$$\left\{ \left(\frac{1}{X_t} - \frac{1}{X_c} \right) \sigma_1 + \left(\frac{1}{Y_t} - \frac{1}{Y_c} \right) \sigma_2 + \frac{\sigma_1^2}{X_t X_c} + \frac{\sigma_2^2}{Y_t Y_c} + \frac{\sigma_{12}^2}{S_{12}^2} - \frac{\sigma_1 \sigma_2}{X_t X_c} \right\} / F \quad (7-24)$$

where: $X_t, X_c, Y_t, Y_c, Z_t, Z_c, S_{12}, S_{23}, S_{31}, F$ are as before.

Note: For small ratios of, for example, $\frac{\sigma_1}{X_t}$, the Hoffman criteria can become *negative* due to the presence of the linear terms.

5. Tsai-Wu Failure Criterion

Tsai-Wu is a tensor polynomial failure criterion. At each integration point, MSC.Marc calculates:

$$\begin{aligned} & \left[\left(\frac{1}{X_t} - \frac{1}{X_c} \right) \sigma_1 + \left(\frac{1}{Y_t} - \frac{1}{Y_c} \right) \sigma_2 + \left(\frac{1}{Z_t} - \frac{1}{Z_c} \right) \sigma_3 + \frac{\sigma_1^2}{X_t X_c} + \frac{\sigma_2^2}{Y_t Y_c} + \frac{\sigma_3^2}{Z_t Z_c} \right. \\ & \quad \left. + \frac{\tau_{12}^2}{S_{12}^2} + \frac{\tau_{23}^2}{S_{23}^2} + \frac{\tau_{13}^2}{S_{13}^2} + 2F_{12}\sigma_1\sigma_2 + 2F_{23}\sigma_2\sigma_3 + 2F_{13}\sigma_1\sigma_3 \right] / F \end{aligned} \quad (7-25)$$

where $X_t, X_c, Y_t, Y_c, Z_t, Z_c, S_{12}, S_{23}, S_{31}, F$ are as before.

F_{12} Interactive strength constant for the 12 plane

F_{23} Interactive strength constant for the 23 plane

F_{13} Interactive strength constant for the 31 plane

For plane stress condition, it becomes

$$\left\{ \left(\frac{1}{X_t} - \frac{1}{X_c} \right) \sigma_1 + \left(\frac{1}{Y_t} - \frac{1}{Y_c} \right) \sigma_2 + \frac{\sigma_1^2}{X_t X_c} + \frac{\sigma_2^2}{Y_t Y_c} + \frac{\sigma_{12}^2}{S_{12}^2} + 2F_{12}\sigma_1\sigma_2 \right\} / F \quad (7-26)$$

Note: In order for the Tsai-Wu failure surface to be closed,

$$F_{12}^2 < \frac{1}{X_t X_c} \cdot \frac{1}{Y_t Y_c} \quad F_{23}^2 < \frac{1}{Y_t Y_c} \cdot \frac{1}{Z_t Z_c} \quad F_{31}^2 < \frac{1}{X_t X_c} \cdot \frac{1}{Z_t Z_c}$$

See Wu, R.Y. and Stachurski, 2, "Evaluation of the Normal Stress Interaction Parameter in the Tensor Polynomial Strength Theory for Anisotropic Materials", *Journal of Composite Materials*, Vol. 18, Sept. 1984, pp. 456-463.

6. User-defined Failure Criteria

Using the user subroutine **UFAIL**, you can evaluate your own failure criterion as a function of stresses and strains at each integration point.

Interlaminar Shear for Thick Shell and Beam Elements

Another addition made for composite analysis is the calculation of interlaminar shears (a parabolic distribution through the thickness direction) for thick shells, elements 22, 75, and 140 and for thick beam element 45. These interlaminar shears are printed in the local coordinate system above and below each layer selected for printing by **PRINT CHOICE** or **PRINT ELEMENT**. These values are also available for postprocessing. The **TSHEAR** parameter must be used for activating the parabolic shear distribution calculations.

In MSC.Marc, the distribution of transverse shear strains through the thickness for thick shell and beam elements was assumed to be constant. From basic strength of materials and the equilibrium of a beam cross section, it is known that the actual distribution is more parabolic in nature. As an additional option, the formulations for elements 22, 75, 140, and 45 have been modified to include a parabolic distribution of transverse shear strain. The formulation is exact for beam element 45, but is approximate for the thick shell elements 22, 75, and 140. Nevertheless, the approximation is expected to give improved results from the previous constant shear distribution. Furthermore, interlaminar shear stresses for composite beams and shells can now be easily calculated.

With the assumption that the stresses in the V^1 and V^2 direction are uncoupled, the equilibrium condition through the thickness is given by

$$\frac{\partial \tau(z)}{\partial z} + \frac{\partial \sigma(z)}{\partial x} = 0 \quad (7-27)$$

where $\sigma(z)$ is the layer axial stress; $\tau(t)$ is the layer shear stress. From beam theory, we have

$$V + \frac{\partial M}{\partial x} = 0 \quad (7-28)$$

where M is the section bending moment and V is the shear force. Assuming that

$$\sigma(z) = f(z)M \quad (7-29)$$

by taking the derivative of Equation 7-29 with respect to x , substituting the result into Equation 7-27, using Equation 7-28 and integrating, we obtain

$$\tau(z) = \int_z f(z) dz \cdot V \quad (7-30)$$

The function $f(z)$ is given from beam theory as

$$f(z) = \frac{E_0(z)}{\bar{EI}}(\bar{z} - z) \quad (7-31)$$

where $E_0(z)$ is the layer initial Young's modulus, $\bar{z}$ is the location of the neutral axis and $\bar{EI}$ is the section bending moment of inertia. Equation 7-31 and Equation 7-29 express the usual bending relation

$$\sigma(z) = -\frac{Mz}{I} \quad (7-32)$$

except that these two equations are written so that the $z = 0$ axis is not necessarily the neutral axis of bending. With respect to this axis, membrane and bending action is, in general, coupled. Note that

$$\bar{z} = \frac{\int z E(z) dz}{\int_z E(z) dz} \quad (7-33)$$

and stress $\tau(z) = 0$ at the top and bottom surface of the shell.

Interlaminar Stresses for Continuum Composite Elements

In MSC.Marc, the interlaminar shear and normal stresses are calculated by averaging the stresses in the stacked two layers. The stresses are transformed into a component tangent to the interface and a component normal to the interface. The two components, considered as shear stress and normal stress, respectively, are printed out in the output file.

By using POST code 501 or 511 (see *MSC.Marc Volume C: Program Input*) representing interlaminar normal and shear stresses respectively, the interlaminar normal or shear stress can be written into a post file in the form of a stress tensor

defined in the global coordinate directions. MSC.Marc Mentat can be used to plot the principal directions of the stress tensor which show the magnitude and the direction of the stress, and the changes based on deformation.

Progressive Composite Failure

A model has been put into MSC.Marc to allow the progressive failure of certain types of composite materials. The aspects of this model are defined below:

1. Failure occurs when any one of the failure criteria is satisfied.
2. The behavior up to the failure point is linear elastic.
3. Upon failure, the material moduli for orthotropic materials at the integration points are changed such that all of the moduli have the lowest moduli entered.
4. Upon failure, for isotropic materials, the failed moduli are taken as 10% of the original moduli.
5. If there is only one modulus, such as in a beam or truss problem, the failed modulus is taken as 10% of the original one.
6. There is no healing of the material.

This is flagged through the [FAIL DATA](#) model definition option.

Gasket

Engine gaskets are used to seal the metal parts of the engine to prevent steam or gas from escaping. They are complex (often multi-layer) components, usually rather thin and typically made of several different materials of varying thickness. The gaskets are carefully designed to have a specific behavior in the thickness direction. This is to ensure that the joints remain sealed when the metal parts are loaded by thermal or mechanical loads. The through-thickness behavior, usually expressed as a relation between the pressure on the gasket and the closure distance of the gasket, is highly nonlinear, often involves large plastic deformations, and is difficult to capture with a standard material model. The alternative of modeling the gasket in detail by taking every individual material into account in the finite element model of the engine is not feasible. It requires a lot of elements which makes the model unacceptably large. Also, determining the material properties of the individual materials might be cumbersome.

The gasket material model addresses these problems by allowing gaskets to be modeled with only one element through the thickness, while the experimentally or analytically determined complex pressure-closure relationship in that direction can be used directly as input for the material model. The material must be used together with the first-order solid composite element types [149](#) (three-dimensional solid element), [151](#) (two-dimensional plane strain element) or [152](#) (two-dimensional axisymmetric element). In that case, these elements consists of one layer and have only one integration point in the thickness direction of the element.

Gaskets can be used in mechanical, thermal or thermo-mechanically coupled analyses. The usual staggered scheme of a heat transfer pass followed by a structural pass is used for coupled analyses. For the heat transfer part, the elements used to model the gaskets are type [175](#) (three-dimensional first-order solid element), type [177](#) (two-dimensional first-order planar element), or type [178](#) (two-dimensional first-order axisymmetric element).

Constitutive Model

The behavior in the thickness direction, the transverse shear behavior, and the membrane behavior are fully uncoupled in the gasket material model. In subsequent sections, these three deformation modes are discussed.

Local Coordinate System

The material model is most conveniently described in terms of a local coordinate system in the integration points of the element (see [Figure 7-8](#)). For three-dimensional elements, the first and second directions of the coordinate system are tangential to the

midsurface of the element at the integration point. The third direction is the thickness direction of the gasket and is perpendicular to the midsurface. For two-dimensional elements, the first direction of the coordinate system is the direction of the midsurface at the integration point, the second direction is the thickness direction of the gasket and is perpendicular to the midsurface, and the third direction coincides with the global 3-direction.

In a total Lagrange formulation, the orientation of the local coordinate system is determined in the undeformed configuration and is fixed. In an updated Lagrange formulation, the orientation is determined in the current configuration and is updated during the analysis.

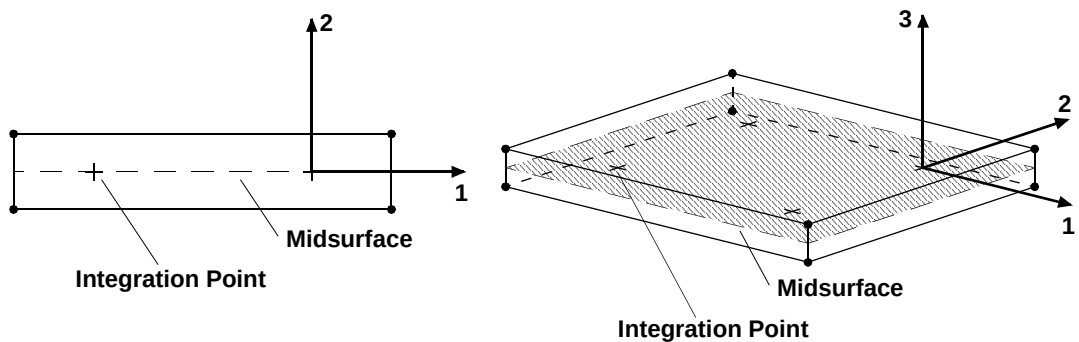


Figure 7-8 The Location of the Integration Points and the Local Coordinate Systems in Two- and Three-dimensional Gasket Elements

Thickness Direction - Compression

In the thickness direction, the material exhibits the typical gasket behavior in compression, as depicted in [Figure 7-9](#). After an initial nonlinear elastic response (section AB), the gasket starts to yield if the pressure p on the gasket exceeds the initial yield pressure p_{y0} . Upon further loading, plastic deformation increases, accompanied by (possibly nonlinear) hardening, until the gasket is fully compressed (section BD). Unloading occurs in this stage along nonlinear elastic paths (section FG, for example). When the gasket is fully compressed, loading and unloading occurs along a new nonlinear elastic path (section CDE), while retaining the permanent deformation built up during compression. No additional plastic deformation is developed once the gasket is fully compressed.

The loading and unloading paths of the gasket are usually established experimentally by compressing the gasket, unloading it again, and repeating this cycle a number of times for increasing pressures. The resulting pressure-closure data can be used as

input for the material model. If the pressure also varies with temperature, then pressure-closure data at different temperatures can be provided. The user must supply the loading path and may specify up to ten unloading paths. In addition, the initial yield pressure p_{y0} must be given. The initial yield pressure can also be varied with temperature and spatial coordinates. The loading path should consist of both the elastic part of the loading path and the hardening part, if present. If no unloading paths are supplied or if the yield pressure is not reached by the loading path, the gasket is assumed to be elastic. In that case, loading and unloading occurs along the loading path.

The loading and unloading paths must be defined using the [TABLE](#) model definition option (see *MSC.Marc Volume C: Program Input*) and must relate the pressure on the gasket to the gasket closure. Optionally, at each closure value, the loading and unloading paths can also be defined as functions of temperature and spatial coordinates using multi-variate [TABLES](#). The unloading paths specify the elastic unloading of the gasket at different amounts of plastic deformation; the closure at zero pressure is taken as the plastic closure on the unloading path. If unloading occurs at an amount of plastic deformation for which no path has been specified, the unloading path is constructed automatically by linear interpolation between the two nearest user supplied paths. The unloading path, supplied by the user, with the largest amount of plastic deformation is taken as the elastic path at full compression of the gasket.

For example, in [Figure 7-9](#), the loading path is given by the sections AB (elastic part) and BD (hardening part) and the initial yield pressure is the pressure at point B. The (single) unloading path is curve CDE. The latter is also the elastic path at full compression of the gasket. The amount of plastic closure on the unloading path is c_{p1} . The dashed curve FG is the unloading path at a certain plastic closure c_p that is constructed by interpolation from the elastic part of the loading path (section AB) and the unloading path CD.

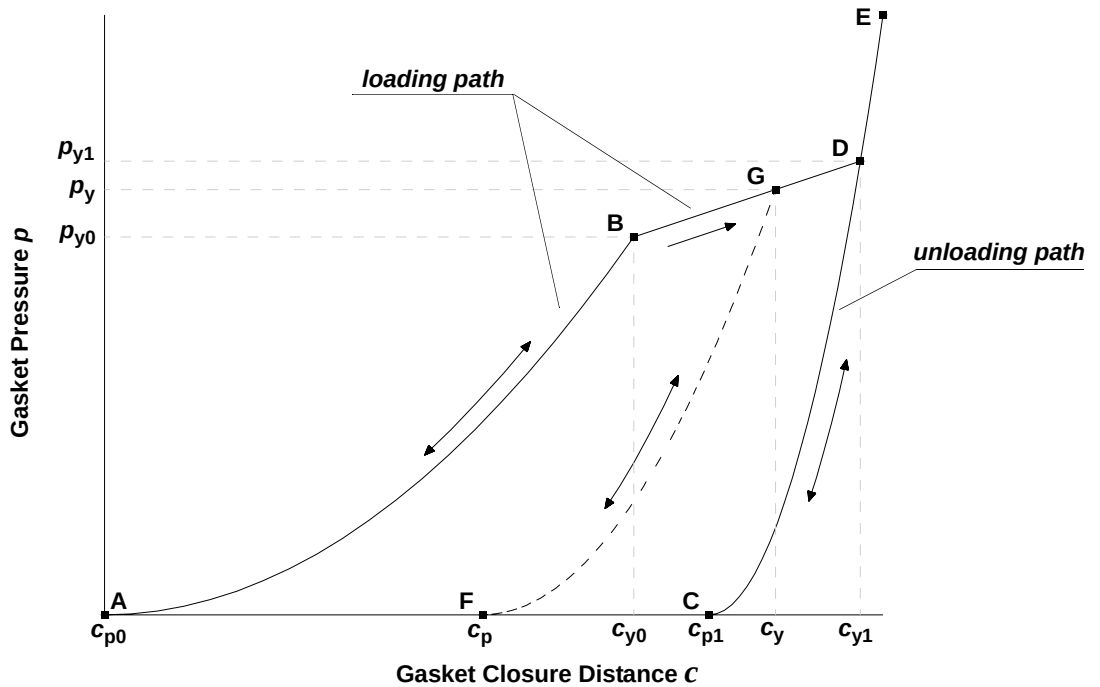


Figure 7-9 Pressure-closure Relation of a Gasket

The compressive behavior in the thickness direction is implemented by decomposing the incremental gasket closure into an elastic and a plastic part:

$$\Delta c = \Delta c^e + \Delta c^p \quad (7-34)$$

Of these two parts, only the elastic part contributes to the pressure. The constitutive equation is given by the following equation:

$$\Delta p = D_c \Delta c^e = D_c (\Delta c - \Delta c^p). \quad (7-35)$$

Here, D_c is the consistent tangent to the pressure-closure curve.

Plastic deformation develops when the pressure p equals the current yield pressure p_y . The latter is a function of the amount of plastic deformation developed so far and is given by the hardening part of the loading path (section BD in [Figure 7-9](#)).

Initial Gap

The thickness of a gasket can vary considerably throughout the sealing region. Since the gasket is modeled with only one element through the thickness, this can lead to meshing difficulties at the boundaries between thick regions and thin regions. The initial gap parameter can be used to solve this. The parameter basically shifts the loading and unloading curves in the positive closure direction. As long as the closure distance of the gasket elements is smaller than the initial gap, no pressure is built up in the gasket. The sealing region can thus be modeled as a flat sheet of uniform thickness and the initial gap parameter can be set for those regions where the gasket is actually thinner than the elements of the finite element mesh used to model it. The initial gap can optionally be varied as a function of spatial coordinates by using a table.

Thickness Direction - Tension

The tensile behavior of the gasket in the thickness direction is linear elastic and is governed by a tensile modulus D_t . The latter is defined as a pressure per unit closure distance (that is, length). D_t can optionally be varied as a function of temperature and spatial coordinates by using a multi-variate table.

Transverse Shear and Membrane Behavior

The transverse shear is defined in the 2-3 and 3-1 planes of the local coordinate system (for three-dimensional elements) or the 1-2 plane (for two-dimensional elements). It is linear elastic and characterized by a transverse shear modulus G_t . G_t can optionally be varied as a function of temperature and spatial coordinates by using a multi-variate table.

The membrane behavior is defined in the local 1-2 plane (for three-dimensional elements) or the local 3-1 plane (for two-dimensional elements) and is linear elastic and isotropic. Young's modulus E_m and Poisson's ratio ν_m that govern the membrane behavior are taken from an existing material that must be defined using the **ISOTROPIC** model definition option. Multiple gasket material can refer to the same isotropic material for their membrane properties (see also the **GASKET** model definition option in *MSC.Marc Volume C: Program Input*).

Thermal Expansion

The thermal expansion of the gasket material is isotropic and the thermal expansion coefficient is taken from the isotropic material that also describes the membrane behavior.

Constitutive Equations

As mentioned above, the behavior in the thickness direction of the gasket is formulated as a relation between the pressure p on the gasket and the gasket closure distance c . In order to formulate the constitutive equations of the gasket material, this relation (Equation 7-35) must first be written in terms of stresses and strains. This depends heavily on the stress and strain tensor employed in the analysis. For small strain analyses, for example, the engineering stress and strain are used. In that case, the incremental gasket closure and the incremental pressure are related to the incremental strain and the incremental stress by

$$\Delta c = -h\Delta\epsilon \quad \text{and} \quad \Delta p = -\Delta\sigma \quad (7-36)$$

in which h is the thickness of the gasket.

The resulting constitutive equation for three-dimensional elements, expressed in the local coordinate system of the integration points, now reads

$$\begin{bmatrix} \Delta\sigma_{11} \\ \Delta\sigma_{22} \\ \Delta\sigma_{33} \\ \Delta\sigma_{12} \\ \Delta\sigma_{23} \\ \Delta\sigma_{31} \end{bmatrix} = \begin{bmatrix} \frac{E_m}{1-\nu_m^2} & \frac{\nu_m E_m}{1-\nu_m^2} & 0 & 0 & 0 & 0 \\ \frac{\nu_m E_m}{1-\nu_m^2} & \frac{E_m}{1-\nu_m^2} & 0 & 0 & 0 & 0 \\ 0 & 0 & C & 0 & 0 & 0 \\ 0 & 0 & 0 & \frac{E_m}{2(1+\nu_m)} & 0 & 0 \\ 0 & 0 & 0 & 0 & G_t & 0 \\ 0 & 0 & 0 & 0 & 0 & G_t \end{bmatrix} \begin{bmatrix} \Delta\epsilon_{11} \\ \Delta\epsilon_{22} \\ \Delta\epsilon_{33} - \Delta\epsilon_{33}^p \\ \Delta\gamma_{12} \\ \Delta\gamma_{23} \\ \Delta\gamma_{31} \end{bmatrix} \quad (7-37)$$

in which $C = hD_c$. For two-dimensional elements, the equation is given by

$$\begin{bmatrix} \Delta\sigma_{11} \\ \Delta\sigma_{22} \\ \Delta\sigma_{33} \\ \Delta\sigma_{12} \end{bmatrix} = \begin{bmatrix} \frac{E_m}{1-\nu_m^2} & 0 & \frac{\nu_m E_m}{1-\nu_m^2} & 0 \\ 0 & C & 0 & 0 \\ \frac{\nu_m E_m}{1-\nu_m^2} & 0 & \frac{E_m}{1-\nu_m^2} & 0 \\ 0 & 0 & 0 & G_t \end{bmatrix} \begin{bmatrix} \Delta\epsilon_{11} \\ \Delta\epsilon_{22} - \Delta\epsilon_{22}^p \\ \Delta\epsilon_{33} \\ \Delta\gamma_{12} \end{bmatrix} \quad (7-38)$$

For large deformations in a total Lagrange formulation, in which the Green-Lagrange strains and the second Piola-Kirchhoff stresses are employed, as well as in an updated Lagrange environment, in which the logarithmic strains and Cauchy stresses are being used, similar but more complex relations can be derived.

Nonlinear Hypoelastic Material

The hypoelastic model is able to represent a nonlinear elastic (reversible) material behavior. For this constitutive theory, MSC.Marc assumes that

$$\dot{\sigma}_{ij} = L_{ijkl} \dot{\epsilon}_{kl} + g_{ij} \quad (7-39)$$

where L is a function of the mechanical strain and g is a function of the temperature.

The stress and strains are true stresses and logarithmic strains, respectively, when used in conjunction with the [UPDATE](#), [FINITE](#), and [LARGE DISP](#).

When used in conjunction with the [LARGE DISP](#) option only, [Equation 7-39](#) is expressed as

$$\dot{S}_{ij} = L_{ijkl} \dot{E}_{kl} + g_{ij} \quad (7-40)$$

where E , S are the Green-Lagrangian strain and second Piola-Kirchhoff stress, respectively.

The [HYPOELASTIC](#) model definition option is necessary to invoke this model. This model can be used with any stress element, including Herrmann formulation elements.

The tensors L and g are defined by you in the user subroutine [HYPELA](#). In order to provide an accurate solution, L should be a tangent stiffness evaluated at the beginning of the iteration. In addition, the total stress should be defined as its exact value at the end of the increment. This allows the residual load correction to work effectively.

In [HYPELA2](#), besides the functionality of [HYPELA](#), additional information is available regarding the kinematics of deformation. In particular, the deformation gradient (F), rotation tensor (R), and the eigenvalues (λ) and eigenvectors (N) to form the stretch tensor (U) are also provided. This information is available only for the continuum elements namely: plane strain, generalized plane strain, plane stress, axisymmetric, axisymmetric with twist, and three-dimensional cases.

For more information on the use of user subroutines, see [MSC.Marc Volume D: User Subroutines and Special Routines](#).

Thermo-Mechanical Shape Memory Model

NiTi alloys with near-equiatomic composition exhibit a reversible, thermoelastic transformation between a high-temperature, ordered cubic (B2) austenitic phase and a low-temperature, monoclinic (B19) martensitic phase. The density change and thus the volumetric are small and on the order of 0.003. The transformation strains are, thus mainly deviatoric, of the order of 0.07-0.085. However, these small dilational strains do not necessarily lead to a lack of pressure sensitivity in the phenomenology. The behavior of nitinol is different depending on whether the materials are subjected to hydrostatic tension or compression.

Typical phenomenology is shown in [Figure 7-10](#) taken from Miyazaki et al. (1981). The curves indicate that upon cooling, the material transformation from austenite to martensite begins once the M_s temperature is reached. Upon further cooling, the volume fraction on martensite is a given function of temperature; the volume fraction becomes 100% martensite when the M_f temperature is reached. Upon heating, transformation from martensite to austenite begins only after A_s temperature is reached. This re-transformation is complete when the A_f temperature is reached. Finally, note that the four transformation temperatures are stress dependent. The experimental data indicate the M_s , M_f , A_s , and A_f may be approximated from their stress-free values, M_s^0 , M_f^0 , A_s^0 , and A_f^0 by

$$M_s = M_s^0 + \frac{\sigma_{eq}}{C_m},$$

$$M_f = M_f^0 + \frac{\sigma_{eq}}{C_m}; \text{ and}$$

$$A_s = A_s^0 + \frac{\sigma_{eq}}{C_a},$$

$$A_f = A_f^0 + \frac{\sigma_{eq}}{C_a}.$$

where σ_{eq} is the von Mises equivalent stress. At a sufficiently high temperature, often called the M_d temperature, transformation to martensite does not occur at any level of stress.

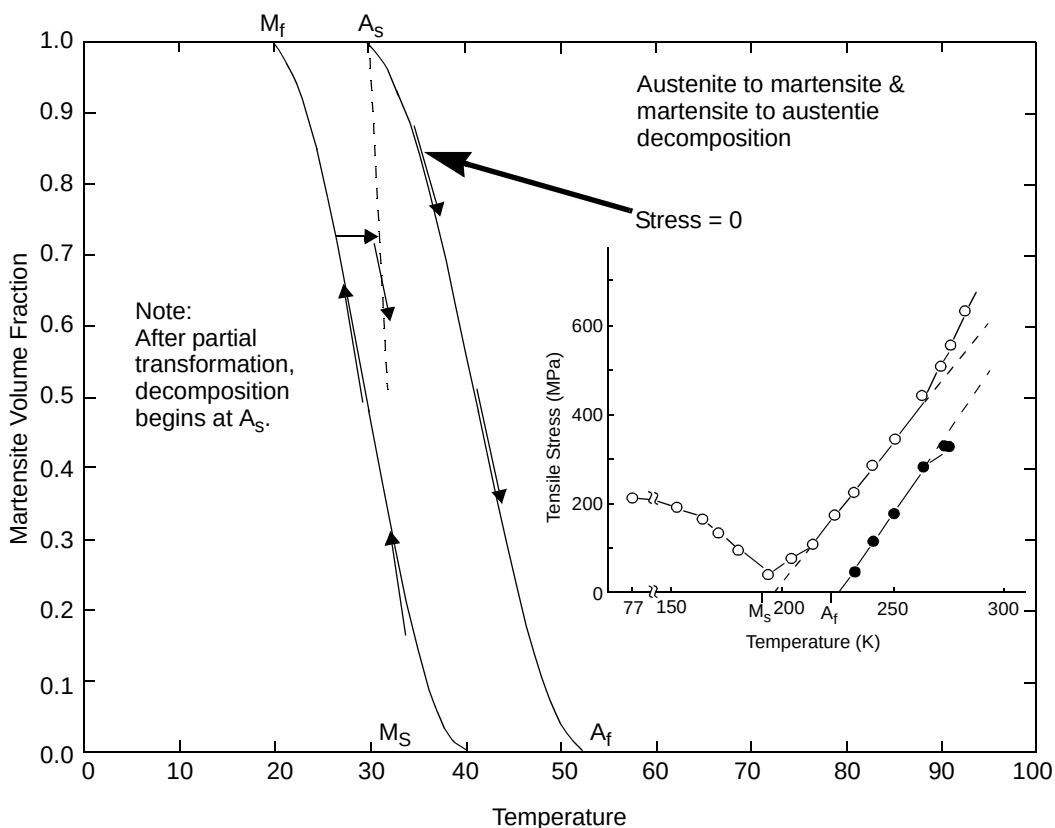


Figure 7-10 Austenite to Martensite and Martensite to Austenite Decomposition

The transformation characteristics such as the transformation temperatures depend sensitively on alloy composition and heat treatment.

Transformation Induced Deformation

For the discussion of the thermo-mechanical response of NiTi, the data of Miyazaki et al. (1981) is shown in [Figure 7-11](#). Following this thermal history, it is observed that, when unstrained specimens with fully austenitic microstructures are cooled, the transformation to martensite begins at a temperature of 190K; the transformation is

complete at 128K. This established the so-called martensite start (M_s^0) and martensite finish (M_f^0) temperatures at 190K and 128K, respectively. With the imposition of an applied uniaxial tensile stress, the low temperature martensite is favored and the M_s^0 and M_f^0 temperatures increase. Upon heating a specimen with fully martensitic microstructure, the reverse transformation is observed to begin at a temperature of 188K and to be complete at 221K. These define the austenite start (A_s^0) and austenite finish (A_f^0) temperatures, respectively. Uniaxial tension tests are carried out in temperature ranges where $T < M_s$, $M_s < T < A_f$, and $A_f < T < T_c$ where T_c is defined as the temperature above which the yield strength of the austenitic phase is lower than the stress required to induce the austenite-to-martensite transformation.

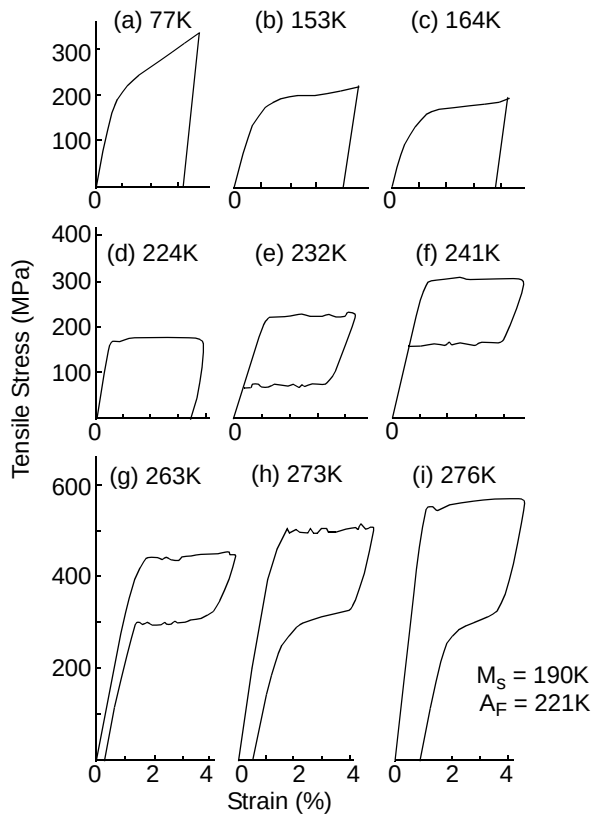


Figure 7-11 Thermal History

In the temperature range where $T < M_f$, the microstructures are all martensitic. The stress versus strain curves display a smooth parabolic type of behavior which is consistent with deformation caused by the movement of defects such as twin boundaries and the boundaries between variants. Note that unloading occurs nearly elastically and that the accumulated deformation, caused by the reorientation of the existing martensite and the transformation of any pre-existing austenite, remains after the specimen is completely unloaded. Note also that the accumulated deformation is entirely due to oriented martensite and this would be recoverable upon heating to temperatures above the $(A_s - A_f)$ range. This would, then, display the shape memory effect.

Pseudoelastic behavior is displayed in the temperature range $A_f < T < T_c$. In this range, the initial microstructures are essentially all austenitic, and stress induced martensite is formed, along with the associated deformation; upon unloading, however, the martensite is unstable and reverts to austenite thereby undoing the accumulated deformation. Note that, as expected, the stress levels rise with increasing temperature. In this range, the transformation induced deformation is nearly all reversible upon unloading.

At temperatures where $T > T_c$, plastic deformation appears to precede the formation of stress induced martensite. The unloading part of the stress versus strain behavior displays nonlinearity and the unloading is now associated with permanent (plastic) deformation. Permanent deformation, due to plastic deformation of the martensite, is non-recoverable and, if such deformation is large, shape memory behavior is lost.

Constitutive Theory

The model formulated below is based on the kinematics of small strains, although the extension to large strain is straightforward. Accordingly, the incremental strain, $\Delta\epsilon$, is simply the sum of the following contributions:

$$\Delta\epsilon = \Delta\epsilon^{el} + \Delta\epsilon^{th} + \Delta\epsilon^{pl} + \Delta\epsilon^{ph} \quad (7-41)$$

In Equation 7-41, $\Delta\epsilon^{el}$ is the incremental elastic, or lattice, strain rate; $\Delta\epsilon^{th}$ is the incremental thermal strain, $\Delta\epsilon^{pl}$ is the incremental visco-plastic strain, and $\Delta\epsilon^{ph}$ is the incremental strain associated with thermoelastic phase transformations. The incremental elastic strain is taken to be simply related to a set of elastic moduli, L , and the incremental Cauchy stress rate, $\Delta\sigma$, as

$$\Delta\sigma = L\Delta\epsilon \quad (7-42)$$

To calculate the coefficient of thermal expansion of the composite, the rule of mixtures is used as

$$\alpha = (1 - f)\alpha^A + f\alpha^M.$$

In the above equations, the superscripts A and M refer to the austenite and martensite values, and f is the volume fraction of martensite.

Phase Transformation Strains

As noted earlier, the phase transformation induced strains are a result of the formation of oriented, stress induced, martensite and the reorientation of randomly oriented thermally induced martensite. To account for this, $\Delta\epsilon^{Ph}$ is expressed as

$$\Delta\epsilon^{Ph} = \Delta\epsilon^{TRIP} + \Delta\epsilon^{TWIN}$$

where

$$\Delta\epsilon^{TRIP} = \Delta f^{(+)}g(\sigma_{eq})\epsilon_{eq}^T \frac{3}{2} \frac{\sigma'}{\sigma_{eq}} + \Delta f^{(+)}\epsilon_V^T I + \Delta f^{(-)}\epsilon^{Ph}. \quad (7-43)$$

and

$$\Delta\epsilon^{TWIN} = f\Delta g(\sigma_{eq})\epsilon_{eq}^T \frac{3\sigma'}{2\sigma_{eq}} \{\sigma_{eq}\} \{\sigma_{eq} - \sigma_{eff}^g\}. \quad (7-44)$$

where $\Delta f = \Delta f^{(+)} + \Delta f^{(-)}$ and $\{ \}$ represents McCauley's bracket where

$$\{x\} = \frac{1}{2} \left(\frac{x + |x|}{|x|} \right), \quad x \neq 0.$$

In Equation 7-43, $\Delta f^{(+)}$ represents the rate at which martensite is formed, ϵ_{eq}^T is the magnitude of the deviatoric part of the transformation, and ϵ_V^T is the volumetric part of the transformation strain. The function $g(\sigma_{eq})$ is schematically depicted in Figure 7-12, and is a measure of the extent to which the martensite transformation strains are aligned with the deviatoric stress. σ_{eq} is the equivalent stress defined as:

$$\sigma_{eq} = \sqrt{\frac{3}{2} \sigma' : \sigma'} \quad \text{where } \sigma' \text{ is the deviatoric stress.}$$

The first two terms in Equation 7-43 describe the development of transformation induced strains due to the formation of stress induced (partially oriented) martensite. $\Delta f^{(-)}$ is the change of formation of austenite; for example, the rate at which the volume fraction of martensite decreases. The last term in Equation 7-43, therefore, represents the recovery of the accumulated phase transformation strain.

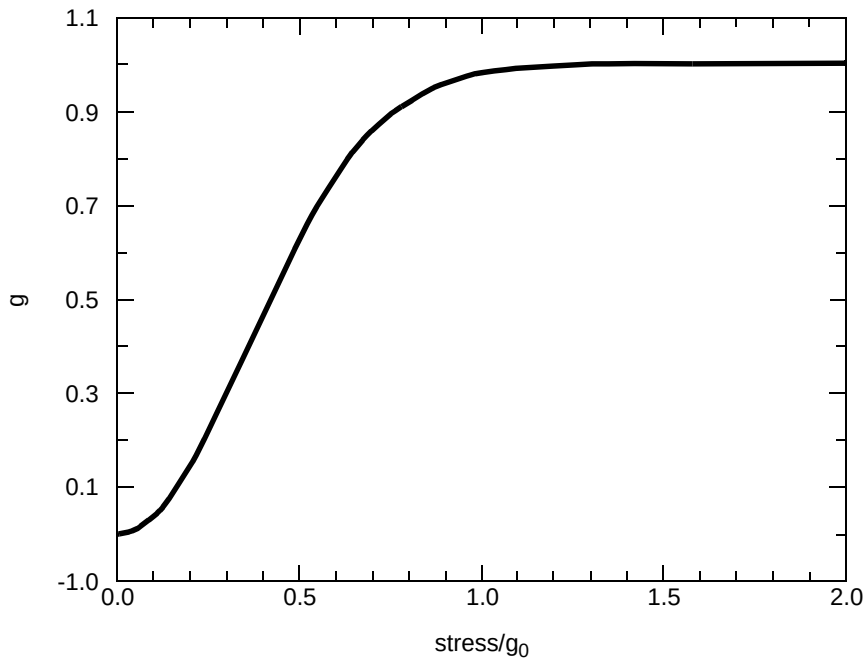


Figure 7-12 Function $g(\epsilon_q)$

Note that there is no dilatational contribution to $\Delta \epsilon^{TWIN}$ since f is fixed. Note that the twinning strain rate is zero when ϵ_{eq} is less than σ_{eff}^g , or when the magnitude of the stress decreases ($\sigma_{eq} < 0$). Hence, σ_{eff}^g can be considered as a stress below which no twinning is possible.

The function g represents the extent to which the transformation strains are coaxial with the applied deviatoric stress. This function can be calibrated with the experimental data. Note for uniaxial stress-strain curves performed below the martensite finish temperature, the material starts as 100% martensite, and that other than elastic strains, the deformation is dominated by the “twinning” of the randomly oriented martensite.

A functional form that leads to sufficient fit to most experimental data has been implemented in MSC.Marc.

$$g(\sigma_{eq}) = 1 - \exp \left[g_a \left(\frac{\sigma_{eq}}{g_0} \right)^{g_b} + g_c \left(\frac{\sigma_{eq}}{g_0} \right)^{g_d} + g_e \left(\frac{\sigma_{eq}}{g_0} \right)^{g_f} \right]$$

In most cases, the first term is sufficient, and a value of $g_a < 0$ and $g_b = 2$ yields the best results. g_0 is a stress level used to non-dimensionalize the constants, and can be chosen such that $g \rightarrow 1$ when the $\sigma_{eq} \rightarrow g_0$. In some cases, it is necessary to include the higher powers of equivalent stress for better experimental fits. In these cases suggested values for $g_d = 2.55$ or 2.75 and $g_e = 3$. However, depending on the values of g_c and g_d , this could lead to maxima or minima values of g in the range of interest. Note that $0 \leq g \leq 1$ and it should be a monotonically increasing function (an increase in the stress level should lead to an increase in the increment of the phase strains). Thus, cut off values of g are provided in MSC.Marc, such that when g reaches its maximum value $g = g_{max}$ at the stress level $\sigma_{eq} = \sigma_{max}^g g_0$, it is held constant at the value g_{max} .

Experimental Data Fitting for Thermo-mechanical Shape Memory Alloy

The properties and transformation/retransformation behavior of Nitinol depend upon alloy chemistry, microstructure and the thermal processing applied to the specimens and eventually to the components built from them. Every time any of the above change, it might be necessary to redo the calibration. Calibration of Nitinol experimental data works best if, in fact, all specimens can be initially rendered as 100% austenite. The list of properties that require calibration is given as follows:

- The “unstressed transformation temperatures”, $M_s^0, M_f^0, A_s^0, A_f^0$.
- The coefficients C_M , and C_A that provide the stress dependence of the transformation temperatures.
- The elastic constants (E^M, E^A, ν^M and ν^A).
- Coefficients of thermal expansion, α^M, α^A .
- The calibration of the detwinning function, $g(\sigma_{eq})$ that provides the description of the degree to which martensite is co-axial with the deviatoric stress state.

- The yield stress of the pure martensite and austenite phase (σ_Y^M and σ_Y^A), and their strain hardening properties.
- The calibration of the transformation strains, ϵ_{eq}^T and ϵ_V^T .

Transformation Temperatures and Their Stress Dependence ($M_s^0, M_f^0, A_s^0, A_f^0, C_M$, and C_A)

For almost any use of shape memory alloy, it is highly desirable that one knows the Transformation Temperatures (TTRs) of the alloy. The TTRs are those temperatures at which the alloy changes from the higher temperature austenite to the lower temperature martensite or vice versa. There are in common use with NiTi alloys to provide helpful data to product designers – Constant Load, DSC and Active A_f . The detail procedures to obtain TTRs using the above methods are shown in website (www.sma-inc.com/NiTiProperties.html). It is recommended that combined dilatometry and DSC tests be performed on unstressed specimens of thermally processed material to establish both the unstressed transformation temperatures and the thermal expansion coefficients. Those tests would provide a baseline set of values for M_s^0, M_f^0, A_s^0 , and A_f^0 . Note that the TTRs are stress dependent parameters, but it is difficult, in practice, to prepare totally unstressed samples. In order to determine the TTRs at zero stress, experimental data must be obtained at two or more stress levels. The particular transformation point of interest can then be extrapolated to zero stress. The estimations of TTRs, C_M , and C_A are shown in Figure 7-13. The typical range of TTRs is -200 to 100°C . So, it is difficult to recommend the default values. As references, there are examples for two different SMA materials below.

SMA 1)

$M_s^0 : -100^\circ\text{C}$, $M_f^0 : -50^\circ\text{C}$, $A_s^0 : 5^\circ\text{C}$, $A_f^0 : 20^\circ\text{C}$, $C_M : 6.0\text{Mpa}/^\circ\text{C}$,
 $C_A : 8.0\text{Mpa}/^\circ\text{C}$

SMA 2)

$M_s^0 : 190\text{K}$, $M_f^0 : 28\text{K}$, $A_s^0 : 188\text{K}$, $A_f^0 : 221\text{K}$, $C_M : 5.33\text{Mpa/K}$,
 $C_A : 6.25\text{Mpa/K}$

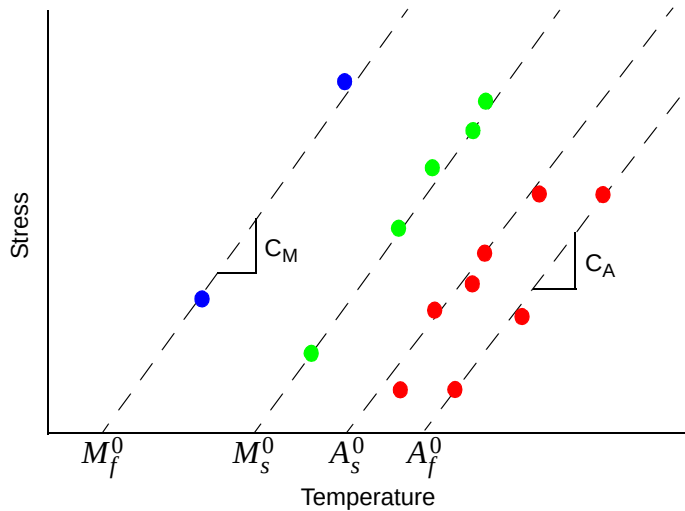


Figure 7-13 Typical Stress vs. Temperature Curve Showing the Stress Dependence of Martensite and Austenite Start and Finish Temperature

Elastic Constants (E^M , E^A , ν^M , and ν^A)

Literature estimates for the elastic moduli of martensite and austenite are typically in the range of $E^M = 28000\text{--}41000$ Mpa, $E^A = 60000\text{--}83000$ Mpa, $\nu^M = \nu^A = 0.33$. However, most experimental data appears to be significantly different than these. It is, therefore, suggested that estimates of these moduli should be made using actual experimental data for the materials being calibrated. Initial loading from a state corresponding to 100% austenite produces a linear elastic response from which E^A can be readily estimated as in [Figure 7-14](#). The modulus of martensite (E^M) is also estimated for the unloading line, again as illustrated in [Figure 7-14](#). In this figure, the loading should be performed to produce 100% martensite and thus the unloading occurs with the elastic response of martensite.

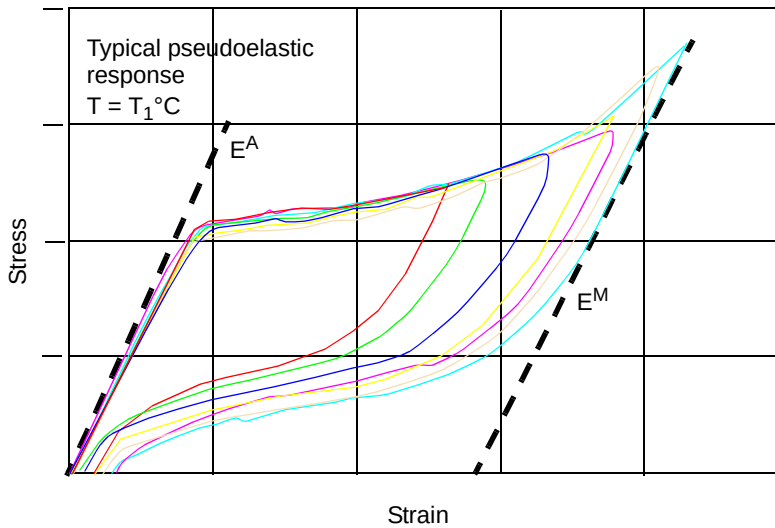


Figure 7-14 Typical Stress-strain Curves in the Pseudo-elastic Regime, Depicting the Elastic Moduli

Thermal expansion coefficients (α^M, α^A)

A recommended method for measuring thermal expansion coefficients is through the use of dilatometry whereby carefully controlled cycles of temperature variations can be made. Barring this type of precise calibration, it is possible to use literature values that have been shown to be consistent with values measured on actual specimens.

These values are given as follows (See for example, TiNi Smart Sheet)

$$\alpha^M = 6.6 \times 10^{-6}/^{\circ}\text{C} = 3.67 \times 10^{-6}/^{\circ}\text{F}; \text{ and}$$

$$\alpha^A = 11.0 \times 10^{-6}/^{\circ}\text{C} = 6.11 \times 10^{-6}/^{\circ}\text{F}$$

Detwinning and the calibration of the g function $g(\sigma_{eq})$

The phenomenology of the NiTi phase transformation is such that the alignment of the martensite varies with the prevailing deviatoric stress. This intensity is measured via the von Mises equivalent stress, σ_{eq} . As shown in Equations 7-43 and 7-44, the scaling function that provides the description of the degree to which the martensite is aligned is $g(\sigma_{eq})$. The most direct path to calibrating this g function is to fit it to the uniaxial stress vs. strain curve for pure, randomly oriented martensite conducted at a temperature below the M_f^0 temperature. Such a curve is shown as Figure 7-15. The solid curve shown in Figure 7-15 is the actual measured record of uniaxial stress vs. total strain for a specimen of 100% martensite tested at a temperature sufficiently low

to ensure it remains 100% martensite. The dot curve is simply a convenient fit to it. The parameter ε_{eq}^T is the “equivalent transformation strain”. Note that the function g is defined as it relates to the development of deviatoric strain due to the alignment of martensite variants. As mentioned in the previous section, in general, the variables $g_a < 0$, $g_b = 2f = 3.0$, $g_c \geq 0$, $g_d = 2.25$ and 2.75 , $g_e \leq 0$ and $g_f = 3.0$ yield a good match to many experimental results. It is often observed that there exists a threshold equivalent stress level below which detwinning does not occur; this stress is referred to as σ_{eff}^g . The value of g at this stress is $g_{eff} = g(\sigma_{ff}^g)$. Note that from [Equation 7-44](#), twinning strain is zero when $\sigma_{eq} < \sigma_{eff}^g$. In addition, it is also found, in practice, that the function g tends to approach unit as a finite equivalent stress level, called σ_0^g . By definition, $g(\sigma_0^g) = 1$. Also, g_a should be chosen to match the general shape of the function. Since the ratio σ_{eq}/g_0 is less than one, the higher powers take effect later, and thus g_c can be added to lower the middle slope of the curve and g_e to fix the final slope of the curve. However, depending on the relative values of g_a , g_b and g_c , this curve might reach a maximum in the range of interest, and therefore, it should be cut-off at its maximum value g_{max} . The value of g_{max} which is reached at a stress value $\sigma_{eq}/g_0 = \sigma_{max}^g$ are also supplied as an input to MSC.Marc.

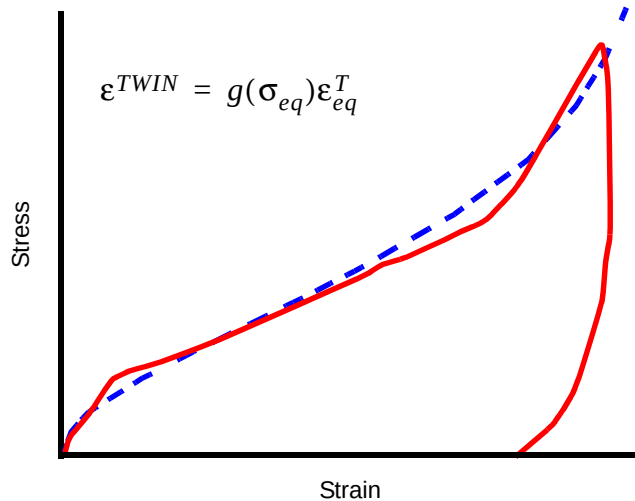


Figure 7-15 Typical Stress-strain Curve of 100% Martensite Tested Below M_f^0 Temperature

Others

Yield stresses of the pure martensite and austenite phases for NiTi: σ_Y^M and σ_Y^A .

$$\sigma_Y^M = 70-140 \text{ Mpa}$$

$$\sigma_Y^A = 195 - 690 \text{ Mpa}$$

The calibration of the transformation strains for NiTi:

$$\varepsilon_{eq}^T \quad (\text{deviatoric transformation strain}) : 0.05-0.085$$

$$\varepsilon_v^T \quad (\text{volumetric transformation strain}) : 0 - 0.003.$$

$$\sigma_{eq} \quad (\text{detwinning stress}) : 100-150 \text{ Mpa}$$

Notes: The current model only supports $ndi = 3$ case (3-D, plane-strain and axisymmetric elements). It does not support either $ndi = 1$ or $ndi = 2$ cases (1-D and plane-stress elements).

The current model uses a nonsymmetric Jacobian matrix. It is recommended that the nonsymmetric solver be used to improve convergence.

Mechanical Shape Memory Model

Shape-memory alloys can undergo reversible changes in the crystallographic symmetry-point-group. Such changes can be interpreted as martensitic phase transformations, that is, as solid-solid diffusion-less phase transformations between a crystallographically more-ordered phase (the austenite or parent phase) and a crystallographically less-ordered phase (the martensite). Typically, the austenite is stable at high temperatures and high values of the stress. For a stress-free state, we indicate with the temperature above which only the austenite is stable and with the temperature below which only the martensite is stable.

The phase transformations between austenite and martensite are the key to explain the superelasticity effect. For the simple case of uniaxial tensile stress, a brief explanation follows (Figure 7-16). Consider a specimen in the austenitic state and at a temperature greater than; accordingly, only the austenite is stable at zero stress. If the specimen is loaded, while keeping the temperature constant, the material presents a nonlinear behavior (ABC) due to stress-induced conversion of austenite into martensite. Upon unloading, while again keeping the temperature constant, a reverse transformation from martensite to austenite occurs (CDA) as a result of the instability of the martensite at zero stress. At the end of the loading/unloading process, no permanent strains are present and the stress-strain path is a closed hysteresis loop.

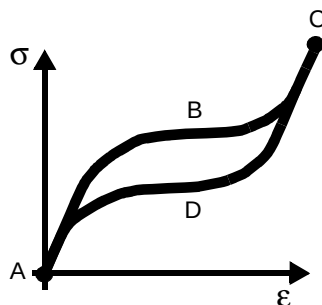


Figure 7-16 Superelasticity

At the crystallographic level, if there is no preferred direction for the occurrence of the transformation, the martensite takes advantage of the existence of different possible habit plates (the contact plane between the austenite and the martensite during a single-crystal transformation), forming a series of crystallographically equivalent variants. The product phase is then termed multiple-variant martensite and it is characterized by a twinned structure. However, if there is a preferred direction for

the occurrence of the transformation (often associated with a state of stress), all the martensite crystals trend to be formed on the most favorable habit plane. The product is then termed single-variant martensite and is characterized by a detwinned structure. According to the existence of different types of single-variant martensite species, the conversion of each single-variant martensite into different single variants is possible. Such a process, known as a reorientation process, can be interpreted as a family of martensite phase transformations and is associated with changes in the parameters governing the single-variant martensite production (hence, it is often associated to nonproportional change of stress).

In addition to the thermo-mechanical shape memory model, a superelastic shape memory alloy model is also implemented in MSC.Marc based on the work of Auricchio [Ref. 1-2].

The superelastic shape memory model has been implemented in MSC.Marc in the framework of multiplicative decomposition. We assume the deformation gradient, F as the control variable, and the martensite fraction, ξ_s as the only scalar internal variable. We also introduce a multiplicative decomposition of F in the form:

$$F = F^e F^{tr}$$

where F^e is the elastic part of F^{tr} is the phase transition part.

Assuming an isotropic elastic response, the Kirchhoff stress τ and the elastic left Cauchy-Green tensor b^e , defined as: $b^e = F^e F^{eT}$, share the same principal directions. Accordingly, the following spectral decompositions can be introduced;

$$\tau = \sum_{A=1}^3 \tau_A n^A \otimes n^A$$

$$t = \sum_{A=1}^3 t_A n^A \otimes n^A$$

$$b^e = \sum_{A=1}^3 (\lambda_A^e)^2 n^A \otimes n^A$$

with λ_A^e the elastic principal stretches and t the deviatoric part, according to the relation:

$$\tau = pI + t \quad (7-45)$$

where I is the second-order identity tensor, p is the pressure, defined as

$p = \text{tr}(\tau)/3$, and $\text{tr}(\cdot)$ is the trace operator. We can write Equation 7-45 with the following component form:

$$\tau_A = p + t_A \quad (7-46)$$

with

$$p = K\theta^e, \quad t_A = 2Ge_A^e.$$

Phase Transformations and Activation Conditions

We consider two phase transformations: the conversion of austenite into martensite ($A \rightarrow S$) and the conversion of martensite into austenite ($S \rightarrow A$). To model the possible phase-transformation pressure-dependence, we introduce a Drucker-Prager-type loading function:

$$F(\tau) = \|t\| + 3\alpha p \quad (7-47)$$

where α is a material parameter and $\|\cdot\|$ indicates the Euclidean norm, such that:

$$\|t\| = \left[\sum_{A=1}^3 (t_A)^2 \right]^{1/2}.$$

Indicating variants in time with a superposed dot, we assume the following linear forms for the evolution of ξ_S :

$$\dot{\xi}_S = H^{AS}(1 - \xi_S) \frac{\dot{F}}{F - R_f^{AS}} \quad \text{for } (A \rightarrow S) \quad (7-48)$$

$$\dot{\xi}_S = H^{SA}\xi_S \frac{\dot{F}}{F - R_f^{AS}} \quad \text{for } (S \rightarrow A) \quad (7-49)$$

where

$$R_f^{AS} = \left[\sigma_f^{AS} \left(\sqrt{\frac{2}{3}} + \alpha \right) \right], R_f^{SA} = \left[\sigma_f^{SA} \left(\sqrt{\frac{2}{3}} + \alpha \right) \right]$$

with σ_s^{AS} , σ_f^{AS} , σ_s^{SA} , and σ_f^{SA} material constants. The scalar quantities H^{AS} and H^{SA} embed the plastic-transformation activation condition – hence, allowing a choice between Equation 7-48 and Equation 7-49 – and they are defined by the relations:

$$H^{AS} = 1, \text{ if } R_s^{AS} < F < R_f^{AS}, \text{ or } \dot{F} > 0. \text{ Otherwise, } H^{AS} = 0.$$

$$H^{SA} = 1, \text{ if } R_f^{SA} < F < R_s^{SA}, \text{ or } \dot{F} < 0. \text{ Otherwise, } H^{SA} = 0.$$

where

$$R_s^{AS} = \left[\sigma_s^{AS} \left(\sqrt{\frac{2}{3}} + \alpha \right) \right], R_s^{SA} = \left[\sigma_s^{SA} \left(\sqrt{\frac{2}{3}} + \alpha \right) \right].$$

Time-discrete Model

The time-discrete model is obtained by integrating the time-continuous model over the time interval $[t_n, t]$. In particular, we use a backward-Euler integration formula for the rate-equations evaluating all the nonrate equations at time t . Written in residual form and clearing fraction from Equations 7-48 and 7-49, the time-discrete evolutionary equations specialize to:

$$R^{AS} = (F - R_f^{AS})\lambda_s - H^{AS}(1 - \xi_s)(F - F_n) = 0 \quad (7-50)$$

$$R^{SA} = (F - R_f^{SA})\lambda_s - H^{SA}\xi_s(F - F_n) = 0 \quad (7-51)$$

where

$$\lambda_s = \int_{t_n}^t \dot{\xi} dt = \xi_s - \xi_{s,n} \quad (7-52)$$

The quantity λ_s in Equation 7-52 can be computed expressing F as a function of λ_s and requiring the satisfaction of the discrete equation relative to the corresponding active phase transition.

The detailed solution algorithm for stress update and consistent tangent modulus are given in the work of Auricchio [Ref. 2].

Experimental Data Fitting for Mechanical Shape Memory Alloy

The experiment for mechanical shape memory alloy is quite simple.

1. To determine the transformation stresses ($\sigma_S^{AS}, \sigma_f^{AS}, \sigma_S^{SA}, \sigma_f^{SA}$):

σ_S^{AS} : Initial Stress for Austenite to Martensite

σ_f^{AS} : Final Stress for Austenite to Martensite

σ_S^{SA} : Initial Stress for Martensite to Austenite

σ_f^{SA} : Initial Stress for Martensite to Austenite

Uniaxial tension test is performed at the same temperature at which the simulation is desired. Here is one example set for SMA materials.

$$\sigma_S^{AS} = 500\text{Mpa}, \sigma_f^{AS} = 600\text{Mpa}, \sigma_S^{SA} = 300\text{Mpa}, \sigma_f^{SA} = 200\text{Mpa}$$

2. α : It is measured from the difference between the response in tension and compression.

Case 1) if the behavior in tension and compression are the same, the value is set to 0.

Case 2) if the behavior in tension and compression have a difference as in the classical case of SMA, the value is usually set to 0.1 if there is no compression data for the phase transformation. One value for the phase transformation in compression, say σ_S^{AS-}

(sigAS_s_compression) is available, α is calculated as follows:

$$\alpha = \text{sqrt}(2/3)(\sigma_S^{AS,-} - \sigma_S^{AS})/(\sigma_S^{AS,-} + \sigma_S^{AS})$$

3. ϵ_L : epsL is a scalar parameter representing the maximum deformation obtainable only by detwinning of the multiple-variant martensite (or maximum strain obtainable by variant orientation). Classical values for epsL are in the range 0.005 and 0.10. Marc sets the default value as 0.07.

Note: The current model only supports ndi = 3 case (3-D, plane-strain and axisymmetric elements). It does not support either ndi = 1 or ndi = 2 cases (1-D and plane-stress elements).

Elastomer

An elastomer is a polymer which shows nonlinear elastic stress-strain behavior. The term elastomer is often used to refer to materials which show a rubber-like behavior, even though no rubbers exist which show a purely elastic behavior. Depending upon the type of rubber, elastomers show a more or less strongly pronounced viscoelastic behavior. MSC.Marc considers both the viscous effects and the elastic aspects of the materials behavior. These materials are characterized by their elastic strain energy function.

Elastomeric materials are elastic in the classical sense. Upon unloading, the stress-strain curve is retraced and there is no permanent deformation. Elastomeric materials are initially isotropic. Figure 7-17 shows a typical stress-strain curve for an elastomeric material.

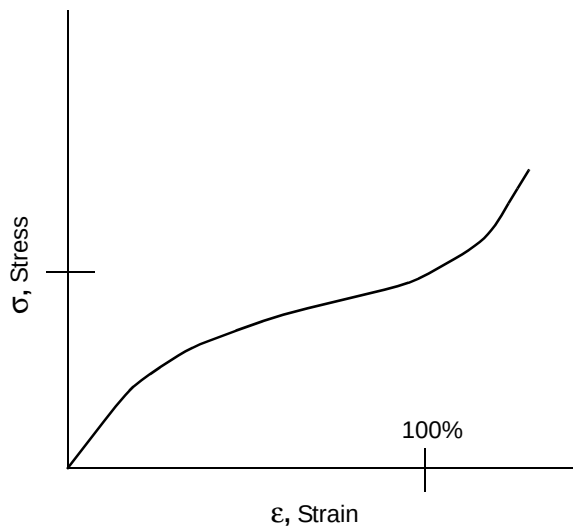


Figure 7-17 A Typical Stress-Strain Curve for an Elastomeric Material

Calculations of stresses in an elastomeric material requires an existence of a strain energy function which is usually defined in terms of invariants or stretch ratios. Significance and calculation of these kinematic quantities is discussed next.

In the rectangular block in Figure 7-18, λ_1 , λ_2 , and λ_3 are the principal stretch ratios along the edges of the block defined by

$$\lambda_i = (L_i + u_i)/L_i \quad (7-53)$$

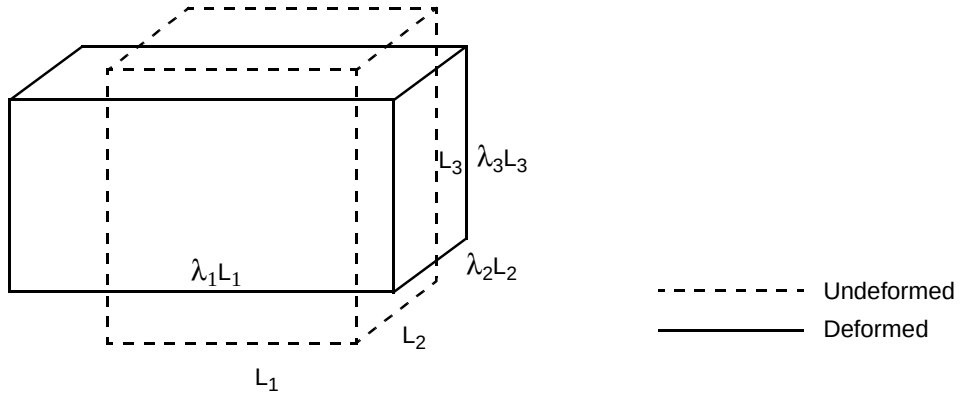


Figure 7-18 Rectangular Rubber Block

In practice, the material behavior is (approximately) incompressible, leading to the constraint equation

$$\lambda_1 \lambda_2 \lambda_3 = 1 \quad (7-54)$$

the strain invariants are defined as

$$\begin{aligned} I_1 &= \lambda_1^2 + \lambda_2^2 + \lambda_3^2 \\ I_2 &= \lambda_1^2 \lambda_2^2 + \lambda_2^2 \lambda_3^2 + \lambda_3^2 \lambda_1^2 \\ I_3 &= \lambda_1^2 \lambda_2^2 \lambda_3^2 \end{aligned} \quad (7-55)$$

Depending on the choice of configurations, for example, reference (at $t = 0$) or current ($t = n + 1$), you obtain total or updated Lagrange formulations for elasticity. The kinematic measures for the two formulations are discussed next.

A. Total Lagrangian Formulation

The strain measure is the Green-Lagrange strain defined as:

$$E_{ij} = \frac{1}{2}(C_{ij} - \delta_{ij}) \quad (7-56)$$

where C_{ij} is the right Cauchy-Green deformation tensor defined as:

$$C_{ij} = F_{ki} F_{kj} \quad (7-57)$$

in which F_{kj} is the deformation gradient (a two-point tensor) written as:

$$F_{kj} = \frac{\partial x_k}{\partial X_j} \quad (7-58)$$

The Jacobian J is defined as:

$$J = \lambda_1 \lambda_2 \lambda_3 = (\det C_{ij})^{\frac{1}{2}} \quad (7-59)$$

Thus, the invariants can be written as:

$$\begin{aligned} I_1 &= C_{ii} && \text{(implied sum on i)} \\ I_2 &= \frac{(C_{ij}C_{ij} - (C_{ii})^2)}{2} \\ I_3 &= \frac{1}{6} e_{ijk} e_{pqr} C_{ip} C_{jq} C_{kr} = \det (C_{ij}) \end{aligned} \quad (7-60)$$

in which e_{ijk} is the permutation tensor. Also, using spectral decomposition theorem,

$$C_{ij} = \lambda_A^2 N_i^A N_j^A \quad (7-61)$$

in which the stretches λ_A^2 are the eigenvalues of the right Cauchy-Green deformation tensor, C_{ij} and the eigenvectors are N_i^A .

B. Updated Lagrange Formulation

The strain measure is the true or logarithmic measure defined as:

$$\epsilon_{ij} = \frac{1}{2} \ln b_{ij} \quad (7-62)$$

where the left Cauchy-Green or finger tensor b_{ij} is defined as:

$$b_{ij} = F_{ik} F_{jk} \quad (7-63)$$

Thus, using the spectral decomposition theorem, the true strains are written as:

$$\epsilon_{ij} = \frac{1}{2} (\ln \lambda_A) n_i^A n_j^A \quad (7-64)$$

where n_i^A is the eigenvectors in the current configuration. It is noted that the true strains can also be approximated using first Padé approximation, which is a rational expansion of the tensor, as:

$$\varepsilon_{ij} = 2(V_{ij} - \delta_{ij})(V_{ij} + \delta_{ij})^{-1} \quad (7-65)$$

where a polar decomposition of the deformation gradient F_{ij} is done into the left stretch tensor V_{ij} and rotation tensor R_{ij} as:

$$F_{ij} = V_{ik}R_{kj}$$

The Jacobian J is defined as:

$$J = \lambda_1\lambda_2\lambda_3 = (\det b_{ij})^{\frac{1}{2}} \quad (7-66)$$

and the invariants are now defined as:

$$\begin{aligned} I_1 &= b_{ii} \\ I_2 &= \frac{1}{2}(b_{ij}b_{ij} - (b_{ii})^2) \end{aligned} \quad (7-67)$$

$$\text{and } I_3 = \frac{1}{6} e_{ijk}e_{pqr}b_{ip}b_{jq}b_{kr} = \det(b_{ij})$$

It is noted that either Equation 7-60 or Equation 7-67 gives the same strain energy since it is scalar and invariant. Also, to account for the incompressibility condition, in both formulations, the strain energy is split into deviatoric and volumetric parts as:

$$W = W_{\text{deviatoric}} + W_{\text{volumetric}} \quad (7-68)$$

Thus, the generalized Mooney-Rivlin (gmr) and the Ogden models for nearly-incompressible elastomeric materials are written as:

$$W_{\text{deviatoric}}^{\text{gmr}} = \sum_{m=1}^N \sum_{n=1}^N C_{mn} (\bar{I}_1 - 3)^m (\bar{I}_2 - 3)^n \quad (7-69)$$

where $\bar{I}_1$ and $\bar{I}_2$ are the first and second deviatoric invariants.

A particular form of the generalized Mooney-Rivlin model, namely the third order deformation (tod) model, is implemented in MSC.Marc.

$$W_{\text{deviatoric}}^{\text{tod}} = C_{10}(\bar{I}_1 - 3) + C_{01}(\bar{I}_2 - 3) + C_{11}(\bar{I}_1 - 3)(\bar{I}_2 - 3) + C_{20}(\bar{I}_1 - 3)^2 + C_{30}(\bar{I}_1 - 3)^3 \quad (7-70)$$

where

$W_{\text{deviatoric}}^{\text{tod}}$ is the deviatoric third order deformation form strain energy function,

$C_{10}, C_{01}, C_{11}, C_{20}, C_{30}$ are material constants obtained from experimental data

Simpler and popular forms of the above strain energy function are obtained as:

$$\begin{aligned} W_{\text{deviatoric}}^{\text{nh}} &= C_{10}(\bar{I}_1 - 3) && \text{Neo-Hookean} \\ W_{\text{deviatoric}}^{\text{mr}} &= C_{10}(\bar{I}_1 - 3) + C_{01}(\bar{I}_2 - 3) && \text{Mooney-Rivlin} \end{aligned} \quad (7-71)$$

Use the **MOONEY** model definition option to activate the elastomeric material option in MSC.Marc and enter the material constants $C_{10}, C_{01}, C_{11}, C_{20}, C_{30}$.

The **TEMPERATURE EFFECTS** model definition option can be used to input the temperature dependency of the constants C_{10} and C_{01} . The user subroutine **UMOONY** can be used to modify all five constants $C_{01}, C_{10}, C_{11}, C_{20}$, and C_{30} . For viscoelastic, the additional model definition option **VISCELMOON** must be included.

The form of strain energy for the Ogden model in MSC.Marc is,

$$W_{\text{deviatoric}}^{\text{ogden}} = \sum_{k=1}^N \frac{\mu_k}{\alpha_k} (\bar{\lambda}_1^{\alpha_k} + \bar{\lambda}_2^{\alpha_k} + \bar{\lambda}_3^{\alpha_k} - 3) \quad (7-72)$$

where $\bar{\lambda}_i^{\alpha_k} = J^{-\frac{\alpha_k}{3}} \lambda_i^{\alpha_k}$ are the deviatoric stretch ratios while C_{mn}, μ_k , and α_k are the material constants obtained from the curve fitting of experimental data. This capability is available in MSC.Marc Mentat.

If no bulk modulus is given, it is taken to be virtually incompressible. This model is different from the Mooney model in several respects. The Mooney material model is with respect to the invariants of the right or left Cauchy-Green strain tensor and implicitly assumes that the material is incompressible. The Ogden formulation is with

respect to the eigenvalues of the right or left Cauchy-Green strain, and the presence of the bulk modulus implies some compressibility. Using a two-term series results in identical behavior as the Mooney mode if:

$$\mu_1 = 2C_{10} \text{ and } \alpha_1 = 2 \text{ and } \mu_2 = -2C_{01} \text{ and } \alpha_2 = -2$$

The material data is given through the **OGDEN** model definition option or the user subroutine **UOGDEN**. For viscoelastic behavior, the additional model definition option **VISCELOGDEN** must be included.

In the Arruda-Boyce strain energy model, the underlying molecular structure of elastomer is represented by an eight-chain model to simulate the non-Gaussian behavior of individual chains in the network. The two parameters, $nk\Theta$ and N (n is the chain density, k is the Boltzmann constant, Θ is the temperature, and N is the number of statistical links of length ℓ in the chain between chemical crosslinks) representing initial modules and limiting chain extensibility and are related to the molecular chain orientation thus representing the physics of network deformation.

As evident in most models describing rubber deformation, the strain energy function constructed by fitting experiment data obtained from one state of deformation to another fails to accurately describe that deformation mode. The Arruda-Boyce model ameliorates this defect and is unique since the standard tensile test data provides sufficient accuracy for multiple modes of deformation.

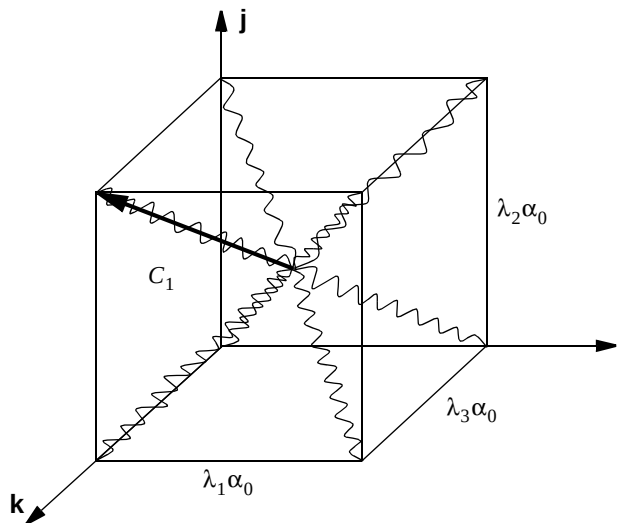


Figure 7-19 Eight Chain Network in Stretched Configuration

The model is constructed using the eight chain network as follows [Ref. 3]:

Consider a cube of dimension α_0 with an unstretched network including eight chains of length $r_0 = \sqrt{N}\ell$, where the fully extended chain has an approximate length of $N\ell$. A chain vector from the center of the cube to a corner can be expressed as:

$$C_1 = \frac{\alpha_0}{2} \lambda_1 \mathbf{i} + \frac{\alpha_0}{2} \lambda_2 \mathbf{j} + \frac{\alpha_0}{2} \lambda_3 \mathbf{k} \quad (7-73)$$

Using geometrical considerations, the chain vector length can be written as:

$$r_{\text{chain}} = \frac{1}{\sqrt{3}} \sqrt{N} \ell (\lambda_1^2 + \lambda_2^2 + \lambda_3^2)^{1/2} \quad (7-74)$$

and

$$\lambda_{\text{chain}} = \frac{r_{\text{chain}}}{r_0} = \frac{1}{\sqrt{3}} (I_1)^{1/2} \quad (7-75)$$

Using statistical mechanics considerations, the work of deformation is proportional to the entropy change on stretching the chains from the unstretched state and may be written in terms of the chain length as:

$$W = nk\Theta N \left(\frac{r_{\text{chain}}}{N\ell} \beta + \ell n \frac{\beta}{\sinh \beta} \right) - \Theta \hat{C} \quad (7-76)$$

where n is the chain density and $\hat{C}$ is a constant. β is an inverse Langevin function correctly accounts for the limiting chain extensibility and is defined as:

$$\beta = L^{-1} \left(\frac{r_{\text{chain}}}{N\ell} \right) \quad (7-77)$$

where Langevin is defined as:

$$\mathfrak{L}(\beta) = \coth \beta - \frac{1}{\beta} \quad (7-78)$$

With [Equation 7-75](#) through [Equation 7-78](#), the Arruda-Boyce model can be written

$$W_{\text{dev}}^{\text{Arruda-Boyce}} = nk\Theta \left[\frac{1}{2}(\bar{I}_1 - 3) + \frac{1}{20N}(\bar{I}_1^2 - 9) + \frac{11}{1050N^2}(\bar{I}_1^3 - 27) \right. \\ \left. + \frac{19}{7000N^3}(\bar{I}_1^4 - 81) + \frac{519}{673750N^4}(\bar{I}_1^5 - 243) \right] \quad (7-79)$$

Also, using the notion of limiting chain extensibility, Gent [\[Ref. 5\]](#) proposed the following constitutive relation:

$$W_{\text{dev}}^{\text{Gent}} = \frac{-EI_m}{6} \log \left[\frac{I_m}{I_m - I_1^*} \right] \quad (7-80)$$

where

$$I_1^* = \bar{I}_1 - 3 \quad (7-81)$$

The constant EI_m is independent of molecular length and, hence, of degree of crosslinking. The model is attractive due to its simplicity, but yet captures the main behavior of a network of extensible molecules over the entire range of possible strains.

The Arruda-Boyce and Gent model can be invoked by using the [ARRUDBOYCE](#) and [GENT](#) model definition options, respectively.

The volumetric part of the strain energy is for all the rubber models in MSC.Marc:

$$W_{\text{volumetric}} = \frac{9K}{2} \left(J^{\frac{1}{3}} - 1 \right)^2 \quad (7-82)$$

when K is the bulk modulus.

It can be noted that the particular form of volumetric strain energy is chosen such that:

- (i) The constraint condition is satisfied for incompressible deformations only; for example:

$$f(I_3) \begin{cases} > 0 & \text{if } I_3 > 0 \\ = 0 & \text{if } I_3 = 1 \\ < 0 & \text{if } I_3 < 0 \end{cases} \quad (7-83)$$

- (ii) The constraint condition does not contribute to the dilatational stiffness.

This yields the constraint function as:

$$f(I_3) = 3 \left(I_3^{\frac{1}{6}} - 1 \right) \quad (7-84)$$

upon substitution of [Equation 7-86](#) in [Equation 7-82](#) and taking the first variation of the variational principle, you obtain the pressure variable as:

$$p = 3K \left(J^{\frac{1}{3}} - 1 \right) \quad (7-85)$$

The equation has a physical significance in that for small deformations, the pressure is linearly related to the volumetric strains by the bulk modulus K .

The discontinuous or continuous damage models discussed in the models section on damage can be included with the generalized Mooney-Rivlin, Ogden, Arruda-Boyce, and Gent models to simulate Mullins effect or fatigue of elastomers when using the updated Lagrangian approach. In the total Lagrangian framework however, this is available for the Ogden model only.

The rubber foam model which is based on Ogden formulation has a strain energy form as follows:

$$W = \sum_{n=1}^N \frac{\mu_n}{\alpha_n} (\lambda_1^{\alpha_n} + \lambda_2^{\alpha_n} + \lambda_3^{\alpha_n} - 3) + \sum_{n=1}^N \frac{\mu_n}{\beta_n} (1 - J^{\beta_n}) \quad (7-86)$$

where μ_n , α_n , β_n are material constants. The model reduces to incompressible Ogden model when β_n equals zero.

You can define any other invariant based models through the use of the user subroutine **UENERG** when using the **MOONEY** option in model definition for elasticity in total or updated Lagrangian framework. A principal stretch-based strain energy function can also be defined through the user subroutine **UPSTRECH** when using the **OGDEN** option in the model definition for elasticity in the updated Lagrangian framework.

Once the strain energy function is defined, the stresses and material tangent can be evaluated for the total and updated Lagrangian formulations as:

A. Total Lagrangian Formulation:

The stress measure in the total Lagrangian formulation is the symmetric second Piola-Kirchhoff stress S_{ij} , calculated as:

$$S_{ij} = \frac{\partial W}{\partial E_{ij}} = 2 \frac{\partial W}{\partial C_{ij}} \quad (7-87)$$

The material elasticity tangent is:

$$D_{ijkl} = \frac{\partial^2 W}{\partial E_{ij} \partial E_{kl}} = 4 \frac{\partial^2 W}{\partial C_{ij} \partial C_{kl}} \quad (7-88)$$

B. Updated Lagrangian Formulation:

The stress measure in the updated Lagrangian formulation is the Cauchy or true stress calculated as:

$$\sigma_{ij} = \frac{2}{J} \frac{\partial W}{\partial b_{ik}} b_{kj} \quad (7-89)$$

The spatial elasticity tangent is:

$$L_{ijkl} = \frac{4}{J} b_{im} \frac{\partial^2 W}{\partial b_{mj} \partial b_{kn}} b_{nl} \quad (7-90)$$

The material constants for the Mooney-Rivlin form can be obtained from experimental data. The Mooney-Rivlin form of the strain energy density function is

$$W_{\text{deviatoric}}^{\text{mooney-rivlin}} = C_{10}(\bar{I}_1 - 3) + C_{01}(\bar{I}_2 - 3) \quad (7-91)$$

For the Mooney-Rivlin model, the force and deformation for a uniaxial test specimen can be related as

$$P = 2A_0 \left(1 - \frac{1}{\lambda_1^3}\right) (\lambda_1 C_{10} + C_{01}) \quad (7-92)$$

which can be written in the form:

$$\frac{P}{2A_0 \left(\lambda_1 - \frac{1}{\lambda_1^2}\right)} = C_{10} + \frac{C_{01}}{\lambda_1} \quad (7-93)$$

where P is the force of the specimen, A_0 is the original area of the specimen, and λ_1 is the uniaxial stretch ratio. This equation provides a simple way to determine the Mooney-Rivlin constants. The Mooney-Rivlin constitutive equation is applicable if the plot of $P / \left[2A_0 \left(\lambda_1 - \frac{1}{\lambda_1^2}\right)\right]$ versus $\frac{1}{\lambda_1}$ should yield a straight line of slope C_{01} and

intercept $(C_{01} + C_{10})$ on the vertical axis $\frac{1}{\lambda_1} = 1$ as shown in [Figure 7-20](#).

If only the Young's modulus E is supplied, and full uniaxial data are not available then

$$C_{01} \cong 0.25C_{10} \quad (7-94)$$

is a reasonable assumption. The constants then follow from the relation:

$$6(C_{10} + C_{01}) \cong E \quad (7-95)$$

The material coefficients for the models can be obtained from MSC.Marc Mentat. This allows you to select which model is most appropriate for your data.

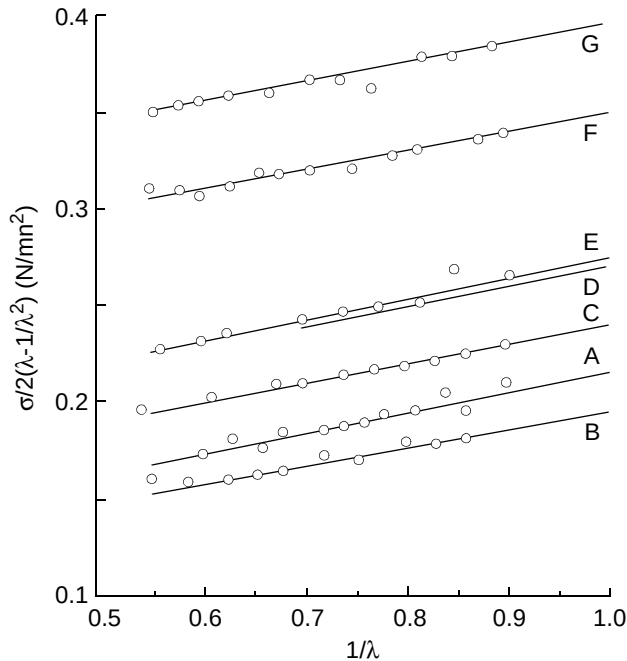


Figure 7-20 Plots for Various Rubbers in Simple Extension for Mooney-Rivlin Model

Updated Lagrange Formulation for Nonlinear Elasticity

The total Lagrange nonlinear elasticity models in MSC.Marc have been augmented with a formulation in an Updated Lagrange framework. Hence, Rezoning can be used for elastomeric materials based upon the current configuration. This is specially useful in large deformation analysis since typically excessive element distortion in elastomeric materials can lead to premature termination of analysis. The new formulation accommodates the generalized Mooney-Rivlin and Ogden material models preserving the same format and strain energy functions as the total Lagrange formulation. In addition, the Arruda-Boyce and the Gent models are only available in the updated Lagrange framework.

The updated Lagrangian rubber elasticity capability can be used in conjunction with both continuous as well as discontinuous damage models. Thermal, as well as viscoelastic, effects can be modeled with the current formulation. While the Mooney model can account for the temperature dependent material properties, the Ogden model does not support the temperature dependence at this time. The singularity ratio of the system is inversely proportional to the order of bulk modulus of the material due to the condensation procedure.

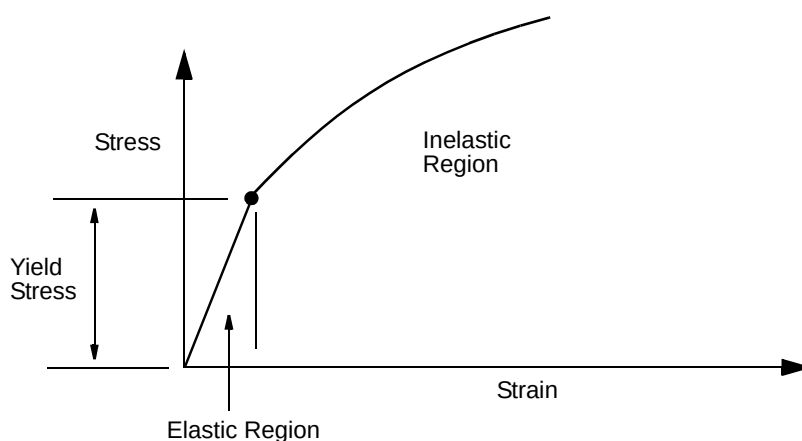
A consistent linearization has been carried out to obtain the tangent modulus. The singularity for the case of two- or three-equal stretch ratios is analytically removed by application of L'Hospital's rule. The current framework with an exact implementation of the finite strain kinematics along with the split of strain energy to handle compressible and nearly incompressible response is eminently suitable for implementation of any nonlinear elastic as well as inelastic material models. In fact, the finite deformation plasticity model based on the multiplicative decomposition,

$F = F^e F^{\theta} F^p$ is implemented in the same framework.

To simulate elastomeric materials, incompressible element(s) are used for plane strain, axisymmetric, and three-dimensional problems for elasticity in total Lagrangian framework. These elements can be used with each other or in combination with other elements in the library. For plane stress, beam, plate or shell analysis, conventional elements can be used. For updated Lagrangian elasticity, both conventional elements (as well as Herrmann elements) can be used for plane strain, axisymmetric, and three-dimensional problems.

Time-independent Inelastic Behavior

In uniaxial tension tests of most metals (and many other materials), the following phenomena can be observed. If the stress in the specimen is below the yield stress of the material, the material behaves elastically and the stress in the specimen is proportional to the strain. If the stress in the specimen is greater than the yield stress, the material no longer exhibits elastic behavior, and the stress-strain relationship becomes nonlinear. [Figure 7-21](#) shows a typical uniaxial stress-strain curve. Both the elastic and inelastic regions are indicated.



Note: Stress and strain are total quantities.

Figure 7-21 Typical Uniaxial Stress-Strain Curve (Uniaxial Test)

Within the elastic region, the stress-strain relationship is unique. As illustrated in [Figure 7-22](#), if the stress in the specimen is increased (loading) from zero (point 0) to σ_1 (point 1), and then decreased (unloading) to zero, the strain in the specimen is also increased from zero to ϵ_1 , and then returned to zero. The elastic strain is completely recovered upon the release of stress in the specimen.

The loading-unloading situation in the inelastic region is different from the elastic behavior. If the specimen is loaded beyond yield to point 2, where the stress in the specimen is σ_2 and the total strain is ϵ_2 , upon release of the stress in the specimen the elastic strain, ϵ_2^e , is completely recovered. However, the inelastic (plastic) strain, ϵ_2^p , remains in the specimen. [Figure 7-22](#) illustrates this relationship. Similarly, if the

specimen is loaded to point 3 and then unloaded to zero stress state, the plastic strain ϵ_3^p remains in the specimen. It is obvious that ϵ_2^p is not equal to ϵ_3^p . We can conclude that in the inelastic region:

- Plastic strain permanently remains in the specimen upon removal of stress.
- The amount of plastic strain remaining in the specimen is dependent upon the stress level at which the unloading starts (path-dependent behavior).

The uniaxial stress-strain curve is usually plotted for total quantities (total stress versus total strain). The total stress-strain curve shown in [Figure 7-21](#) can be replotted as a total stress versus plastic strain curve, as shown in [Figure 7-23](#). The slope of the total stress versus plastic strain curve is defined as the workhardening slope (H) of the material. The workhardening slope is a function of plastic strain.

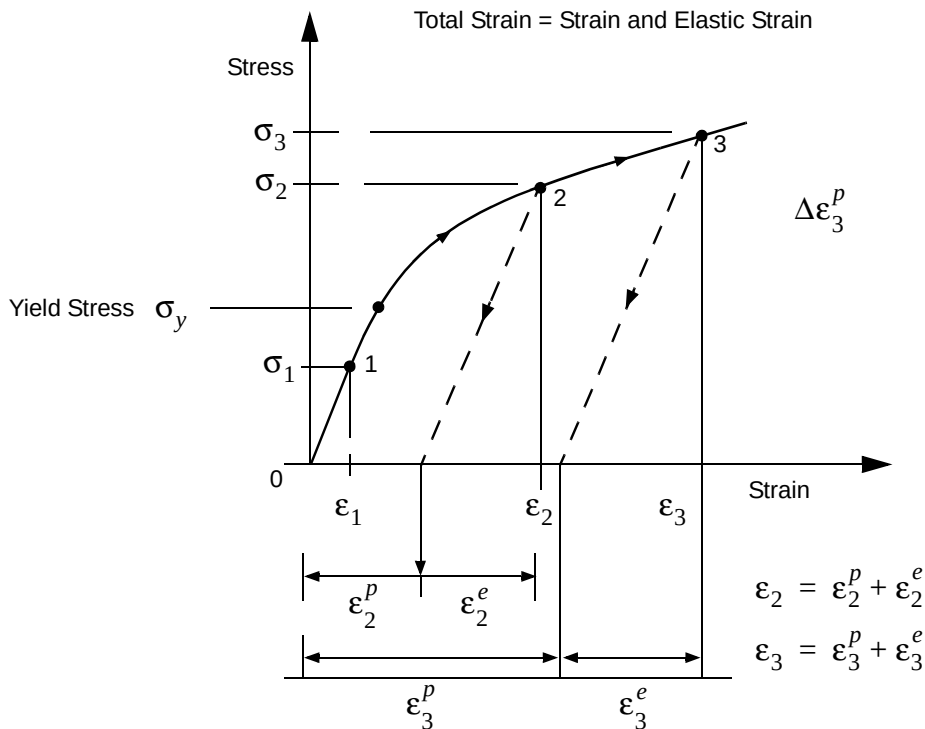


Figure 7-22 Schematic of Simple Loading - Unloading (Uniaxial Test)

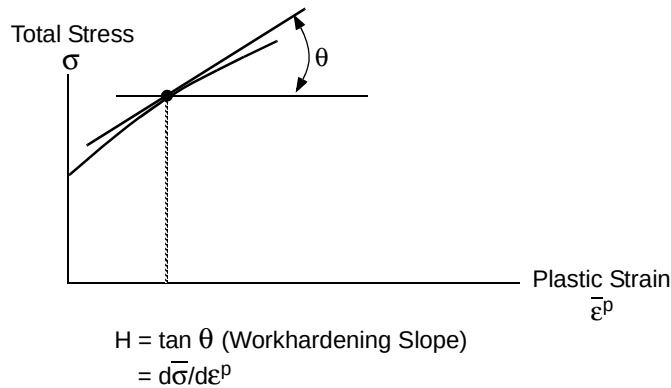


Figure 7-23 Definition of Workhardening Slope (Uniaxial Test)

The stress-strain curve shown in [Figure 7-21](#) is directly plotted from experimental data. It can be simplified for the purpose of numerical modeling. A few simplifications are shown in [Figure 7-24](#) and are listed below:

1. Bilinear representation – constant workhardening slope
2. Elastic perfectly-plastic material – no workhardening
3. Perfectly-plastic material – no workhardening and no elastic response
4. Piecewise linear representation – multiple constant workhardening slopes
5. Strain-softening material – negative workhardening slope

In addition to elastic material constants (Young's modulus and Poisson's ratio), it is essential to include yield stress and workhardening slopes when dealing with inelastic (plastic) material behavior. These quantities can vary with parameters such as temperature and strain rate. Since the yield stress is generally measured from uniaxial tests, and the stresses in real structures are usually multiaxial, the yield condition of a multiaxial stress state must be considered. The conditions of subsequent yield (workhardening rules) must also be studied.

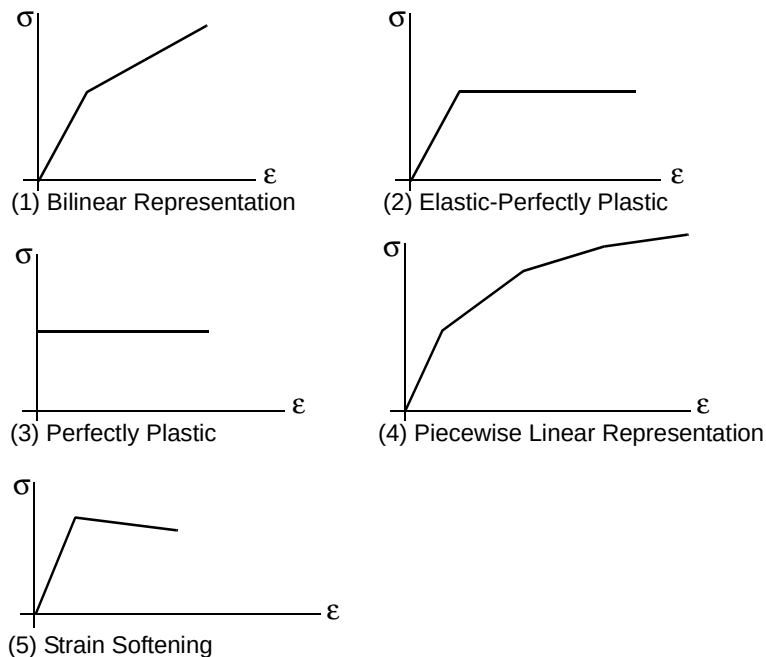


Figure 7-24 Simplified Stress-Strain Curves (Uniaxial Test)

Yield Conditions

The yield stress of a material is a measured stress level that separates the elastic and inelastic behavior of the material. The magnitude of the yield stress is generally obtained from a uniaxial test. However, the stresses in a structure are usually multiaxial. A measurement of yielding for the multiaxial state of stress is called the yield condition. Depending on how the multiaxial state of stress is represented, there can be many forms of yield conditions. For example, the yield condition can be dependent on all stress components, on shear components only, or on hydrostatic stress. A number of yield conditions are available in MSC.Marc, and are discussed in this section.

von Mises Yield Condition

Although many forms of yield conditions are available, the von Mises criterion is the most widely used. The success of the von Mises criterion is due to the continuous nature of the function that defines this criterion and its agreement with observed behavior for the commonly encountered ductile materials. The von Mises criterion states that yield occurs when the effective (or equivalent) stress (σ) equals the yield stress (σ_y) as measured in a uniaxial test. [Figure 7-25](#) shows the von Mises yield surface in two-dimensional and three-dimensional stress space.

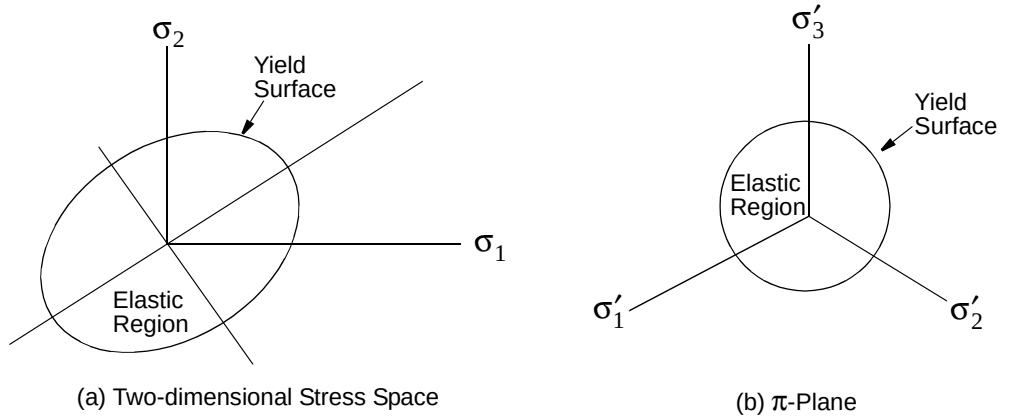


Figure 7-25 von Mises Yield Surface

For an isotropic material:

$$\bar{\sigma} = [(\sigma_1 - \sigma_2)^2 + (\sigma_2 - \sigma_3)^2 + (\sigma_3 - \sigma_1)^2]^{1/2} / \sqrt{2} \quad (7-96)$$

where σ_1 , σ_2 , and σ_3 are the principal Cauchy stresses.

σ can also be expressed in terms of nonprincipal Cauchy stresses.

$$(\bar{\sigma} = [(\sigma_x - \sigma_y)^2 + (\sigma_y - \sigma_z)^2 + (\sigma_z - \sigma_x)^2 + 6(\tau_{xy}^2 + \tau_{yz}^2 + \tau_{zx}^2)]^{1/2} / \sqrt{2} \quad (7-97)$$

The yield condition can also be expressed in terms of the deviatoric stresses as:

$$\bar{\sigma} = \sqrt{\frac{3}{2} \sigma'_{ij} \sigma'_{ij}} \quad (7-98)$$

where σ'_{ij} is the deviatoric Cauchy stress expressed as

$$\sigma'_{ij} = \sigma_{ij} - \frac{1}{3} \sigma_{kk} \delta_{ij} \quad (7-99)$$

For isotropic material, the von Mises yield condition is the default condition in MSC.Marc. The initial yield stress σ_y is defined in the [ISOTROPIC](#) and [ORTHOTROPIC](#) options. A user-defined finite strain, isotropic plasticity material model can be implemented through the user subroutine [UFINITE](#). In this case, the finite strain kinematics is taken care of in MSC.Marc. You have to do the small strain return mapping only. See [MSC.Marc Volume D: User Subroutines and Special Routines](#) for more details.

Hill's [1948] Yield Function

The anisotropic yield function (of Hill) and stress potential are assumed as

$$\bar{\sigma} = [a_1(\sigma_y - \sigma_z)^2 + a_2(\sigma_z - \sigma_x)^2 + a_3(\sigma_x - \sigma_y)^2 + 3a_4\tau_{zx}^2 + 3a_5\tau_{yz}^2 + 3a_6\tau_{xy}^2]^{1/2} / \sqrt{2} \quad (7-100)$$

where $\bar{\sigma}$ is the equivalent tensile yield stress for isotropic behavior.

Ratios of actual to isotropic yield (in the preferred orientation) are defined in the array YRDIR for direct tension yielding, and in YRSHR for yield in a shear (the ratio of actual shear yield to $\sigma/\sqrt{3}$ isotropic shear yield). Then the a_1 above are defined by:

$$a_1 = \frac{1}{YRDIR(2)**2} + \frac{1}{YRDIR(3)**2} - \frac{1}{YRDIR(1)**2} \quad (7-101)$$

$$a_2 = \frac{1}{YRDIR(3)**2} + \frac{1}{YRDIR(1)**2} - \frac{1}{YRDIR(2)**2} \quad (7-102)$$

$$a_3 = \frac{1}{YRDIR(1)**2} + \frac{1}{YRDIR(2)**2} - \frac{1}{YRDIR(3)**2} \quad (7-103)$$

$$a_4 = \frac{2}{YRSHR(3)**2} \quad (7-104)$$

$$a_5 = \frac{2}{YRSHR(2)**2} \quad (7-105)$$

$$a_6 = \frac{2}{YRSHR(1)**2} \quad (7-106)$$

For anisotropic material, use model definition options **ISOTROPIC**, **ORTHOTROPIC** or **ANISOTROPIC** to indicate the anisotropy. Use the **ORTHOTROPIC** option or the user subroutine **ANPLAS** for the specification of anisotropic yield condition (constants a_1 through a_6 , as defined above), and the model definition option **ORIENTATION** or, the user subroutine **ORIENT**, if necessary to specify preferred orientations.

Hill's (1948) Yield Criterion has been extensively used in sheet metal forming, especially for steel. The experimental data can be related to the MSC.Marc input for the Hill's Yield Criterion for the shells or plane stress case as given below.

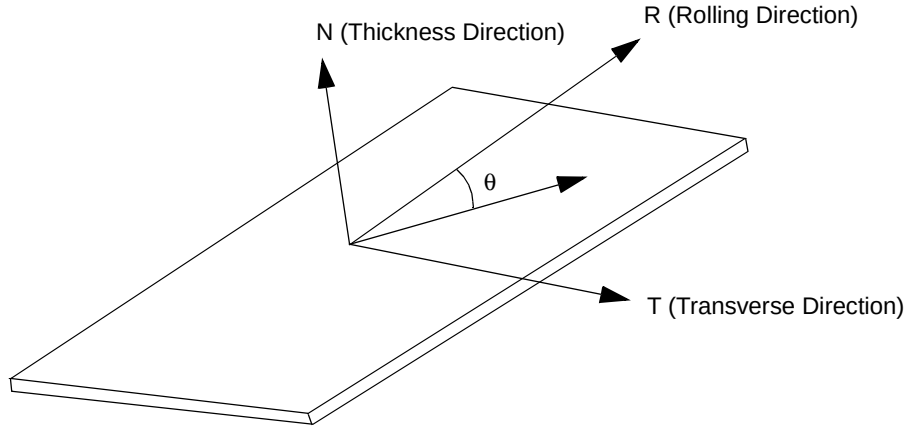


Figure 7-26 Axes of Anisotropy

The sample of tensile coupon cut from a sheet in the three directions, $\theta = 0$ (rolling), 45° and 90° (transverse) is tested to obtain $\sigma = \sigma_0$, σ_{45} , and σ_{90} , respectively. Similarly, the anisotropy parameter defined as:

$$r = \frac{\epsilon_{\text{width}}}{\epsilon_{\text{thickness}}} \quad (7-107)$$

is obtained for the 0° , 45° , and 90° directions. The yield stress in the third (thickness) direction can be written as:

$$\sigma_N = \sigma_0 \sqrt{\frac{r_{90}(1 + r_0)}{r_0 + r_{90}}} = \sigma_{90} \sqrt{\frac{r_0(1 + r_{90})}{r_0 + r_{90}}} \quad (7-108)$$

The direct stress coefficients are now:

$$YRDIR(1) = \frac{\sigma_0}{\sigma_{av}} \quad (7-109)$$

$$YRDIR(2) = \frac{\sigma_{90}}{\sigma_{av}} \quad (7-110)$$

$$YRDIR(3) = \frac{\sigma_N}{\sigma_{av}} \quad (7-111)$$

where σ_{av} is the initial yield stress on the stress-strain curve used. If the stress-strain curve is averaged from all directions, σ_{av} is defined as $\sigma_{av} = \frac{\sigma_0 + 2\sigma_{45} + \sigma_{90}}{4}$ in orthotropic plasticity. Similarly, the shear coefficients can be evaluated at:

$$YRSHR(1) = YRDIR(3) \sqrt{\frac{3}{2r_{45} + 1}} \quad (7-112)$$

$$YRSHR(2) = YRSHR(3) = 1.0 \quad (7-113)$$

It is noted that the transverse direction along the thickness is usually considered isotropic which is reasonable on physical grounds. Also, notice that for complete isotropy, $\sigma_0 = \sigma_{45} = \sigma_{90}$ and $r_0 = r_{45} = r_{90} = 1$, which yields the von Mises yield criterion.

Barlat's (1991) Yield Function

Barlat et al. [Ref. 6] proposed a general criterion for planar anisotropy that is particularly suitable for aluminum alloy sheets. This criterion has been shown to be consistent with polycrystal-based yield surfaces which often exhibit small radii of curvature near uniaxial and balanced biaxial tension stress states. An advantage of this criterion is that its formulation is relatively simple as compared with the formulation for polycrystalline modeling and, therefore, it can be easily incorporated into finite element codes for the analysis of metal forming problems.

For three dimensional deformation, the yield function f is defined as (Barlat et al. [Ref. 6])

$$f = |S_1 - S_2|^m + |S_2 - S_3|^m + |S_3 - S_1|^m = 2\bar{\sigma}^m \quad (7-114)$$

where $S_i = 1, 2, 3$ are principal values of a symmetric matrix $S_{\alpha\beta}$ defined with respect to the components of the Cauchy stress as

$$S = \begin{bmatrix} \frac{C_3(\sigma_{xx} - \sigma_{yy}) - C_2(\sigma_{zz} - \sigma_{xx})}{3} & C_6\sigma_{xy} & C_5\sigma_{zx} \\ C_6\sigma_{xy} & \frac{C_1(\sigma_{yy} - \sigma_{zz}) - C_3(\sigma_{xx} - \sigma_{yy})}{3} & C_4\sigma_{zy} \\ C_5\sigma_{zx} & C_4\sigma_{zy} & \frac{C_2(\sigma_{zz} - \sigma_{xx}) - C_1(\sigma_{yy} - \sigma_{zz})}{3} \end{bmatrix} \quad (7-115)$$

In Equation 7-115, the symmetry axes (x, y, z), which represent the mutually orthogonal axes of anisotropy, are aligned with the initial rolling, transverse, and normal directions of the sheet. During deformation, the anisotropic yield surface of each material element rotates so that the symmetry axes are all in different directions during deformation. Therefore, it is necessary to trace the rotation of the yield surface during deformation in order to calculate the plastic strain increment properly. The rotation of the anisotropy axes is carried out based on the polar decomposition method.

The material coefficients, $C_{i=1 \sim 6}$ in Equation 7-115 represent anisotropic properties. When $C_{i=1 \sim 6} = 1$, the material is isotropic and Equation 7-114 reduces to the Tresca yield condition for $m = 1$ or ∞ , and the von Mises yield criterion for $m = 2$ or 4 . The exponent “ m ” is mainly associated with the crystal structure of the material. A higher “ m ” value has the effect of decreasing the radius of curvature of rounded vertices near the uniaxial and balanced biaxial tension ranges of the yield surface, in agreement with polycrystal models. Values of $m = 8$ for FCC materials (like aluminum) and $m = 6$ for BCC materials (like steel) are recommended. The yield surface has been proven to be convex for $m \geq 1$. Figure 7-27 shows the yield surfaces obtained from von-Mises, Hill and Barlat yield functions for Aluminum 2008-T4 alloy.

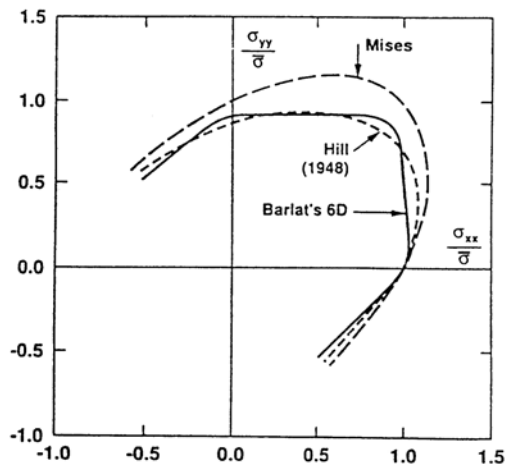


Figure 7-27 Comparison of yield surfaces obtained from von-Mises, Hill and Barlat yield functions

Utilizing the normality rule, the associated plastic strain increment $\Delta \varepsilon_{\alpha\beta}^p$ is obtained from the yield function f as

$$\Delta \varepsilon_{\alpha\beta}^p = \lambda \frac{\partial f}{\partial \sigma_{\alpha\beta}} \quad (7-116)$$

where λ is a scalar function. The calculation of $\frac{\partial f}{\partial \sigma_{\alpha\beta}}$ in Equation 7-116 is lengthy but straightforward. The stress integration to obtain $\Delta \varepsilon_{\alpha\beta}^p$ along with the calculation of $\frac{\partial f}{\partial \sigma_{\alpha\beta}}$ are shown in the works by Chung and Shah [Ref. 7] and Yoon et al. [Ref. 8].

In order to obtain four unknown independent coefficients (C_1, C_2, C_3, C_6) with the assumption of $C_4 = C_5 = 1$ (isotropic properties for transverse directions), it is necessary to use four test results ($\sigma_0, \sigma_{45}, \sigma_{90}, \sigma_b$), where $\sigma_0, \sigma_{45}, \sigma_{90}$ are the tensile yield stresses at $0^\circ, 45^\circ$, and 90° from the rolling direction, and σ_b is the balanced biaxial yield stress measured from bulge test. A detailed procedure to calculate the coefficients of Barlat's yield function are summarized in the work of Yoon et al. [Ref. 9]. In MSC.Marc, Barlat's coefficients are automatically calculated from user inputs for $\sigma_0, \sigma_{45}, \sigma_{90}, \sigma_b$ in MSC.Marc Mentat. If biaxial data is not available, generally MSC.Marc assumes $\sigma_b = \sigma_0$. Barlat's yield function can be accessed from **ISOTROPIC**, **ORTHOTROPIC** or **ANISOTROPIC** model definition options and can be used in conjunction with the **ORIENTATION** option.

Mohr-Coulomb Material (Hydrostatic Stress Dependence)

MSC.Marc includes options for elastic-plastic behavior based on a yield surface that exhibits hydrostatic stress dependence. Such behavior is observed in a wide class of soil and rock-like materials. These materials are generally classified as Mohr-Coulomb materials (generalized von Mises materials). Ice is also thought to be a Mohr-Coulomb material. The generalized Mohr-Coulomb model developed by Drucker and Prager is implemented in MSC.Marc. There are two types of Mohr-Coulomb materials: linear and parabolic. Each is discussed on the following pages.

Linear Mohr-Coulomb Material

The deviatoric yield function, as shown in Figure 7-28, is assumed to be a linear function of the hydrostatic stress.

$$f = \alpha J_1 + J_2^{1/2} - \frac{\bar{\sigma}}{\sqrt{3}} = 0 \quad (7-117)$$

where

$$J_1 = \sigma_{ii} \quad (7-118)$$

$$J_2 = \frac{1}{2} \sigma'_{ij} \sigma'_{ij} \quad (7-119)$$

Analysis of linear Mohr-Coulomb material based on the constitutive description above is available in MSC.Marc through the **ISOTROPIC** model definition option. Through the **ISOTROPIC** option, the values of σ and α are entered. Note that, throughout the program, the convention that the tensile direct stress is positive is maintained, contrary to its use in many soil mechanics texts.

The constants α and $\bar{\sigma}$ can be related to c and ϕ by

$$c = \frac{\bar{\sigma}}{[3(1 - 12\alpha^2)]^{1/2}} ; \quad \frac{3\alpha}{(1 - 3\alpha^2)^{1/2}} = \sin \phi \quad (7-120)$$

where c is the cohesion and ϕ is the angle of friction.

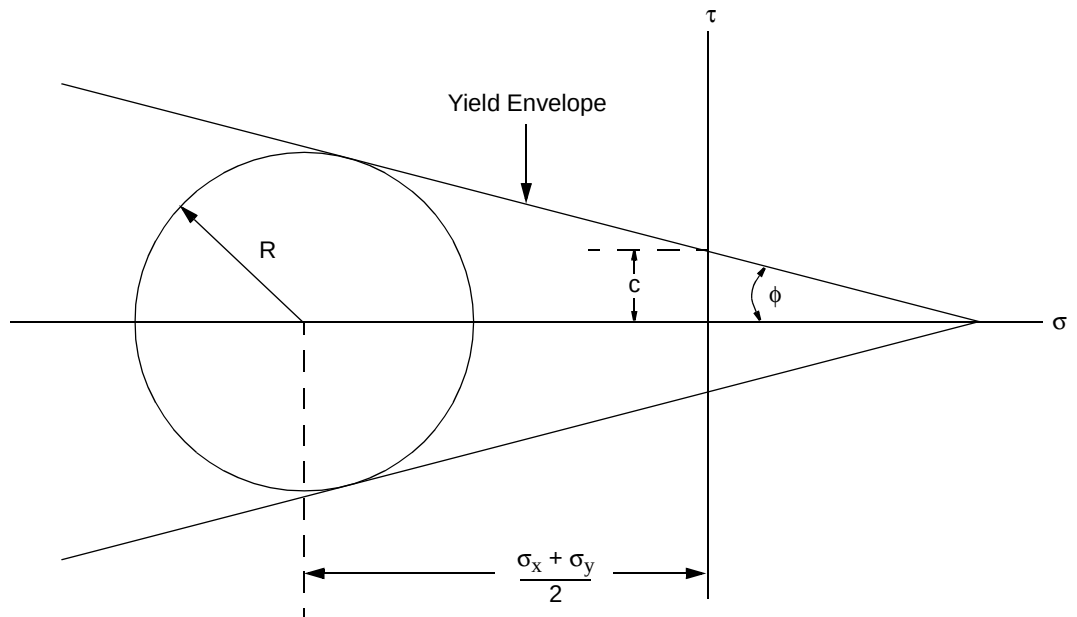


Figure 7-28 Yield Envelope of Plane Strain (Linear Mohr-Coulomb Material)

Parabolic Mohr-Coulomb Material

The hydrostatic dependence is generalized to give a yield envelope which is parabolic in the case of plane strain (see Figure 7-29).

$$f = (3J_2 + \sqrt{3}\beta\bar{\sigma}J_1)^{1/2} - \bar{\sigma} = 0 \quad (7-121)$$

The parabolic yield surface is obtained in MSC.Marc through the **ISOTROPIC** model definition option. Enter the values σ and β through the model definition option **ISOTROPIC**.

$$\bar{\sigma}^2 = 3\left(c^2 - \frac{\alpha^2}{3}\right) \quad \beta = \frac{\alpha}{(3(3c^2 - \alpha^2))^{1/2}} \quad (7-122)$$

where c is the cohesion.

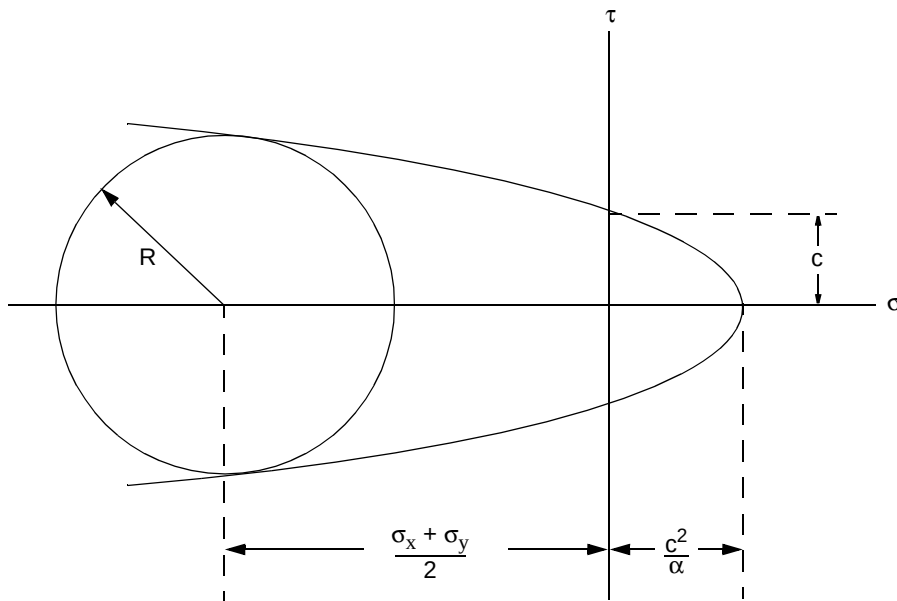


Figure 7-29 Resultant Yield Condition of Plane Strain (Parabolic Mohr-Coulomb Material)

Buyukozturk Criterion (Hydrostatic Stress Dependence)

This yield criterion [Ref. 4], which originally has been proposed as a failure criterion, has the general form:

$$f = \beta \sqrt{3} \bar{\sigma} J_1 + \gamma J_1^2 + 3J_2 - \bar{\sigma}^2 \quad (7-123)$$

Through the **ISOTROPIC** model definition option, the user has to define $\bar{\sigma}$ and the factor β , where γ has a fixed value of 0.2. The Buyukozturk criterion reduces to the parabolic Mohr-Coulomb criterion if $\gamma = 0$.

Powder Material

Some materials, during certain stages of manufacturing are granular in nature. In particular, powder metals are often used in certain forging operations and during hot isostatic pressing (HIP). These material properties are functions of both the temperature and the densification. It should be noted that the soil model discussed in this chapter also exhibits some of these characteristics. In the model incorporated into MSC.Marc, a unified viscoplastic approach is used. The yield function is

$$F = \frac{1}{\gamma} \left(\frac{3}{2} \sigma' \sigma' + \frac{p^2}{\beta^2} \right)^{1/2} - \sigma_y \quad (7-124)$$

where σ_y is the uniaxial yield stress, σ' is the deviatoric stress tensor, and p is the hydrostatic pressure. γ , β are material parameters.

σ_y can be a function of temperature and relative density, γ , β are functions only of relative density.

Typically, we allow:

$$\begin{aligned} \beta &= (q_1 + q_2 \rho^{q_3})^{q_4} \\ \gamma &= (b_1 + b_2 \rho^{b_3})^{b_4} \end{aligned} \quad (7-125)$$

where ρ is the relative density.

As the powder becomes more dense, ρ approaches 1 and the classical von Mises model is recovered.

It should be noted that the elastic properties are also functions of relative density. In particular, as the material becomes fully dense, the Poisson's ratio approaches 0.5.

As most processes involving powder materials are both pressure and thermally driven, it can be necessary to perform a coupled analysis. MSC.Marc also allows you to specify density effects for the thermal properties, conductivity and specific heat. The

basic input data is entered through the **POWDER** option. In addition to the **TEMPERATURE EFFECTS** option, there is a **DENSITY EFFECTS** option. The initial relative density is entered through the **RELATIVE DENSITY** option.

Note that this material can undergo both large shear and volumetric strains. The **LARGE DISP**, **UPDATE**, and **FINITE** parameters should also be used. In HIP processes, the can would typically be modeled using a conventional elastic plastic material law. Additionally, the **FOLLOW FOR** parameter would be used to ensure that the pressure remain normal to the deformed material.

The necessary experiments required for determination of the constants for the Shima Model are as follows:

1. To determine the parameter p , a uniaxial compression test is performed:

$$\sigma_{11} = p, \sigma_{22} = \sigma_{33}, \sigma_{ij} = 0 \text{ as } i \neq j \quad (7-126)$$

By measuring the incremental strain during a period of time, β can be obtained

$$\beta = \left(\frac{2}{3} \frac{d\epsilon_{11}^p - d\epsilon_{22}^p}{d\epsilon_{ii}^p} \right)^{+0.5} \quad (7-127)$$

2. The second parameter, γ , can be obtained by performing a triaxial (uniform) compression test:

$$\sigma_{11} = \sigma_{22} = \sigma_{33} \rightarrow \sigma_{eq} = \frac{p}{\gamma\beta} \quad (7-128)$$

By neglecting the elastic strains, you obtain:

$$d\epsilon_{ii} = 0 \rightarrow \sigma_y = \frac{p}{\gamma\beta} \quad (7-129)$$

or

$$\gamma = \frac{p}{\sigma_y\beta} \quad (7-130)$$

For thermo-mechanically coupled problems, the temperature dependent functions are obtained by repeating the above test procedures at different temperatures.

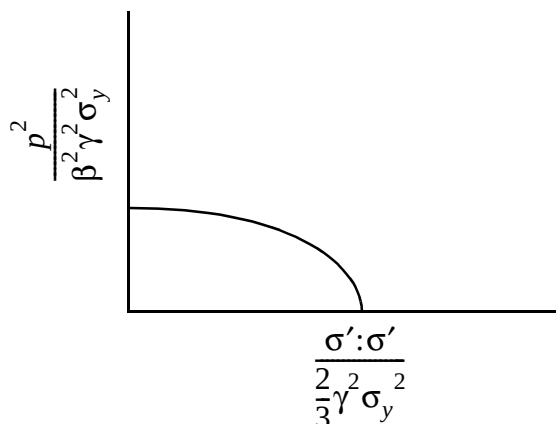


Figure 7-30 Yield Function of Shima Model

Oak Ridge National Laboratory Options

In MSC.Marc, the ORNL options are based on the definitions of ORNL-TM- 3602 [Ref. 10] for stainless steels and ORNL recommendations [Ref. 16] for 2 1/4 Cr-1 Mo steel.

The initial yield stress should be used for the initial inelastic loading calculations for both the stainless steels and 2 1/4 Cr-1 Mo steel. The 10th-cycle yield stress should be used for the hardened material. The 100th-cycle yield stress must be used in the following circumstances:

1. To accommodate cyclic softening of 2 1/4 Cr-1 Mo steel after many load cycles
2. After a long period of high temperature exposure
3. After the occurrence of creep strain

To enter initial and 10th-cycle yield stresses, use the model definition option **ISOTROPIC** or **ORTHOTROPIC**.

Effects of Temperature and Strain Rates on Yield Stress

This section describes MSC.Marc capabilities with respect to the effect of temperature and strain rate.

MSC.Marc allows you to input a temperature-dependent yield stress. To enter the yield stress at a reference temperature, use the model definition options **ISOTROPIC** or **ORTHOTROPIC**. To enter variations of yield stress with temperatures, use the model definition options **TEMPERATURE EFFECTS** and **ORTHO TEMP**. Repeat the model

definition options [TEMPERATURE EFFECTS](#) and [ORTHO TEMP](#) for each material, as necessary. The effect of temperatures on yielding is discussed further in [Constitutive Relations](#).

MSC.Marc allows you to enter a strain rate dependent yield stress, for use in dynamic and flow (for example, extrusion) problems. To use the strain rate dependent yield stress in static analysis, enter a fictitious time using the [TIME STEP](#) option. The zero-strain-rate yield stress is given on the [ISOTROPIC](#) or [ORTHOTROPIC](#) options. Repeat the model definition option [STRAIN RATE](#) for each different material where strain rate data are necessary. Refer to [Constitutive Relations](#) for more information on the strain-rate effect on yielding.

Workhardening Rules

In a uniaxial test, the workhardening slope is defined as the slope of the stress-plastic strain curve. The workhardening slope relates the incremental stress to incremental plastic strain in the inelastic region and dictates the conditions of subsequent yielding. A number of workhardening rules (isotropic, kinematic, and combined) are available in MSC.Marc. A description of these workhardening rules is given below. The uniaxial stress-plastic strain curve can be represented by a piecewise linear function through the [WORK HARD](#) option. As an alternative, you can specify workhardening through the user subroutine [WKSLP](#).

There are two methods to enter this information, using the [WORK HARD](#) option. In the first method, you must enter workhardening slopes for uniaxial stress data as a change in stress per unit of plastic strain (see [Figure 7-31](#)) and the plastic strain at which these slopes become effective (breakpoint).

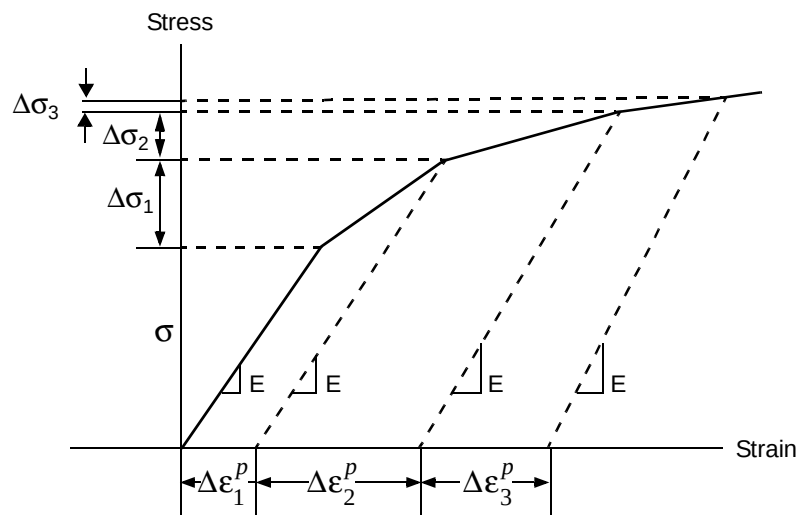


Figure 7-31 Workhardening Slopes

<u>Slope</u>	<u>Breakpoint</u>
$\frac{\Delta\sigma_1}{\Delta\epsilon_1^p}$	0.0
$\frac{\Delta\sigma_2}{\Delta\epsilon_2^p}$	$\Delta\epsilon_1^p$
$\frac{\Delta\sigma_3}{\Delta\epsilon_3^p}$	$\Delta\epsilon_1^p + \Delta\epsilon_2^p$

and so on.

Note: The slopes of the workhardening curves should be based on a plot of the stress versus plastic strain for a tensile test. The elastic strain components of the stress-strain curve should not be included. The first breakpoint of the workhardening curve should be 0.0.

In the second method, you enter a of yield stress, plastic strain points. This option is flagged by adding the word DATA to the work hard statement.

Note: The data points should be based on a plot of the stress versus plastic strain for a tensile test. The elastic strain components should not be included. The first plastic strain should equal 0.0 and the first stress should agree with that given as the yield stress in the ISOTROPIC or ORTHOTROPIC options.

The yield stress and the workhardening data must be compatible with the procedure used in the analysis. For small strain analyses, the engineering stress and engineering strain are appropriate. If only the LARGE DISP parameter is used, the yield stress

should be entered as the second Piola-Kirchhoff stress, and the workhard data be given with respect to plastic Green-Lagrange strains. If the [LARGE DISP](#), [UPDATE](#), [FINITE](#) parameters or [PLASTICITY](#) 3, 4, or 5 are used, the yield stress must be defined as a true or Cauchy stress, and the workhardening data with respect to logarithmic plastic strains.

Isotropic Hardening

The isotropic workhardening rule assumes that the center of the yield surface remains stationary in the stress space, but that the size (radius) of the yield surface expands, due to workhardening. The change of the von Mises yield surface is plotted in [Figure 7-32b](#).

A review of the load path of a uniaxial test that involves both the loading and unloading of a specimen will assist in describing the isotropic workhardening rule. The specimen is first loaded from stress free (point 0) to initial yield at point 1, as shown in [Figure 7-32a](#). It is then continuously loaded to point 2. Then, unloading from 2 to 3 following the elastic slope E (Young's modulus) and then elastic reloading from 3 to 2 takes place. Finally, the specimen is plastically loaded again from 2 to 4 and elastically unloaded from 4 to 5. Reverse plastic loading occurs between 5 and 6.

It is obvious that the stress at 1 is equal to the initial yield stress σ_y and stresses at points 2 and 4 are larger than σ_y , due to workhardening. During unloading, the stress state can remain elastic (for example, point 3), or it can reach a subsequent (reversed) yield point (for example, point 5). The isotropic workhardening rule states that the reverse yield occurs at current stress level in the reversed direction. Let σ_4 be the stress level at point 4. Then, the reverse yield can only take place at a stress level of $-\sigma_4$ (point 5).

The isotropic workhardening model (with a work slope of 0) is the default option in MSC.Marc. To explicitly specify the isotropic hardening option in MSC.Marc, use the model definition options [ISOTROPIC](#) or [ORTHOTROPIC](#). To input workhardening slope data, use the [WORK HARD](#) option or [WKSPLP](#) user subroutine.

For many materials, the isotropic workhardening model is inaccurate if unloading occurs (as in cyclic loading problems). For these problems, the kinematic hardening model or the combined hardening model represents the material better.

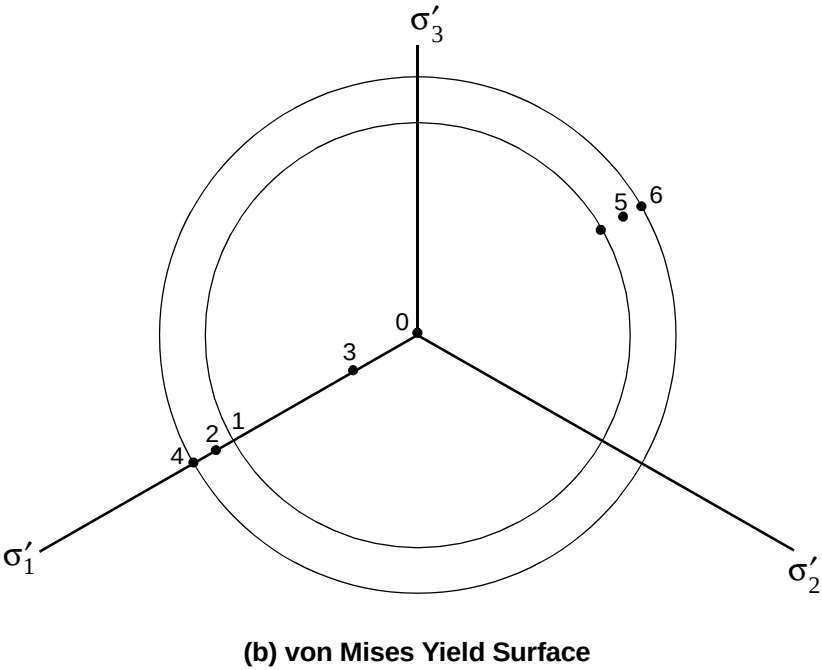
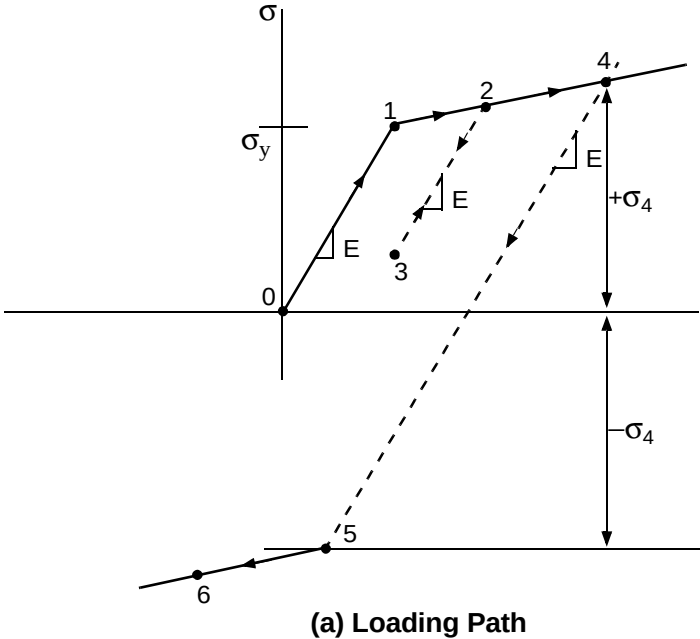


Figure 7-32 Schematic of Isotropic Hardening Rule (Uniaxial Test)

Kinematic Hardening

Under the kinematic hardening rule, the von Mises yield surface does not change in size or shape, but the center of the yield surface can move in stress space. [Figure 7-33b](#) illustrates this condition. Ziegler's law is used to define the translation of the yield surface in the stress space.

The loading path of a uniaxial test is shown in [Figure 7-33a](#). The specimen is loaded in the following order: from stress free (point 0) to initial yield (point 1), 2 (loading), 3 (unloading), 2 (reloading), 4 (loading), 5 and 6 (unloading). As in isotropic hardening, stress at 1 is equal to the initial yield stress σ_y , and stresses at 2 and 4 are higher than σ_y , due to workhardening. Point 3 is elastic, and reverse yield takes place at point 5. Under the kinematic hardening rule, the reverse yield occurs at the level of $\sigma_5 = (\sigma_4 - 2\sigma_y)$, rather than at the stress level of $-\sigma_4$. Similarly, if the specimen is loaded to a higher stress level σ_7 (point 7), and then unloaded to the subsequent yield point 8, the stress at point 8 is $\sigma_8 = (\sigma_7 - 2\sigma_y)$. If the specimen is unloaded from a (tensile) stress state (such as point 4 and 7), the reverse yield can occur at a stress state in either the reverse (point 5) or the same (point 8) direction.

To invoke the kinematic hardening in MSC.Marc, use the model definition options [ISOTROPIC](#) or [ORTHOTROPIC](#). To input workhardening slope data, use the [WORK HARD](#) option or [WKSLP](#) user subroutine.

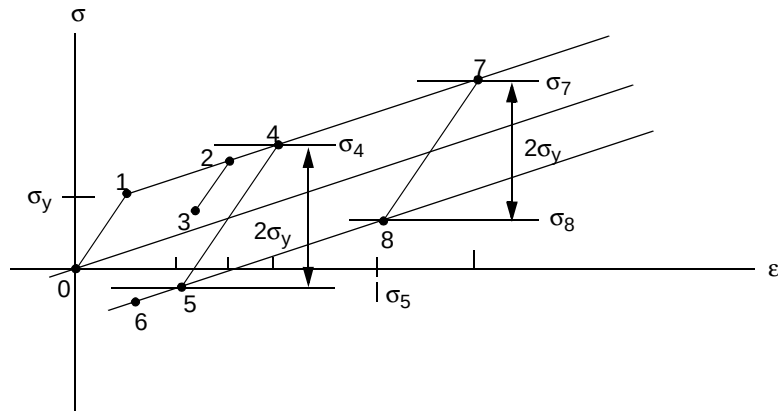
For many materials, the kinematic hardening model gives a better representation of loading/unloading behavior than the isotropic hardening model. For cyclic loading, however, the kinematic hardening model can represent neither cyclic hardening nor cyclic softening.

Combined Hardening

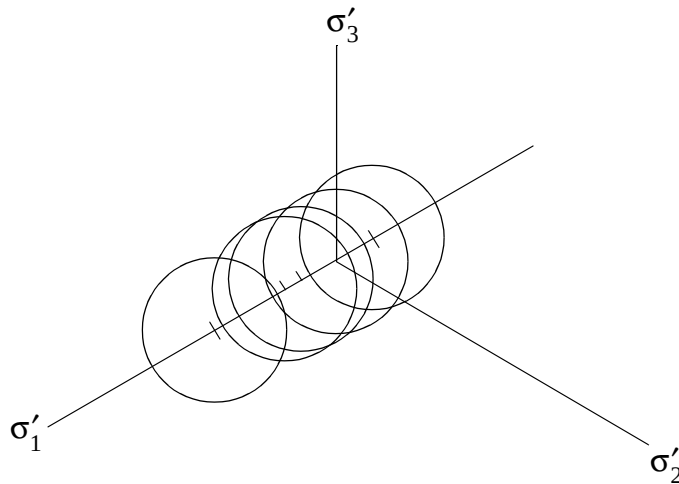
[Figure 7-34](#) shows a material with highly nonlinear hardening. Here, the initial hardening is assumed to be almost entirely isotropic, but after some plastic straining, the elastic range attains an essentially constant value (that is, pure kinematic hardening). The basic assumption of the combined hardening model is that such behavior is reasonably approximated by a classical constant kinematic hardening constraint, with the superposition of initial isotropic hardening. The isotropic hardening rate eventually decays to zero as a function of the equivalent plastic strain measured by

$$\bar{\epsilon}^p = \int \dot{\epsilon}^p dt = \int \left(\frac{2}{3} \dot{\epsilon}_{ij}^p \dot{\epsilon}_{ij}^p \right)^{1/2} dt \quad (7-131)$$

This implies a constant shift of the center of the elastic domain, with a growth of elastic domain around this center until pure kinematic hardening is attained. In this model, there is a variable proportion between the isotropic and kinematic contributions that depends on the extent of plastic deformation (as measured by $\bar{\epsilon}^p$).



(a) Loading Path



(b) von Mises Yield Surface

Figure 7-33 Schematic of Kinematic Hardening Rule (Uniaxial Test)

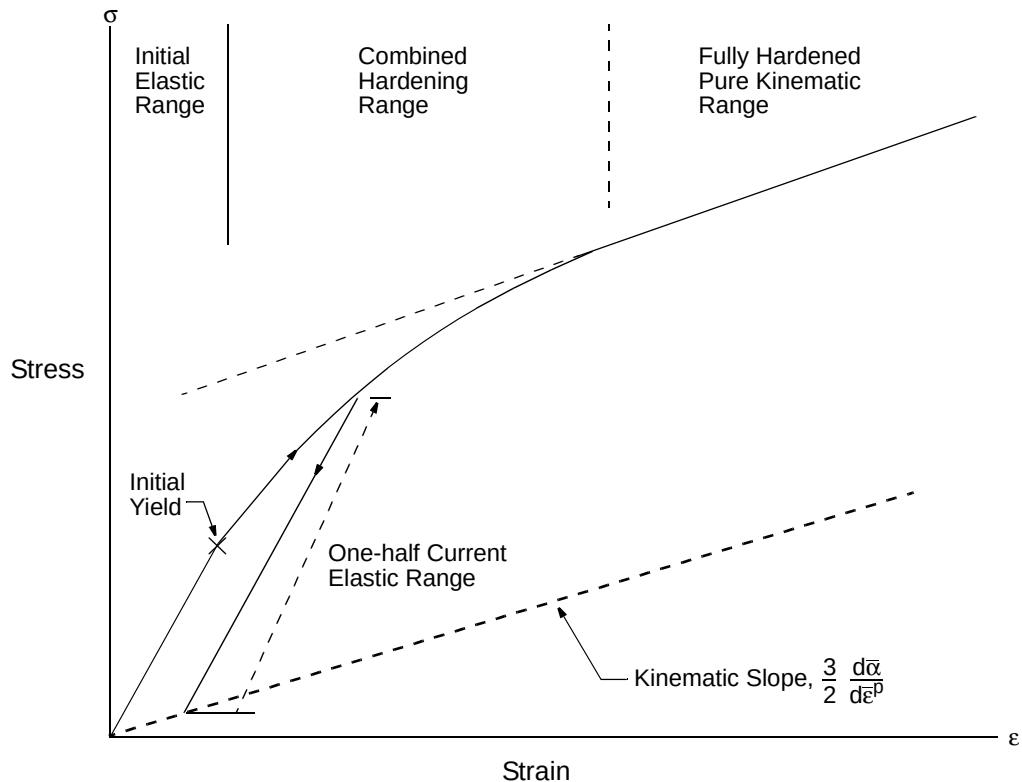


Figure 7-34 Basic Uniaxial Tension Behavior of the Combined Hardening Model

Use the model definition options **ISOTROPIC** or **ORTHOTROPIC** to activate the combined workhardening option in MSC.Marc. Use the **WORK HARD** option or **WKSPL** user subroutine to input workhardening slope data.

The workhardening data at small strains governs the isotropic behavior, and the data at large strains governs the kinematic hardening behavior. If the last workhardening slope is zero, the behavior is the same as the isotropic hardening model.

In order to provide the generality of combined hardening model, the fraction factor, f_h is introduced as a user input. As shown in Figure 7-35, f is the value between 0 and 1.

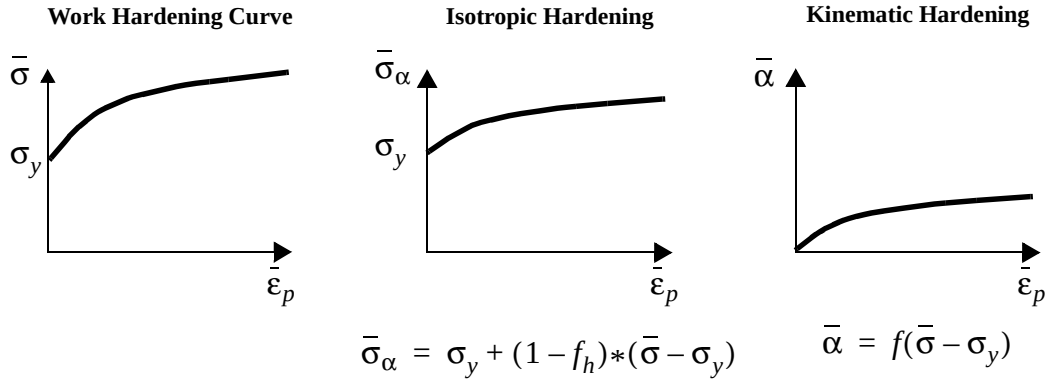


Figure 7-35 Schematic explanation for kinematic hardening fraction

Isotropic hardening $f_h = 0$

Kinematic hardening $f_h = 1$

Combined hardening $0 < f_h < 1$ (default $f = 0.5$)

Combined hardening model utilizing kinematic fraction factor is available for Hill and Barlat models in [ISOTROPIC](#), [ORTHOTROPIC](#), and [ANISOTROPIC](#) options.

Flow Rule

Yield stress and workhardening rules are two experimentally related phenomena that characterize plastic material behavior. The flow rule is also essential in establishing the incremental stress-strain relations for plastic material. The flow rule describes the differential changes in the plastic strain components $d\epsilon^p$ as a function of the current stress state.

The Prandtl-Reuss representation of the flow rule is available in MSC.Marc. In conjunction with the von Mises yield function, this can be represented as:

$$d\epsilon_{ij}^p = d\bar{\epsilon}^p \frac{\partial \bar{\sigma}}{\partial \sigma'_{ij}} \quad (7-132)$$

where $d\bar{\epsilon}^p$ and σ are equivalent plastic strain increment and equivalent stress, respectively.

The significance of this representation is illustrated in [Figure 7-36](#). This figure illustrates the “stress-space” for the two-dimensional case. The solid curve gives the yield surface (locus of all stress states causing yield) as defined by the von Mises criterion.

[Equation 7-132](#) expresses the condition that the direction of inelastic straining is normal to the yield surface. This condition is called either the normality condition or the associated flow rule.

If the von Mises yield surface is used, then the normal is equal to the deviatoric stress.

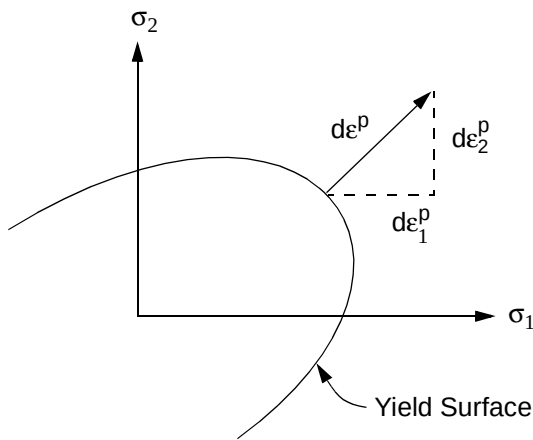


Figure 7-36 Yield Surface and Normality Criterion 2-D Stress Space

Constitutive Relations

This section presents the constitutive relation that describes the incremental stress-strain relation for an elastic-plastic material. The material behavior is governed by the incremental theory of plasticity, the von Mises yield criterion, and the isotropic hardening rule.

Let the workhardening coefficient H be expressed as

$$H = d\bar{\sigma} / d\bar{\epsilon}^p \quad (7-133)$$

and the flow rule be expressed as

$$d\epsilon^p = d\bar{\epsilon}^p \left\{ \frac{d\bar{\sigma}}{d\sigma} \right\} \quad (7-134)$$

Consider the differential form of the familiar stress-strain law, with the plastic strains interpreted as initial strains

$$d\sigma = C : d\varepsilon - C : d\varepsilon^p \quad (7-135)$$

where C is the elasticity matrix defined by Hooke's law.

After substitution of [Equation 7-134](#), this becomes

$$d\sigma = C : d\varepsilon - C : \frac{\partial \bar{\sigma}}{\partial \sigma} d\bar{\varepsilon}^p \quad (7-136)$$

Contracting [Equation 7-136](#) by $\frac{\partial \bar{\sigma}}{\partial \sigma}$

$$\left\{ \frac{\partial \bar{\sigma}}{\partial \sigma} \right\} : d\sigma = \left\{ \frac{\partial \bar{\sigma}}{\partial \sigma} \right\} : C : d\varepsilon - \left\{ \frac{\partial \bar{\sigma}}{\partial \sigma} \right\} : C : \left\{ \frac{\partial \bar{\sigma}}{\partial \sigma} \right\} d\bar{\varepsilon}^p \quad (7-137)$$

and recognizing that

$$d\bar{\sigma} = \frac{\partial \bar{\sigma}}{\partial \sigma} : d\sigma \quad (7-138)$$

with use of [Equation 7-133](#) in place of the left-hand side,

$$H d\bar{\varepsilon}^p = \left\{ \frac{\partial \bar{\sigma}}{\partial \sigma} \right\} : C : d\varepsilon - \left\{ \frac{\partial \bar{\sigma}}{\partial \sigma} \right\} : C : \left\{ \frac{\partial \bar{\sigma}}{\partial \sigma} \right\} d\bar{\varepsilon}^p \quad (7-139)$$

By rearrangement

$$d\bar{\varepsilon}^p = \frac{\left\{ \frac{\partial \bar{\sigma}}{\partial \sigma} \right\} : C : d\varepsilon}{H + \left\{ \frac{\partial \bar{\sigma}}{\partial \sigma} \right\} : C : \left\{ \frac{d\bar{\sigma}}{d\sigma} \right\}} \quad (7-140)$$

Finally, by substitution of this expression into [Equation 7-136](#), we obtain

$$d\sigma = L^{ep} : d\varepsilon \quad (7-141)$$

where L^{ep} is the elasto-plastic, small strain tangent moduli expressed as:

$$L^{ep} = C - \frac{\left(C : \frac{\partial \bar{\sigma}}{\partial \sigma}\right) \otimes \left(C : \frac{\partial \bar{\sigma}}{\partial \sigma}\right)}{H + \left(\frac{\partial \bar{\sigma}}{\partial \sigma}\right) : C : \left(\frac{\partial \bar{\sigma}}{\partial \sigma}\right)} \quad (7-142)$$

The case of perfect plasticity, where $H = 0$, causes no difficulty.

Temperature Effects

This section discusses the effects of temperature-dependent plasticity on the constitutive relation.

The following constitutive relations for thermo-plasticity were developed by Naghdi. Temperature effects are discussed using the isotropic hardening model and the von Mises yield condition.

The stress rate can be expressed in the form

$$\dot{\sigma}_{ij} = L_{ijkl} \dot{\epsilon}_{kl} + h_{ij} \dot{T} \quad (7-143)$$

For elastic-plastic behavior, the moduli L_{ijkl} are

$$L_{ijkl} = C_{ijkl} - \left(C_{ijmn} \frac{\partial \bar{\sigma}}{\partial \sigma_{mn}} \frac{\partial \bar{\sigma}}{\partial \sigma_{pq}} C_{pqkl} \right) / D \quad (7-144)$$

and for purely elastic response

$$L_{ijkl} = C_{ijkl} \quad (7-145)$$

The term that relates the stress increment to the increment of temperature for elastic-plastic behavior is

$$h_{ij} = X_{ij} - C_{ijkl} \alpha_{kl} - \left(C_{ijkl} \frac{\partial \bar{\sigma}}{\partial \sigma_{kl}} \left(\sigma_{pq} X_{pq} - \frac{2}{3} \bar{\sigma} \frac{\partial \bar{\sigma}}{\partial T} \right) \right) / D \quad (7-146)$$

and for purely elastic response

$$H_{ij} = X_{ij} - C_{ijkl} \alpha_{kl} \quad (7-147)$$

where

$$D = \frac{4}{9} \bar{\sigma}^2 \frac{\partial \bar{\sigma}}{\partial \bar{\epsilon}^p} + \frac{\partial \bar{\sigma}}{\partial \sigma_{ij}} C_{ijkl} \frac{\partial \bar{\sigma}}{\partial \sigma_{kl}} \quad (7-148)$$

and

$$X_{ij} = \frac{\partial C_{ijkl}}{\partial T} \epsilon_{kl}^e \quad (7-149)$$

and α_{kl} are the coefficients of thermal expansion.

Strain Rate Effects

This section discusses the influence of strain rate on the elastic-plastic constitutive relation.

Strain rate effects cause the structural response of a body to change because they influence the material properties of the body. These material changes lead to an instantaneous change in the strength of the material. Strain rate effects become more pronounced for temperatures greater than half the melting temperature (T_m). The following discussion explains the effect of strain rate on the size of the yield surface.

Using the von Mises yield condition and normality rule, we obtain an expression for the stress rate of the form

$$\dot{\sigma}_{ij} = L_{ijkl} \dot{\epsilon}_{kl} + r_{ij} \ddot{\epsilon}^p \quad (7-150)$$

For elastic-plastic response

$$L_{ijkl} = C_{ijkl} - \left(C_{ijmn} \frac{\partial \bar{\sigma}}{\partial \sigma_{mn}} \frac{\partial \bar{\sigma}}{\partial \sigma_{pq}} C_{pqkl} \right) / D \quad (7-151)$$

and

$$r_{ij} = C_{ijmn} \frac{\partial \bar{\sigma}}{\partial \sigma_{mn}} \frac{2}{3} \bar{\sigma} \frac{\partial \bar{\sigma}}{\partial \bar{\epsilon}^p} / D \quad (7-152)$$

where

$$D = \frac{4}{9} \bar{\sigma}^2 \frac{\partial \bar{\sigma}}{\partial \bar{\epsilon}^p} + \frac{\partial \bar{\sigma}}{\partial \sigma_{ij}} C_{ijkl} \frac{\partial \bar{\sigma}}{\partial \sigma_{kl}} \quad (7-153)$$

Time-independent Cyclic Plasticity

The cyclic plasticity model is based on the work of Chaboche [Ref. 23]. The current version of MSC.Marc consists only of the basic model and plastic-strain-range memorization. The associated time-dependent model is described on [Time-dependent Cyclic Plasticity](#).

The model combines the isotropic hardening rule, to describe the cyclic hardening ([Figure 7-37a](#)) or softening, and the nonlinear kinematic hardening to capture the proper characteristic of cyclic plasticity like Bauschinger ([Figure 7-37b](#)), ratcheting ([Figure 7-37c](#)), and mean-stress relaxation ([Figure 7-37d](#)) effect. The influence of the plastic strain range on the stabilized cyclic response is taken into account by introducing the plastic-strain-range memorization variable ([Figure 7-37e](#)).

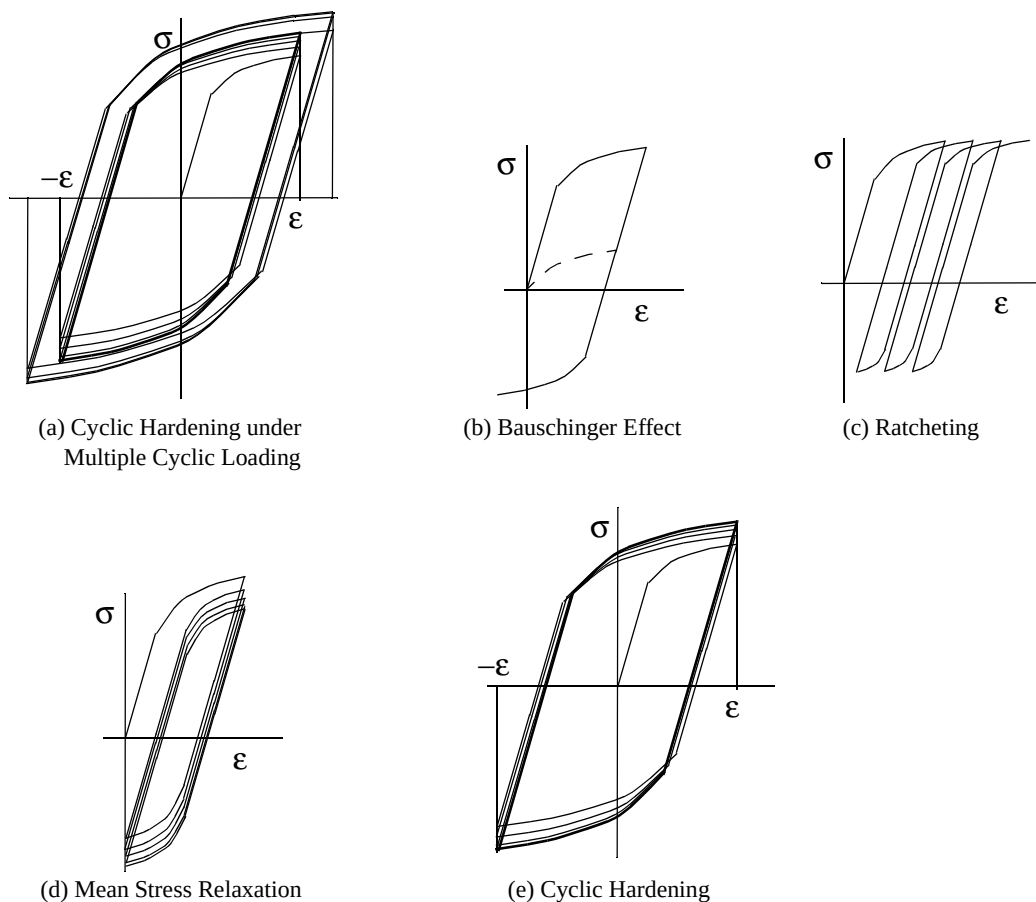


Figure 7-37 Typical Behavior of Material that can be Simulated with Cyclic Plasticity Model

The total strain rate can be divided into elastic and plastic part as follows:

$$\dot{\boldsymbol{\varepsilon}} = \dot{\boldsymbol{\varepsilon}}^e + \dot{\boldsymbol{\varepsilon}}^p \quad (7-154)$$

and Hooke's law describes the elastic stress-strain relationship as follows:

$$\boldsymbol{\sigma} = L\boldsymbol{\varepsilon}^e \quad (7-155)$$

The rate-independent plastic strain increment is derived from the normality (associative) rule as follows:

$$\dot{\boldsymbol{\varepsilon}}^p = \dot{\lambda} \frac{\partial f}{\partial \boldsymbol{\sigma}}$$

where f is the Misses yield function defined as follows:

$$f \equiv \bar{\sigma} - (R + k)$$

$$\text{where } \bar{\sigma} = \left(\frac{3ss}{2} \right)^{\frac{1}{2}}, s_{ij} = \sigma'_{ij} - \frac{1}{3} \delta_{ij} \sigma'_{kk} \text{ and } \sigma' = \sigma - X$$

X is the back stress tensor representing the center of the yield surface in stress space.

Isotropic Hardening/Softening

The isotropic hardening/softening determines the size of the elastic region during the plastic loading. In this model, it is controlled by parameter R and k . The initial conditions of cyclic hardening are given as $k = \sigma_y$ and $R = 0$, while a cyclic softening is initially described by $k = \sigma_y - R_0$ and $R = R_0$. The evolution equation for the variable R is described as follows:

$$\dot{R} = b(R_\infty - R)\dot{\lambda}$$

where b and R_∞ are material constants. R_∞ represents the limit of the isotropic hardening/softening. In case of hardening, then $R = R_\infty(1 - e^{-b\epsilon_{ps}})$.

Nonlinear Kinematic Hardening

The nonlinear kinematic hardening is defined from the linear-Ziegler rule by adding the so-called recall term as shown in the evolution of the back stress tensor below:

$$\dot{X} = \frac{C}{R+k}(\sigma - X) - \gamma X \dot{\lambda}$$

where C and γ are two material constants. $\gamma = 0$ stands for linear-kinematic rule.

Plastic-strain-range Memorization

Several experimental observations show that the asymptotic stress value of cyclic hardening can depend on the prior history. The influence of plastic-strain range on the stabilized cyclic response is evident from the comparison between the different histories of loading used to obtained the cyclic curve. Therefore, an introduction of new internal variables that memorize the prior maximum plastic range is introduced by defining a “memory” surface in the plastic strain space as follows:

$$F = \frac{2}{3}\epsilon_e(\epsilon_p - \zeta) - \rho$$

The evolution of the state variables are as follows

$$\begin{aligned}\dot{\rho} &= \eta H(F) \langle n \bullet n^* \rangle \dot{\lambda} \\ \dot{\zeta} &= \sqrt{3/2} (1 - \eta) H(F) \langle n \bullet n^* \rangle n^* \dot{\lambda}\end{aligned}$$

where n and n^* are the unit normal to the yield surface $f = 0$ and to the memory surface $F = 0$ defined as follows:

$$n = \sqrt{\frac{2}{3}} \frac{\epsilon_p}{\dot{\lambda}} \text{ and } n^* = \sqrt{\frac{2}{3}} \frac{\epsilon_p - \zeta}{\rho}$$

The coefficient η is introduced in order to induce a progressive memory. For $\eta = 0.5$ then, the memorization is instantaneous and stabilization occurs after one cycle. A progressive memory is given by $\eta < 0.5$.

The dependency between cyclic plastic flow and the plastic strain range is introduced by considering an asymptotic isotropic state as follows:

$$R_\infty = Q_M + (Q_0 - Q_M)e^{-2\mu\rho}$$

where Q_M , Q_0 and μ are material constants.

Plastic Evolution Process and Elasto-plastic “Classical” Modular Matrix

The plastic evolution process must conform to the consistency condition, $\dot{f} = 0$. From this condition the plastic “rate” multiplier can be derived as follows:

$$\dot{\lambda} = \frac{a^T L}{a^T L a + \frac{C}{R+k} a^T (\sigma - X) - \gamma a^T X + b(R_\infty - R)} \dot{\epsilon} \quad (7-156)$$

$$\text{where } a = \frac{\partial f}{\partial \sigma} = \frac{3}{2} \frac{s}{R+k}.$$

Since the process involves nonlinear equation, iteration process using predictor-corrector technique is used. The predictor is calculated using the “trial” elastic stresses as follows:

$$\sigma_B = \sigma_A + L \Delta \sigma$$

and then calculate f_B based on σ_B and hardening parameter on A. Using the Taylor expansion at B, then

$$\delta \lambda = \frac{f_B}{a^T L a + \frac{C}{R+k} a^T (\sigma - X) - \gamma a^T X + b(R_\infty - R)}$$

Having the global plasticity iteration converged, then the iteration to satisfy the plastic strain memorization is started. If both iterations are converged, then the total plasticity iteration is considered completed.

Inserting Equation 7-162 into Equation 7-161 and using Equation 7-160, the elasto-plastic “classical” tangent modular matrix can be derived as follows:

$$L^{EP} = L \left(I - \frac{a a^T L}{a^T L a + \frac{C}{R+k} a^T (\sigma - X) - \gamma a^T X + b(R_\infty - R)} \right) \quad (7-157)$$

Time-dependent Inelastic Behavior

Force-displacement relationships vary in different material models. A perfectly elastic material and a perfectly viscous material can be represented by a spring and a dashpot, respectively (as shown in [Figure 7-38](#)). In a perfectly elastic material, the deformation is proportional to the applied load. In a perfectly viscous material, the rate of change of the deformation over time is proportional to the load.

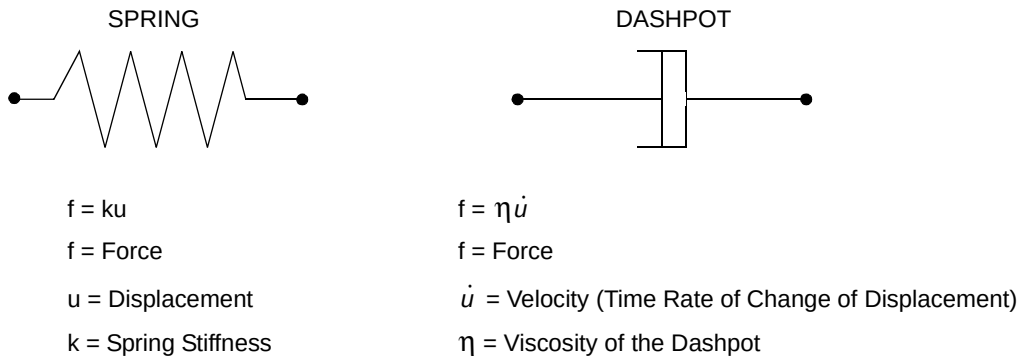


Figure 7-38 Perfectly Elastic (Spring) and Viscous (Dashpot) Materials

In the class of viscoelastic and creeping materials, the application of a constant load is followed by a deformation, which can be made up of an instantaneous deformation (elastic effect) followed by a continual deformation with time (viscous effect). Eventually, it can become pure viscous flow. Continued deformation under constant load is termed creep (see [Figure 7-39](#)).

A viscoelastic material can be subjected to sudden application of a constant deformation. This results in an instantaneous proportional load (elastic effect), followed by a gradual reduction of the required load with time, until a limiting value of the load is attained. The decreasing of load for a constant deformation, is termed relaxation (see [Figure 7-40](#)).

Viscoelastic and creeping materials can be represented by models consisting of both springs and dashpots because the material displays both elastic effects and viscous effects. This implies that the material either continues to flow for a given stress, or the stress decreases with time for a given strain. The measured relation between stress and strain is generally very complex.

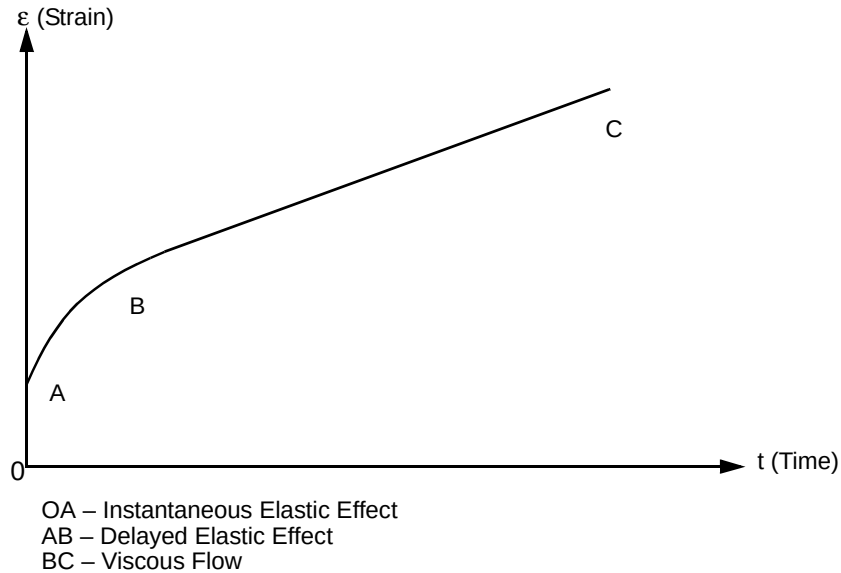


Figure 7-39 The Creep Curve

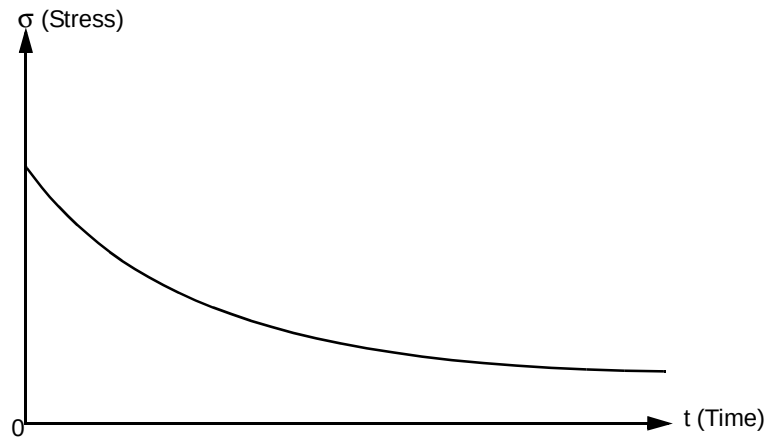


Figure 7-40 The Relaxation Curve

Two models that are commonly used to relate stress and strain are the Maxwell and Kelvin (Voigt or Kelvin-Voigt) models. A description of these models is given below.

The mathematical relation which holds for the Maxwell solid is

$$\dot{\epsilon} = \alpha \dot{\sigma} + \beta \sigma \quad (7-158)$$

In the one-dimensional case for normal stress

$$\alpha = \frac{1}{E}$$

$$\beta = \frac{1}{\eta}$$
(7-159)

This relation can be depicted as a spring and dashpot in series, as shown in [Figure 7-41](#). The integration of [Equation 7-158](#) yields

$$\varepsilon = \frac{\sigma}{E} + \int \frac{\sigma}{\eta} dt$$
(7-160)

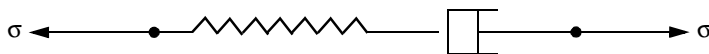


Figure 7-41 Maxwell Solid

The strain and stress responses of the Maxwell Solid model are shown in [Figure 7-42](#) and [Figure 7-43](#), respectively.

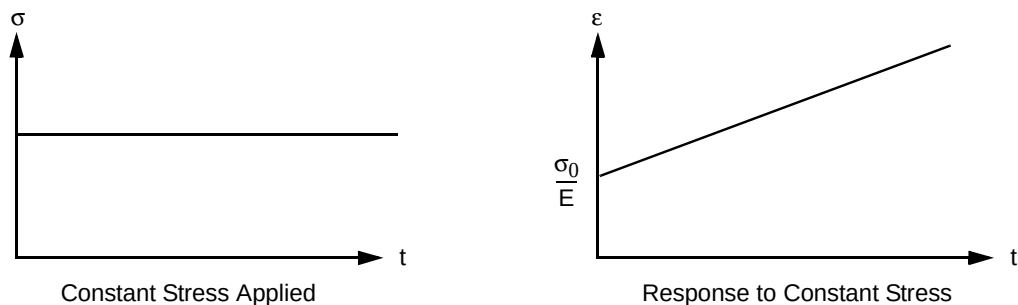


Figure 7-42 Strain Response to Applied Constant Stress (Maxwell Solid)

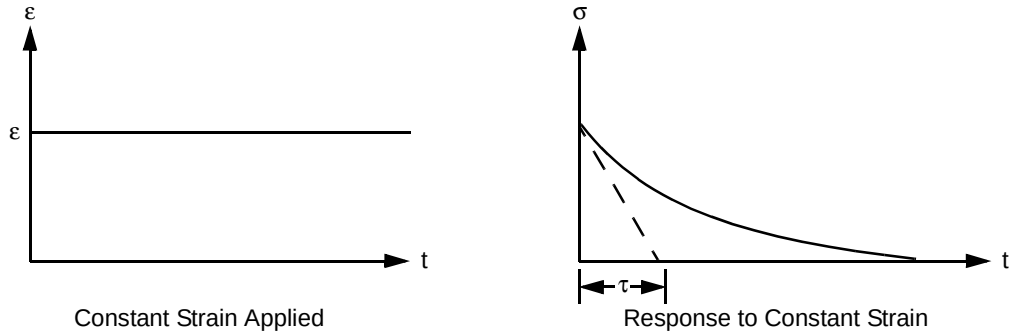


Figure 7-43 Stress Response to Applied Constant Strain (Maxwell Solid)

The mathematical relation which holds for the Kelvin (Voigt or Kelvin-Voigt) solid is

$$\sigma = \alpha \epsilon + \beta \dot{\epsilon} \quad (7-161)$$

This equation is depicted as a spring and dashpot in parallel. (See [Figure 7-44](#)). When $\beta = 0$ (no dashpot), the system is a linearly elastic system in which $\alpha = E$, the elastic modulus.

When $E = 0$ (no spring), the “solid” obeys Newton’s equation for a viscous fluid and $\beta = \eta$, the viscous coefficient. Thus, we can rewrite [Equation 7-161](#) in the form

$$\sigma = E\epsilon + \eta \dot{\epsilon} \quad (7-162)$$

In the above relation, we have considered one-dimensional normal stress and strain. The relation holds equally well for shear stress τ and shear strain γ in which $\alpha = G$, the shear modulus, and $\beta = \eta$, the viscous coefficient. [Equation 7-161](#) can be rewritten as

$$\tau = G\gamma + \eta \dot{\gamma} \quad (7-163)$$

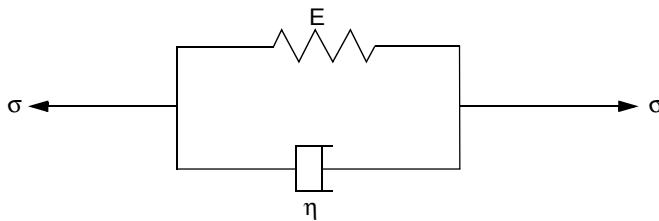


Figure 7-44 Kelvin (Voigt or Kelvin-Voigt) Solid

The strain responses of the Kelvin Solid model are depicted in Figure 7-45. For multiaxial situations, these equations can be generalized to tensor quantities.

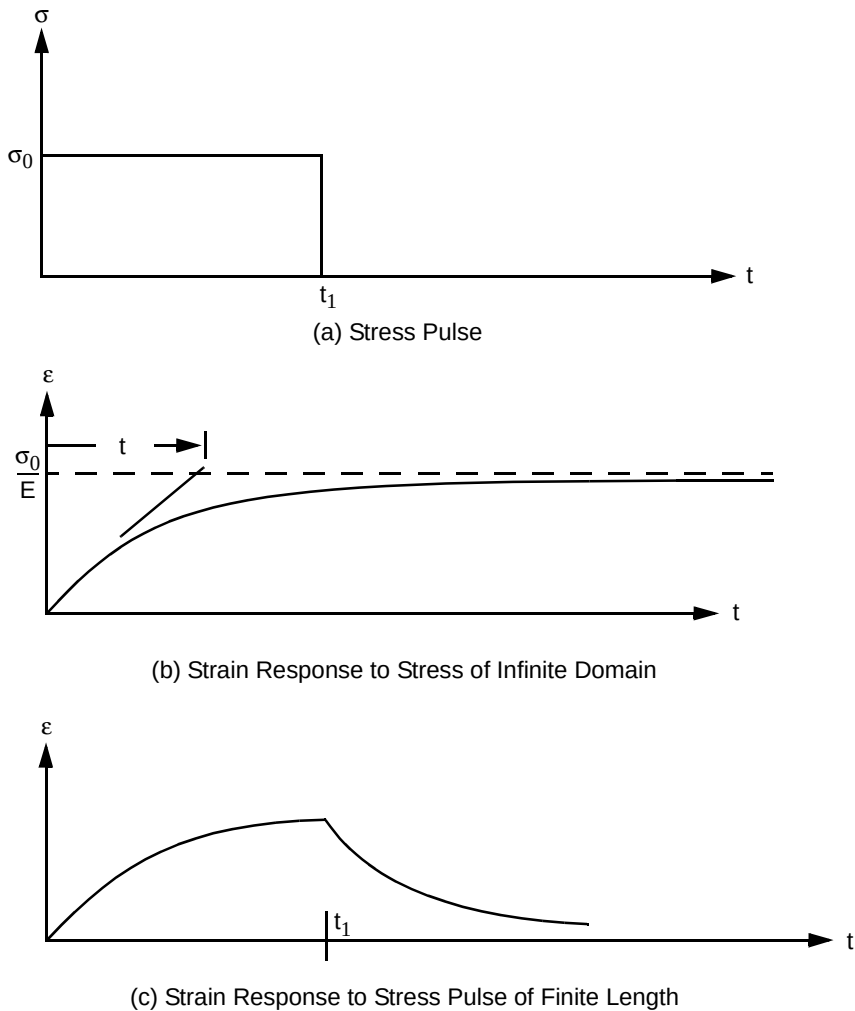


Figure 7-45 Strain Response to Applied Stress (Kelvin Solid)

To invoke the Maxwell model, use the **CREEP** parameter. The creep strain can be specified as either deviatoric creep strain (conventional creep) or dilatational creep strain (swelling). To invoke the Kelvin model, also use the **CREEP** parameter and **CRPVIS** user subroutine.

Creep (Maxwell Model)

Creep is an important factor in elevated-temperature stress analysis. In MSC.Marc, creep is represented by a Maxwell model. Creep is a time-dependent, inelastic behavior, and can occur at any stress level (that is, either below or above the yield stress of a material). In many cases, creep is accompanied by plasticity which occurs above the yield stress of the material. The creep behavior can be characterized as primary, secondary, and tertiary creep, as shown in [Figure 7-46](#). Engineering analysis is often limited to the primary and secondary creep regions. Tertiary creep in a uniaxial specimen is usually associated with geometric instabilities, such as necking. The major difference between the primary and secondary creep is that the creep strain rate is much larger in the primary creep region than it is in the secondary creep region. The creep strain rate is the slope of the creep strain-time curve. The creep strain rate is generally dependent on stress, temperature, and time.

The creep data can be specified in either an exponent form or in a piecewise linear curve. To specify creep data, use the model definition option [CREEP](#). Subroutine [CRPLAW](#) allows alternative forms of creep behavior to be programmed directly.

$$\dot{\bar{\epsilon}}^c = \frac{d\bar{\epsilon}^c}{dt} \quad (7-164)$$

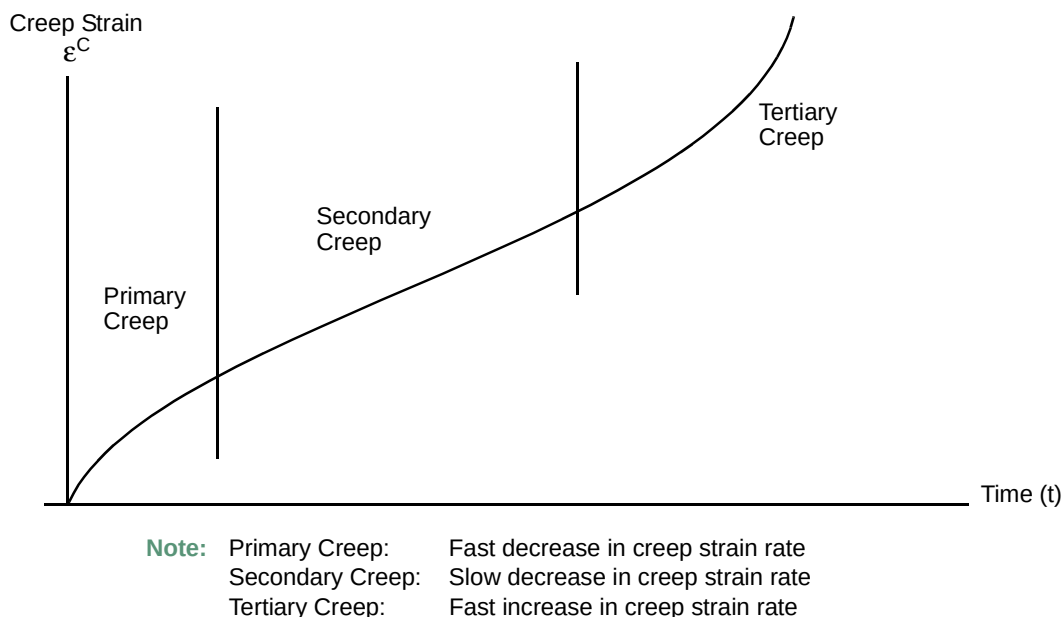


Figure 7-46 Creep Strain Versus Time (Uniaxial Test at Constant Stress and Temperature)

MSC.Marc offers two schemes for modeling creep in conjunction with plasticity: (a) treating creep strains and plastic strains separately; and (b) modeling creep strains and plastic strains in a unified fashion (viscoplasticity). Both schemes can be treated using two different procedures: explicit and implicit.

Creep (Explicit Formulation)

There are four possible modes of input for creep constitutive data.

1. Express the dependence of equivalent creep strain rate on any independent parameter through a piecewise linear relationship. The equivalent creep strain rate is then assumed to be a piecewise linear approximation to

$$\dot{\bar{\epsilon}}^c = A \cdot f(\bar{\sigma}) \cdot g(\bar{\epsilon}^c) \cdot h(T) \cdot \frac{dk(t)}{dt} \quad (7-165)$$

where A is a constant; $\dot{\bar{\epsilon}}^c$ is equivalent creep strain rate; and $\bar{\sigma}$, $\bar{\epsilon}^c$, T , and t are equivalent stress, equivalent creep strain, temperature and time, respectively. The functions f , g , h , and k are piecewise linear and entered in the form as either slope-break point data or function-variable data. This representation is shown in Figure 7-47. Enter functions f , g , h , and k through the model definition option **CREEP**. (Any of the functions f , g , h , or k can be set to unity by setting the number of piecewise linear slopes for that relation to zero on the input data.)

2. The dependence of equivalent creep strain rate on any independent parameter can be given directly in power law form by the appropriate exponent. The equivalent creep strain rate is

$$\dot{\bar{\epsilon}}^c = A \bar{\sigma}^m \cdot (\bar{\epsilon}^c)^n \cdot T^p \cdot (qt^{q-1}) \quad (7-166)$$

Enter the constants A , m , n , p , and q directly through the model definition option **CREEP**. This is often adequate for engineering metals at constant temperature where Norton's rule is a good approximation.

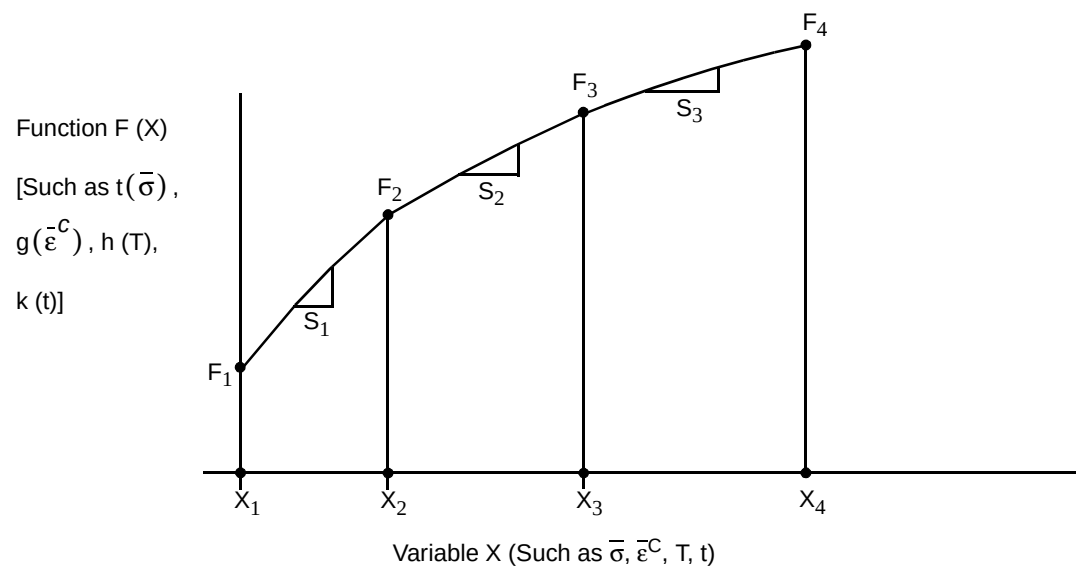
$$\dot{\bar{\epsilon}}^c = A \bar{\sigma}^n \quad (7-167)$$

3. Define the equivalent creep strain rate directly with the user subroutine **CRPLAW**.
4. Use the **ISOTROPIC** option to activate the ORNL (Oak Ridge National Laboratory rules) capability of the program.

Isotropic creep behavior is based on a von Mises creep potential described by the equivalent creep law

$$\dot{\bar{\epsilon}} = f(\bar{\sigma}, \bar{\epsilon}^c, T, t)$$

(7-168)



(1) Slope-Break Point Data

<u>Slope</u>	<u>Break Point</u>
S_1	X_1
S_2	X_2
S_3	X_3

(2) Function-Variable Data

<u>Function</u>	<u>Variable</u>
F_1	X_1
F_2	X_2
F_3	X_3
F_4	X_4

Figure 7-47 Piecewise Linear Representation of Creep Data

The material creep behavior is described by

$$\dot{\epsilon}_{ij}^c = \dot{\epsilon}^c \left\{ \frac{\partial \bar{\sigma}}{\partial \sigma_{ij}} \right\} \quad (7-169)$$

During creep, the creep strain rate usually decreases. This effect is called creep hardening and can be a function of time or creep strain. The following section discusses the difference between these two types of hardening.

Consider a simple power law that illustrates the difference between time and strain-hardening rules for the calculation of the creep strain rate.

$$\epsilon^c = \beta t^n \quad (7-170)$$

where ϵ^c is the creep strain, β and n are values obtained from experiments and t is time. The creep rate can be obtained by taking the derivative ϵ^c with respect to time

$$\dot{\epsilon}^c = \frac{d\epsilon^c}{dt} = n\beta t^{n-1} \quad (7-171)$$

However, t being greater than 0, we can compute the time t as

$$t = \left(\frac{\epsilon^c}{\beta} \right)^{1/n} \quad (7-172)$$

Substituting Equation 7-171 into Equation 7-170 we have

$$\dot{\epsilon}^c = n\beta t^{n-1} = n(\beta^{1/n}(\epsilon^c)^{((n-1)/n)}) \quad (7-173)$$

Equation 7-171 shows that the creep strain rate is a function of time (time hardening). Equation 7-173 indicates that the creep strain rate is dependent on the creep strain (strain hardening). The creep strain rates calculated from these two hardening rules generally are different. The selection of a hardening rule in creep analysis must be based on data obtained from experimental results. Figure 7-48 and Figure 7-49 show time and strain hardening rules in a variable state of stress. It is assumed that the stress in a structure varies from σ_1 to σ_2 to σ_3 ; depending upon the model chosen, different creep strain rates are calculated accordingly at points 1, 2, 3, and 4. Obviously, creep strain rates obtained from the time hardening rule are quite different from those obtained by the strain hardening rule.

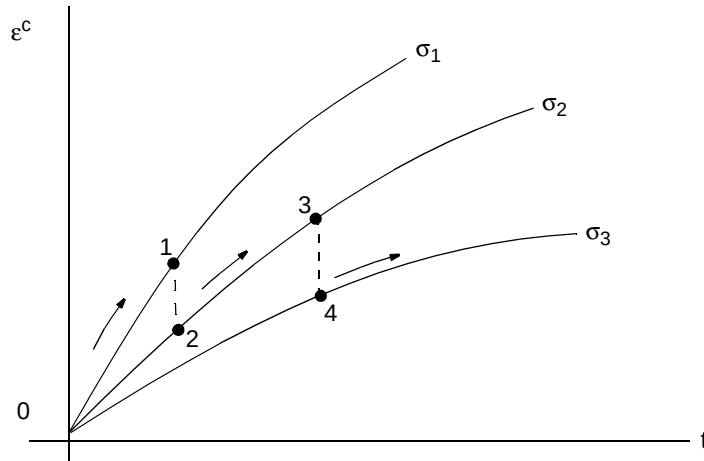


Figure 7-48 Time Hardening

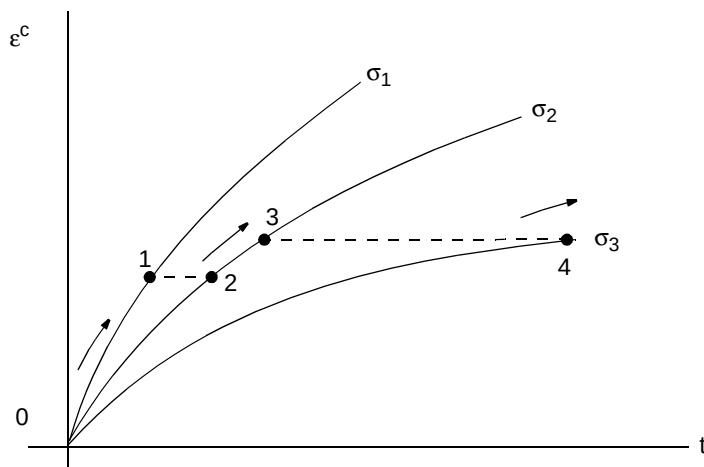


Figure 7-49 Strain Hardening

Oak Ridge National Laboratory Laws

Oak Ridge National Laboratory (ORNL) has performed a large number of creep tests on stainless and other alloy steels. It has also set certain rules that characterize creep behavior for application in nuclear structures. A summary of the ORNL rules on creep is given below. The references listed at the end of this section offer a more detailed discussion of the ORNL rules.

1. Auxiliary Rules for Applying Strain-Hardening to Situations Involving Stress Reversals

The Blackburn Creep Law is required as user subroutine **CRPLAW**. The parameter EQCP (first parameter in **CRPLAW**) is defined as

$$\bar{\epsilon}^c = \left(\frac{2}{3} \Sigma \Delta \epsilon_{ij}^c \Sigma \Delta \epsilon_{ij}^c \right)^{1/2} \quad (7-174)$$

when the ORNL constitutive option is flagged through use of the **ISOTROPIC** option. In all other cases, the definition

$$\bar{\epsilon}^c = \Sigma \left(\frac{2}{3} \Delta \epsilon_{ij}^c \Delta \epsilon_{ij}^c \right)^{1/2} \quad (7-175)$$

is retained. The equivalent primary creep strain passes into **CRPLAW** in EQCPNC, the second parameter. The second parameter must be redefined in that routine as the equivalent (total) creep strain increment. The first parameter (EQCP) must be redefined as the equivalent primary creep strain increment when the ORNL constitutive option is flagged. During analysis with the ORNL option, equivalent creep strain stores the distance between the two shifted origins in creep strain space (ϵ in ORNL-TM-3602). The sign on this value indicates which origin is currently active, so that a negative sign indicates use of the “negative” origin ($-\epsilon_{ij}$).

2. Plasticity Effect on Creep

The effect of plastic strains on creep must be accommodated for the time-dependent creep behavior of 2 1/4 Cr -1 Mo Steel. Since plastic strains in one direction reduce the prior creep strain hardening accumulated in the reverse direction, ORNL recommends that the softening influence due to plastic strains be treated much the same as when reversed creep strain occurs. The following quantities are defined:

$$N_{ij}^+ = (\epsilon_{ij}^I - \epsilon_{ij}^+) / G^+ \quad (7-176)$$

$$N_{ij}^- = (\epsilon_{ij}^I - \epsilon_{ij}^-) / G^- \quad (7-177)$$

where ϵ_{ij}^I is instantaneous creep strain components

$$\epsilon_{ij}^+; \quad \epsilon_{ij}^- \quad = \text{positive and negative strain origins} \quad (7-178)$$

and

$$G^+ = G(\epsilon_{ij}^I - \epsilon_{ij}^+) = [2/3(\epsilon_{ij}^I - \epsilon_{ij}^+)(\epsilon_{ij}^I - \epsilon_{ij}^+)]^{1/2} \quad (7-179)$$

$$G^- = G(\epsilon_{ij}^I - \epsilon_{ij}^-) = [2/3(\epsilon_{ij}^I - \epsilon_{ij}^-)(\epsilon_{ij}^I - \epsilon_{ij}^-)]^{1/2} \quad (7-180)$$

Swelling

Marc allows pure swelling (dilatational creep) effect in a creep analysis. To use the swelling option, perform a regular creep analysis as discussed earlier. Use the user subroutine **VSWELL** to define the increment of volumetric swelling $\left(\frac{\Delta V}{V}\right)$. The increment of volumetric swelling is generally a function of neutron flux, time, and temperatures.

For example, radiation-induced swelling strain model for 20% C. W. Stainless Steel 316 can be expressed as:

$$\frac{\Delta V}{V} = R\phi t + \frac{R}{\alpha} \ln \left[\frac{1 + \exp(\alpha(\tau - \phi t))}{1 + \exp \tau} \right] \quad (7-181)$$

where R , t , and α are functions of temperature, ϕ is neutron flux, and t is time.

Creep (Implicit Formulation)

This formulation, as opposed to that described in the previous section, is fully implicit. A fully implicit formulation is unconditionally stable for any choice of time step size; hence, allowing a larger time step than permissible using the explicit method. Additionally, this is more accurate than the explicit method. The disadvantage is that each increment may be more computationally expensive. This model is activated using the **CREEP** parameter. There are two methods for defining the inelastic strain rate. The **CREEP** model definition option can be used to define a Maxwell creep model. The back stress must be specified through the field normally reserved for the yield stress in the **ISOTROPIC** or **ORTHOTROPIC** options. The yield stress must be specified through the field normally reserved for the 10th cycle yield stress in the **ISOTROPIC** option. There is no plastic strain when the stress is less than the yield stress and there is no creep strain when the stress is less than the back stress.

The equivalent creep strain increment is expressed as

$$\dot{\bar{\epsilon}}^c = A \bar{\sigma}^m \cdot (\bar{\epsilon}^c)^n \cdot T^p \cdot (qt^{q-1}) \quad (7-182)$$

Enter the constants A , m , n , p , and q directly through the **CREEP** model definition option. A more general expression for the equivalent creep strain rate is given by:

$$\dot{\bar{\epsilon}}^c = A \cdot \bar{\sigma}^m \cdot g(\bar{\epsilon}^c) \cdot h(T) \cdot \frac{dk(t)}{dt} \quad (7-183)$$

Enter the terms A and m and the functions g , h , and k through user subroutine **UCRPLW**.

The creep strain components are given by:

$$\Delta \bar{\epsilon}_{ij}^c = \frac{3}{2} \Delta \bar{\epsilon}^c \frac{\sigma'_{ij}}{\bar{\sigma}}$$

where σ'_{ij} is the deviatoric stress at the end of the increment and $\bar{\sigma}_y$ is the back stress.

A is a function of temperature, time, etc. An algorithmic tangent is used to form the stiffness matrix.

Based on a parameter defined in the **CREEP** parameter, one of three tangent matrices is formed. The first is using an elastic tangent, which requires more iterations, but can be computationally efficient because re-assembly might not be required. The second is an algorithmic tangent that provides the best behavior for small strain power law creep. The third is a secant (approximate) tangent that gives the best behavior for general viscoplastic models. When creep is specified in conjunction with plasticity, the elastic tangent option is not available.

Viscoplasticity (Explicit Formulation)

The creep (Maxwell) model can be modified to include a plastic element (as shown in **Figure 7-50**). This plastic element is inactive when the stress (σ) is less than the yield stress (σ_y) of the material. The modified model is an elasto-viscoplasticity model and is capable of producing some observed effects of creep and plasticity. In addition, the viscoplastic model can be used to generate time-independent plasticity solutions when stationary conditions are reached. At the other extreme, the viscoplastic model can reproduce standard creep phenomena. The model allows the treatment of nonassociated flow rules and strain softening which present difficulties in conventional (tangent modulus) plasticity analyses.

The viscoplasticity option can be used to implement very general constitutive relations with the aid of the following user subroutines: [ZERO](#), [YIEL](#), [NASSOC](#), and [CRPLAW](#). See [Nonlinear Analysis](#) in [Chapter 5](#) for details on how to use these procedures.

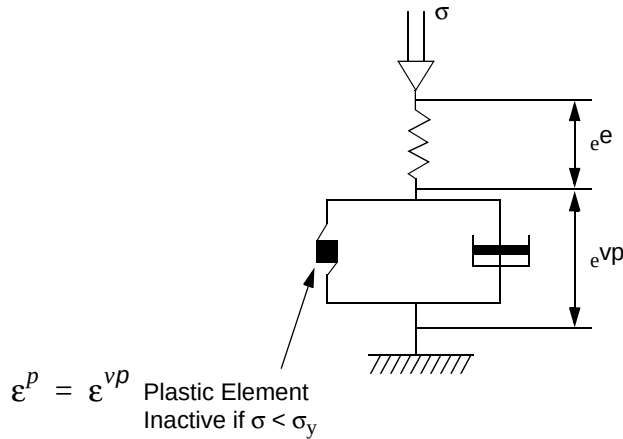


Figure 7-50 Uniaxial Representation of Viscoplastic Material

Time-dependent Cyclic Plasticity

The time-dependent effect of the model described in [Time-independent Cyclic Plasticity](#) on page 7-88 is modeled using the unified viscoplastic framework. The viscoplastic potential is based on “overstress” quantity as follows:

$$\Omega = \frac{K}{n+1} \left\langle \frac{f}{K} \right\rangle^{n+1}$$

The viscoplastic strain rate is defined as follows:

$$\dot{\epsilon}_{vp} = \frac{\partial \Omega}{\partial \sigma} = \frac{3}{2} \dot{\lambda} \frac{\sigma' - X'}{\tilde{\sigma}_e(\sigma - X)}$$

where

$$\dot{\lambda} = \left\langle \frac{f}{K} \right\rangle^n \quad (7-184)$$

In this case, the viscoplastic stress is $\sigma_{vp} = K \dot{\lambda}^{\frac{1}{n}}$.

The hardening rules are chosen to be identical to the time-independent case.

Viscoplastic Evolution Process and Visco-plastic “Classical” Modular Matrix

The iteration procedure, starting from the trial elastic stress as the predictor, is based on the implicit integration of Equation 7-184 that can be expressed as follows:

$$r_\lambda = -\frac{\Delta\lambda}{\Delta t} + \left\langle \frac{f}{K} \right\rangle^n \quad (7-185)$$

Using Newton iteration scheme, the incremental value of $\Delta\lambda$ can be expressed as follows:

$$\delta\lambda = \frac{\Delta t \, r_n^i}{1 + \frac{n}{K} \left\langle \frac{f}{K} \right\rangle^{n-1} \left(a^T L a + \frac{2}{3} C a^T a - \gamma a^T X + b(R_\infty - R) \right)}$$

where r_λ^i is the i-th iteration of the residual of Equation 7-185.

The tangent modular matrix is based on the assumption that $\delta r_\lambda = 0$. Therefore the classical visco-plastic modular matrix can be expressed as follows:

$$L^{VP} = L \left(I - \frac{a a^T L}{a^T L a + \frac{2}{3} C a^T a - \gamma a^T X + b(R_\infty - R) + \frac{1}{\frac{n}{K} \left\langle \frac{f}{K} \right\rangle^{n-1} \Delta t}} \right)$$

This matrix is in line with the consistent model derived in [Ref. 20].

Viscoplasticity (Implicit Formulation)

To allow for the implementation of general unified creep-plasticity or viscoplastic models, user subroutine **UVSCPL** is available. This routine requires you to define only the inelastic strain rate. The program automatically calculates a tangent stiffness matrix (only elastic tangent or secant tangent can be used). This option is activated by indicating that the material is **VISCO PLAS** in the **ISOTROPIC** or **ORTHOTROPIC** option.

Viscoelastic Material

MSC.Marc has two models that represent viscoelastic materials. The first can be defined as a Kelvin-Voigt model. The latter is a general hereditary integral approach.

Kelvin-Voigt Model

The Kelvin model allows the rate of change of the inelastic strain to be a function of the total stress and previous strain. To activate the Kelvin model in MSC.Marc, use the **CREEP** parameter.

The Kelvin material behavior (viscoelasticity) is modeled by assuming an additional creep strain ϵ_{ij}^k , governed by

$$\frac{d}{dt}\epsilon_{ij}^k = A_{ijkl}\sigma'_{kl} - B_{ijkl}\epsilon_{kl}^k \quad (7-186)$$

where A and B are defined in the user subroutine **CRPVIS** and the total strain is

$$\epsilon_{ij} = \epsilon_{ij}^e + \epsilon_{ij}^p + \epsilon_{ij}^c + \epsilon_{ij}^k + \epsilon_{ij}^{th} \quad (7-187)$$

$$\epsilon_{ij}^{th} = \text{thermal strain components} \quad (7-188)$$

$$\epsilon_{ij}^e = \text{elastic strain components (instantaneous response)} \quad (7-189)$$

$$\epsilon_{ij}^p = \text{plastic strain components} \quad (7-190)$$

$$\epsilon_{ij}^c = \text{creep strains defined via the CRPLAW and VSWELL user subroutines} \quad (7-191)$$

$$\epsilon_{ij}^k = \text{Kelvin model strain components as defined above} \quad (7-192)$$

The **CRPVIS** user subroutine is called at each integration point of each element when the Kelvin model is used.

Use the **AUTO CREEP** option to define the time step and to set the tolerance control for the maximum strain in any increment.

The **CREEP** option allows Maxwell models to be included in series with the Kelvin model.

Hereditary Integral Model

The stress-strain equations in viscoelasticity are not only dependent on the current stress and strain state (as represented in the Kelvin model), but also on the entire history of development of these states. This constitutive behavior is most readily expressed in terms of hereditary or Duhamel integrals. These integrals are formed by considering the stress or strain build-up at successive times. Two equivalent integral forms exist: the stress relaxation form and the creep function form. In MSC.Marc, the stress relaxation form is used.

The viscoelasticity option in MSC.Marc can be used for both the small strain and large strain Mooney or Ogden material (total Lagrange formulation) stress-relaxation problems. A description of these models is as follows:

Small Strain Viscoelasticity

In the stress relaxation form, the constitutive relation can be written as a hereditary integral formulation

$$\sigma_{ij}(t) = \int_0^t G_{ijkl}(t-\tau) \frac{d\epsilon_{kl}(\tau)}{d\tau} d\tau + G_{ijkl}(t) \epsilon_{kl}(0) \quad (7-193)$$

The functions G_{ijkl} are called stress relaxation functions. They represent the response to a unit applied strain and have characteristic relaxation times associated with them. The relaxation functions for materials with a fading memory can be expressed in terms of Prony or exponential series.

$$G_{ijkl}(t) = G_{ijkl}^{\infty} + \sum_{n=1}^N G_{ijkl}^n \exp(-t/\lambda^n) \quad (7-194)$$

in which G_{ijkl}^n is a tensor of amplitudes and λ^n is a positive time constant (relaxation time). In the current implementation, it is assumed that the time constant is isotropic. In [Equation 7-194](#), G_{ijkl}^{∞} represents the long term modulus of the material.

The short term moduli (describing the instantaneous elastic effect) are then given by

$$G_{ijkl}^0 = G_{ijkl}(0) = G_{ijkl}^{\infty} + \sum_{n=1}^N G_{ijkl}^n \quad (7-195)$$

The stress can now be considered as the summation of the stresses in a generalized Maxwell model (Figure 7-51)

$$\sigma_{ij}(t) = \sigma_{ij}^{\infty}(t) + \sum_{n=1}^N \sigma_{ij}^n(t) \quad (7-196)$$

where

$$\sigma_{ij}^{\infty} = G_{ijkl}^{\infty} \epsilon_{kl}(t) \quad (7-197)$$

$$\sigma_{ij}^n = \int_0^t G_{ijkl}^n \exp[-(t-\tau)/\lambda^n] \frac{d\epsilon_{kl}(\tau)}{d\tau} d\tau \quad (7-198)$$

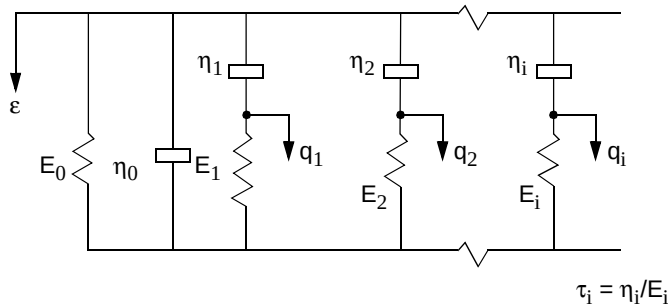


Figure 7-51 The Generalized Maxwell or Stress Relaxation Form

For integration of the constitutive equation, the total time interval is subdivided into a number of subintervals (t_{m-1}, t_m) with time-step $h = t_m - t_{m-1}$. A recursive relation can now be derived expressing the stress increment in terms of the values of the internal stresses σ_{ij}^n at the start of the interval. With the assumption that the strain varies linearly during the time interval h , we obtain the increment stress-strain relation as

$$\Delta \sigma_{ij}(t_m) = \left[G_{ijkl}^{\infty} + \sum_{n=1}^N \beta^n(h) G_{ijkl}^n \right] \Delta \epsilon_{kl} - \sum_{n=1}^N \alpha^n(h) \sigma_{ij}^n(t_m - h) \quad (7-199)$$

where

$$\alpha^n(h) = 1 - \exp(-h/\lambda^n) \quad (7-200)$$

and

$$\beta^n(h) = \alpha^n(h)\lambda^n/h \quad (7-201)$$

In MSC.Marc, the incremental equation for the total stress is expressed in terms of the short term moduli (See [Equation 7-195](#)).

$$\Delta\sigma_{ij}(t_m) = \left[G_{ijkl}^0 - \sum_{n=1}^N \{1 - \beta^n(h)\} G_{ijkl}^n \right] \Delta\epsilon_{kl}(t_m) - \sum_{n=1}^N \alpha^n(h) \sigma_{ij}^n(t_m - h) \quad (7-202)$$

In this way, the instantaneous elastic moduli can be specified through the [ISOTROPIC](#) or [ORTHOTROPIC](#) options. Moreover, since the [TEMPERATURE EFFECTS](#) option acts on the instantaneous elastic moduli, it is more straightforward to use the short term values instead of the long term ones. Note that the set of equations given by [Equation 7-202](#) can directly be used for both anisotropic and isotropic materials.

Isotropic Viscoelastic Material

For an isotropic viscoelastic material, MSC.Marc assumes that the deviatoric and volumetric behavior are fully uncoupled and that the behavior can be described by a time dependent shear and bulk modules. The bulk moduli is generally assumed to be time independent; however, this is an unnecessary restriction of the general theory.

Both the shear and bulk moduli can be expressed in a Prony series

$$G(t) = G^\infty + \sum_{n=1}^N G^n \exp(-t/\lambda_d^n) \quad (7-203)$$

$$K(t) = K^\infty + \sum_{n=1}^N K^n \exp(-t/\lambda_v^n) \quad (7-204)$$

with short term values given by

$$G^0 = G^\infty + \sum_{n=1}^N G^n \quad (7-205)$$

$$K^0 = K^\infty + \sum_{n=1}^N K^n \quad (7-206)$$

Let the deviatoric and volumetric component matrices π_d and π_v be given by

$$\pi_d = \begin{bmatrix} 4/3 & -2/3 & -2/3 & 0 & 0 & 0 \\ -2/3 & 4/3 & -2/3 & 0 & 0 & 0 \\ -2/3 & -2/3 & 4/3 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 & 0 & 1 \end{bmatrix} \quad (7-207)$$

$$\pi_v = \begin{bmatrix} 1 & 1 & 1 & 0 & 0 & 0 \\ 1 & 1 & 1 & 0 & 0 & 0 \\ 1 & 1 & 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 \end{bmatrix}$$

The increment set of equations is then given by

$$\Delta\sigma(t_m) = \left\{ G^0 - \sum_{n=1}^{N_d} [1 - \beta_d^n(h)] G^n \right\} \pi_d \Delta\epsilon(t_m) - \sum_{n=1}^{N_d} \alpha_d^n(h) \sigma_d^n(t_m - h) - \left\{ K^0 - \sum_{n=1}^{N_v} [1 - \beta_v^n(h)] K^n \right\} \pi_v \Delta\epsilon(t_m) - \sum_{n=1}^{N_v} \alpha_v^n(h) \sigma_v^n(t_m - h) \quad (7-208)$$

and

$$\begin{aligned} \Delta\sigma_d^n(t_m) &= \beta_d^n(h) G^n \pi_d \Delta\epsilon(t_m) - \alpha_d^n(h) \sigma_d^n(t_m - h) \\ \Delta\sigma_v^n(t_m) &= \beta_v^n(h) K^n \pi_v \Delta\epsilon(t_m) - \alpha_v^n(h) \sigma_v^n(t_m - h) \end{aligned} \quad (7-209)$$

Note that the deviatoric and volumetric response are fully decoupled.

The instantaneous moduli need to be given in the **ISOTROPIC** option. Time dependent values (shear moduli G_n and time constants λ_d^n ; bulk moduli K^n and time constants λ_v^n) need to be entered in the **VISCELPROP** option.

Time-stepping is performed using the **TIME STEP** or **AUTO STEP** option in the history definition block.

Note that the algorithm is exact for linear variations of the strain during the increment. The algorithm is implicit; hence, for each change in time-step, a new assembly of the stiffness matrix is required.

Anisotropic Viscoelastic Material

Equation 7-199 can be used for the analysis of anisotropic viscoelastic materials. The tensor of amplitudes G_{ijkl}^0 must be entered through the

ORTHOTROPIC option. However, both the G_{ijkl}^n and the λ^n must be entered using the **VISCELORTH** option.

Alternatively, a complete set of moduli (21 components) can be specified in the [HOOKVI](#) user subroutine.

The [ORIENTATION](#) option or [ORIENT](#) user subroutine can be used to define a preferred orientation both for the short time moduli G_{ijkl}^0 and the amplitude functions G_{ijkl}^n .

Incompressible Isotropic Viscoelastic Materials

Incompressible elements in MSC.Marc allow the analysis of incompressible and nearly incompressible materials in plane strain, axisymmetric and three-dimensional problems. The incompressibility of the element is simulated through the use of an perturbed Lagrangian variational principle based on the Herrmann formulation.

The constitutive equation for a material with no time dependence in the volumetric behavior can be expressed as

$$\Delta\sigma_{ij}(t_m) = 2 \left\{ G_{ijkl}^0 - \sum_{n=1}^N [1 - \beta^n(h)] G_{ijkl}^n \right\} \left[\Delta\epsilon_{kl}(t_m) - \frac{1}{3} \Delta\epsilon_{pp}(t_m) \delta_{kl} \right] - \sum_{n=1}^N \alpha^n(h) (\sigma'_{ij})^n(t_m) + \frac{1}{3} \sigma_{kk} \delta_{ij} \quad (7-210)$$

$$\Delta\sigma_{pp}(t_m) = 3K^0 \Delta\epsilon_{pp}(t_m) \quad (7-211)$$

The hydrostatic pressure term is used as an independent variable in the variational principle. The Herrmann pressure variable is now defined in the same way as in the formulation for time independent elastic materials.

$$H = \frac{\sigma_{pp}}{2G^0(1 + \nu^0)} \quad (7-212)$$

The constitutive [Equation 7-210](#) and [Equation 7-211](#) can then be rewritten

$$\Delta\sigma_{ij}(t_m) = 2G^e(\Delta\epsilon_{ij} + \nu^* H \delta_{ij}) - \sum_{n=1}^N \alpha^n(h) (\sigma'_{ij})^n(t_m - h) \quad (7-213)$$

where

$$G^e = G^0 - \sum_{n=1}^N [1 - \beta^n(h) G^n] \quad (7-214)$$

$$\nu^* = \frac{G^0(1 + \nu^0) - G^e(1 - 2\nu^0)}{3G^e} \quad (7-215)$$

Thermo-Rheologically Simple Behavior

The rate processes in many viscoelastic materials are known to be highly sensitive to temperature changes. Such temperature-dependent properties cannot be neglected in the presence of any appreciable temperature variation. For example, there is a large class of polymers which are adequately represented by linear viscoelastic laws at uniform temperature. These polymers exhibit an approximate translational shift of all the characteristic response functions with a change of temperature, along a logarithmic time axis. This shift occurs without a change of shape. These temperature-sensitive viscoelastic materials are characterized as Thermo-Rheologically Simple.

A “reduced” or “pseudo” time can be defined for the materials of this type and for a given temperature field. This new parameter is a function of both time and space variables. The viscoelastic law has the same form as one at constant temperature in real time. If the shifted time is used, however, the transformed viscoelastic equilibrium and compatibility equations are not equivalent to the corresponding elastic equations.

In the case where the temperature varies with time, the extended constitutive law implies a nonlinear dependence of the instantaneous stress state at each material point of the body upon the entire local temperature history. In other words, the functionals are linear in the strains but nonlinear in the temperature.

The time scale of experimental data is extended for Thermo-Rheologically Simple materials. All characteristic functions of the material must obey the same property. The shift function is a basic property of the material and must be determined experimentally. As a consequence of the shifting of the mechanical properties data parallel to the time axis (see [Figure 7-52](#)), the values of the zero and infinite frequency complex moduli do not change due to shifting. Hence, elastic materials with temperature-dependent characteristics neither belong to nor are consistent with the above hypothesis for the class of Thermo-Rheologically Simple viscoelastic solids.

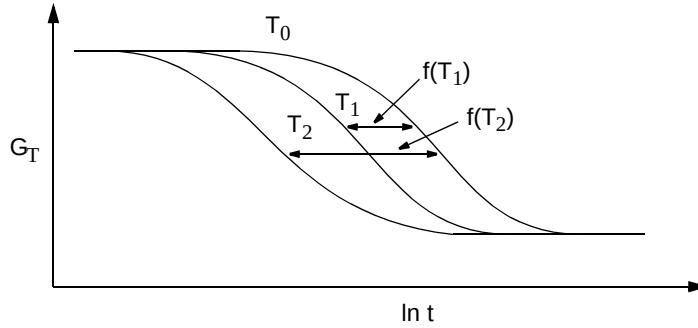


Figure 7-52 Relaxation Modulus vs. Time at Different Temperatures

Let $E(\ln t)$ be the relaxation modulus as a function of $\ln t$ at uniform temperature, T . Then

$$E_T(\ln t) = E_{T_0}[\ln t + f(T)] * (\rho T / \rho_0 T_0) \quad (7-216)$$

where $f(T)$ is measured relative to some arbitrary temperature T . The modulus curve shifts towards shorter times with an increase of temperature; $f(T)$ is a positive increasing function for $T > T_0$. If $G_T(t)$ denotes the relaxation modulus as a function of time at uniform temperature T , so that,

$$G_T(t) = E_T(\ln t) \quad (7-217)$$

then

$$G_T(t) = G_{T_0}(\xi) \quad (7-218)$$

The relaxation modulus (and the other characteristic functions) at an arbitrary uniform temperature is thus expressed by the base temperature behavior related to a new time scale that depends on that temperature.

There is some mapping of the time coordinate for nonuniform, nonconstant temperature, $T(x, t)$, which depends on the position

For a nonuniform, nonconstant temperature, the shift function is $a(T(x, t))$ and the rate of change of reduced time can be written as:

$$d\xi = a_T[T(x, t)] dt$$

MSC.Marc offers two explicit forms for entering the shift function. The first is based on the familiar Williams-Landel-Ferry (WLF) equation. Rewriting the above expression for reduced time as

$$\xi(x, t) = \int_0^t \frac{dt'}{a_T[T(x, t')]} \quad (7-219)$$

then the WLF form state that

$$\log_{10} a_T(T) = \frac{-C_1(T - T_0)}{C_2 + (T - T_0)} = -h(T) \quad (7-220)$$

and

$$\xi(t) = \int_0^t 10^{h[T(t')]} dt' \quad (7-221)$$

Typically, the glassy transition point is taken as the reference temperature in the above relation. The logarithmic shift can also be expressed in a polynomial expansion about the arbitrary reference point as

$$\log_{10} A_T(T) = \sum_{i=0}^m a_i (T - T_0)^i \quad (7-222)$$

Enter the shift function parameters associated with Thermo-Rheologically Simple behavior through the [SHIFT FUNCTION](#) model definition option. As an alternative to the WLF function, MSC.Marc allows use of series expansion or specification via the [TRSFAC](#) user subroutine.

In addition to the Thermo-Rheologically Simple material behavior variations of initial stress-strain moduli G_{ijkl}^0 , the temperature of the other mechanical properties (coefficient of thermal expansion, etc.) due to changes in temperature can be specified via the [TEMPERATURE EFFECTS](#) option.

Note, however, that only the instantaneous moduli are effected by the [TEMPERATURE EFFECTS](#) option. Hence, the long term moduli given by

$$G_{ijkl}^\infty = G_{ijkl}^0(t) - \sum_{n=1}^N G_{ijkl}^n \quad (7-223)$$

can easily become negative if the temperature effects are not defined properly.

Large Strain Viscoelasticity

For an elastomeric time independent material, the constitutive equation is expressed in terms of an energy function W . For a large strain viscoelastic material, Simo generalized the small strain viscoelasticity material behavior to a large strain viscoelastic material. The energy functional then becomes

$$\psi(E_{ij}, Q_{ij}^n) = \psi^0(E_{ij}) - \sum_{n=1}^N Q_{ij}^n E_{ij} + \sum_{n=1}^N \psi_I^n(Q_{ij}^n) \quad (7-224)$$

where E_{ij} are the components of the Green-Lagrange strain tensor, Q_{ij}^n internal variables and ψ^0 the elastic strain energy density for instantaneous deformations. In MSC.Marc, it is assumed that $\psi^0 = W$, meaning that the energy density for instantaneous deformations is given by the third order James Green and Simpson form or the Ogden form. When used with Updated Lagrange, one can also use the Arruda-Boyce or Gent model.

The components of the second Piola-Kirchhoff stress then follow from

$$S_{ij} = \frac{\partial \psi}{\partial E_{ij}} = \frac{\partial \psi^0}{\partial E_{ij}} - \sum_{n=1}^N Q_{ij}^n \quad (7-225)$$

The energy function can also be written in terms of the long term moduli resulting in a different set of internal variables T_{ij}^n

$$\psi(E_{ij}, T_{ij}^n) = \psi^\infty(E_{ij}) + \sum_{n=1}^N T_{ij}^n E_{ij} \quad (7-226)$$

where ψ^∞ is the elastic strain energy for long term deformations. Using this energy definition, the stresses are obtained from

$$S_{ij} = \frac{\partial \psi^\infty(E)}{\partial E_{ij}} + \sum_{n=1}^N T_{ij}^n \quad (7-227)$$

Observing the similarity with the equations for small strain viscoelasticity the internal variables can be obtained from a convolution expression

$$T_{ij}^n = \int_0^t \dot{S}_{ij}^n(\tau) \exp[-(t-\tau)/\lambda^n] d\tau \quad (7-228)$$

where S_{ij}^n are internal stresses obtained from energy functions.

$$S_{ij}^n = \frac{\partial \Psi^n}{\partial E_{ij}} \quad (7-229)$$

Let the total strain energy be expressed as a Prony series expansion

$$\Psi = \Psi^\infty + \sum_{n=1}^N \Psi^n \exp(-t/\lambda^n) \quad (7-230)$$

If, in the energy function, each term in the series expansion has a similar form, [Equation 7-220](#) can be rewritten

$$\Psi = \Psi^\infty + \sum_{n=1}^N \delta^n \Psi^0 \exp(-t/\lambda^n) \quad (7-231)$$

where δ^n is a scalar multiplier for the energy function based on the short term values.

The stress-strain relation is now given by

$$S_{ij}(t) = S_{ij}^\infty(t) + \sum_{n=1}^N T_{ij}^n(t) \quad (7-232)$$

$$S_{ij} = \frac{\partial \Psi^\infty}{\partial E_{ij}} = \left(1 - \sum_{n=1}^N \delta^n \right) \frac{\partial \Psi^0}{\partial E_{ij}} \quad (7-233)$$

$$T_{ij}^n = \int_0^t \delta^n S_{ij}^0(t) \exp[-(t-\tau)/\lambda^n] d\tau \quad (7-234)$$

Analogue to the derivation for small strain viscoelasticity, a recursive relation can be derived expressing the stress increment in terms of values of the internal stresses at the start of the increment.

In MSC.Marc, the instantaneous values of the energy function are always given on the **MOONEY** or **OGDEN** option, the equations are reformulated in terms of the short time values of the energy function

$$\Delta S_{ij}(t_m) = \left(1 - \sum_{n=1}^N 1 - \beta^n(h) \delta^n \right) \{ S_{ij}^0(t_m) - S_{ij}^0(t_m - h) \} - \sum_{n=1}^N \alpha^n S_{ij}^n(t_m - h) \quad (7-235)$$

$$\Delta S_{ij}(t_m) = \beta^n(h) \delta^n [S_{ij}^0(t_m) - S_{ij}^n(t_m - h)] - \alpha^n(h) S_{ij}^n(t_m - h) \quad (7-236)$$

It is assumed that the viscoelastic behavior in MSC.Marc acts only on the deviatoric behavior. The incompressible behavior is taken into account using special Herrmann elements.

Large strain viscoelasticity is invoked by use of the **VISCELMOON** or **VISCELOGDEN** option in the constitutive option of the model definition block.

The time dependent multipliers δ^n and associated relaxation times λ^n as defined by Equation 7-221 are given in the **VISCELMOON** or **VISCELOGDEN** option. For the Ogden model, both deviatoric and dilatational relaxation behavior is allowed.

Viscoelasticity can be modeled with Arruda-Boyce and Gent models using **VISCELMOON** option.

Time-stepping can be performed using the **TIME STEP** with **AUTO LOAD** or **AUTO STEP** option of the history definition block.

The free energy function versus time data being used for large strain viscoelasticity can be generated by fitting the experimental data in MSC.Marc Mentat provided the following two tests are done:

1. Standard quasi-static tests (tensile, planar-shear, simple-shear, equi-biaxial tension, volumetric) to determine the elastomer free energy W^0 constants.
2. Standard relaxation tests to obtain stress versus time.

Narayanaswamy Model

The annealing of flat glass requires that the residual stresses be of an acceptable magnitude, while the specification for optical glass components usually includes a homogenous refractive index. The design of heat treated processes (for example, annealing) can be accomplished using the Narayanaswamy model. This allows you to study the time dependence of physical properties (for example, volumes) of glass subjected to a change in temperature.

The glass transition is a region of temperature in which molecular rearrangements occur on a scale of minutes or hours, so that the properties of a liquid change at a rate that is easily observed. Below the glass transition temperature T_g , the material is extremely viscous and a solidus state exists. Above T_g , the equilibrium structure is arrived at easily and the material is in liquidus state. Hence, the glass transition is revealed by a change in the temperature dependence of some property of a liquid during cooling. If a mechanical stress is applied to a liquid in the transition region, a time-dependent change in dimensions results due to the phenomenon of visco-elasticity.

If a liquid in the transition region is subjected to a sudden change in temperature, a time-dependent change in volume occurs as shown in [Figure 7-53](#). The latter process is called structural relaxation. Hence, structural relaxation governs the time-dependent response of a liquid to a change of temperature.

Suppose a glass is equilibrated at temperature T_1 , and suddenly cooled to T_2 at t_0 . The instantaneous change in volume is $\alpha_g(T_2 - T_1)$, followed by relaxation towards the equilibrium value $V(\infty, T_2)$. The total change in volume due to the temperature change is $\alpha_1(T_2 - T_1)$ as shown in [Figure 7-53b](#). The rate of volume change depends on a characteristic time called the relaxation time.

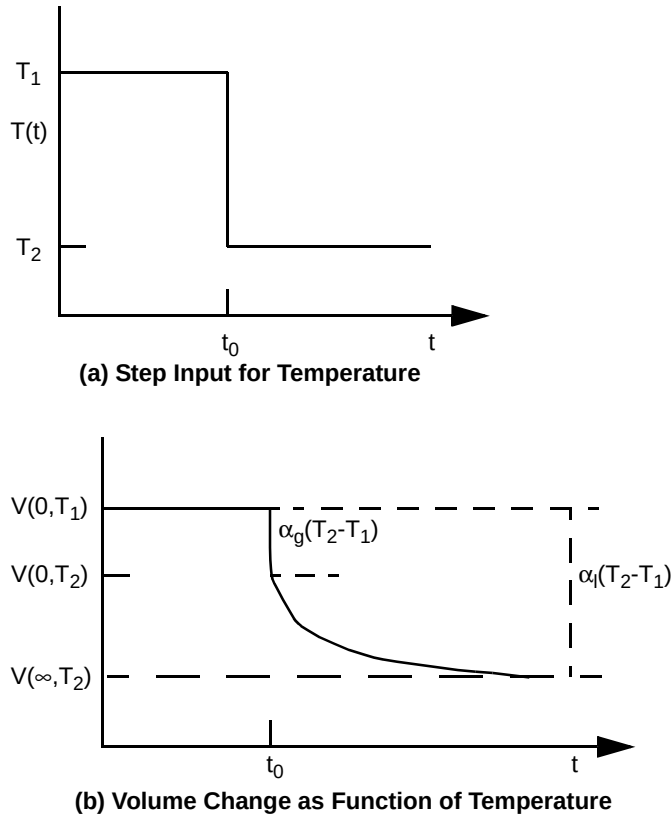


Figure 7-53 Structural Relaxation Phenomenon

The slope of dV/dT changes from the high value characteristic of the fluid α_1 to the low characteristic of the glass α_g as shown in [Figure 7-54](#). The glass transition temperature T_g is a point in the center of the transition region. The low-temperature slope α_g represents the change in volume V caused by vibration of the atoms in their potential wells. In the (glassy) temperature range, the atoms are frozen into a particular configuration. As the temperature T increases, the atoms acquire enough energy to break bonds and rearrange into new structures. That allows the volume to increase more rapidly, so $\alpha_1 > \alpha_g$. The difference $\alpha = \alpha_1 - \alpha_g$ represents the structural contribution to the volume.

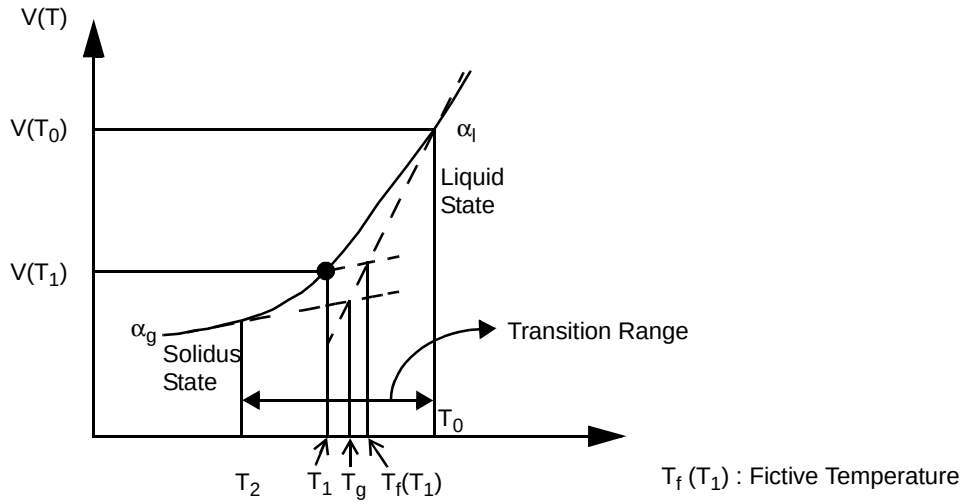


Figure 7-54 Property (Volume) – Temperature Plot

When a liquid is cooled and reheated, a hysteresis is observed as shown in [Figure 7-55](#).

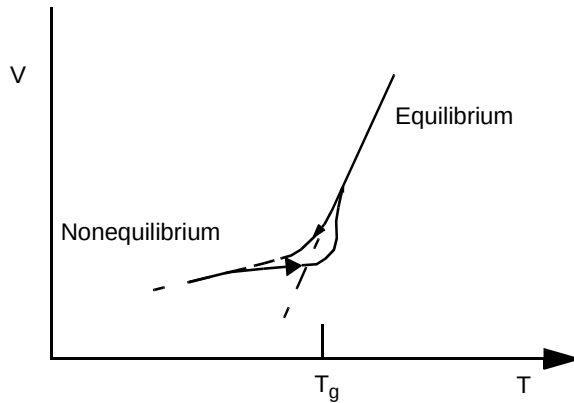


Figure 7-55 Volume Change During Cyclic Temperature History

Unfortunately, the notion of a glass transition temperature is insufficient as real glassy materials generally exhibit a temperature regime, called a transition range, across which their bulk properties gradually change from being solid-like to liquid-like in nature.

As discussed earlier, properties have a time dependence in the transition range. An explanation for the strong time dependence lies in that the material resides at a nonequilibrium temperature which lags behind the applied temperature during the heating-cooling cycle. The nonequilibrium temperature is called the fictive temperature, T_f , as shown in Figure 7-54. The fictive temperature at T_1 , $T_f(T_1)$ is found by extrapolating a line from $V(T_1)$ with slope α_g to intersect a line extrapolated from $V(T_0)$ with slope α_l (see Figure 7-54). For $T \leq T_2$ (well below the glass transition), T_f reaches a limiting value that is called T_g . If the material were equilibrated at $T_f(T_1)$, then instantaneously cooled to T_1 , it would change along the line with slope α_g because no structural rearrangement could occur. Therefore, it would have the same volume as the continuously cooled sample.

The response of the volume change can be described by:

$$V(T_2, t) = V(T_1, \infty) + \alpha_g(T_2 - T_1) + (\alpha_l - \alpha_g)(T_f(t) - T_2) \quad (7-237)$$

where $T_f(t)$ is the current value of the fictive temperature. The response function, M_v , which dictates the value of the fictive temperature is assumed to be linear in its argument and governs both the value of the fictive temperature as well as the material property of interest.

$$\frac{V(t) - V(\infty)}{V(0) - V(\infty)} = M_v(\xi(t)) = \frac{T_f(t) - T(\infty)}{T(0) - T(\infty)} \quad (7-238)$$

By virtue of its linearity, Boltzmann's superposition principle can be invoked to calculate the fictive temperature at any time:

$$T_f(t) = T(t) - \int_{-\infty}^t M_v(\xi(t) - \xi(t')) \frac{d}{dt'} (T(t')) (dt') \quad (7-239)$$

The concept of reduced time, $\xi(t)$, is introduced in the spirit of Thermo-Rheologically Simple materials to capture the disparate nonlinear response curves on a single master curve.

The reduced time used in MSC.Marc is given by the following expression:

$$\xi(t) = \int_{-\infty}^t \frac{\tau_{ref}}{\tau(T(t'))} dt' \quad (7-240)$$

Here τ_{ref} is the reference relaxation time of the material evaluated at a suitable reference temperature, T_{ref} . The relaxation time τ at the given time and temperature can be represented as:

$$\tau = \tau_{ref} \cdot \exp\left(-\frac{H}{R}\left[\frac{1}{T_{ref}} - \frac{x}{T} - \frac{(1-x)}{T_f}\right]\right) \quad (7-241)$$

The parameter x allows you to dictate how much of the fictive temperature participates in the prescription of the relaxation time, and must, therefore, range between 0 and 1.

H is the activation energy for the particular process and R is the gas constant. A typical response function is:

$$M_v(\xi) = \exp\left(-\frac{\xi}{\tau}\right) \quad (7-242)$$

Multiple structural relaxation times can exist. The response function has, therefore, been implemented as:

$$M_v(\xi) = \sum_{i=1}^n (W_g)_i \cdot \exp\left(-\frac{\xi}{\tau_i}\right)$$

For a complete description of the model, it is necessary to prescribe the following:

1. The weight $(W_g)_i$ for each term in the series (usually $\sum (W_g)_i \approx 1$).
2. The reference relaxation times $\tau_{i, ref}$.
3. The fraction parameter x and the activation energy-gas constant ratio.
4. The solid and liquid coefficients of thermal of expansion, α_g and α_l through the **VISCEL EXP** option.

A stable algorithm is employed to calculate the convolution integrals. For improved accuracy it is recommended that the time steps used during the simulation be sufficiently small.

In [Figure 7-56](#), the volume of cube of material, which is allowed to contract freely and is experiencing a 100°C quench, is displayed.

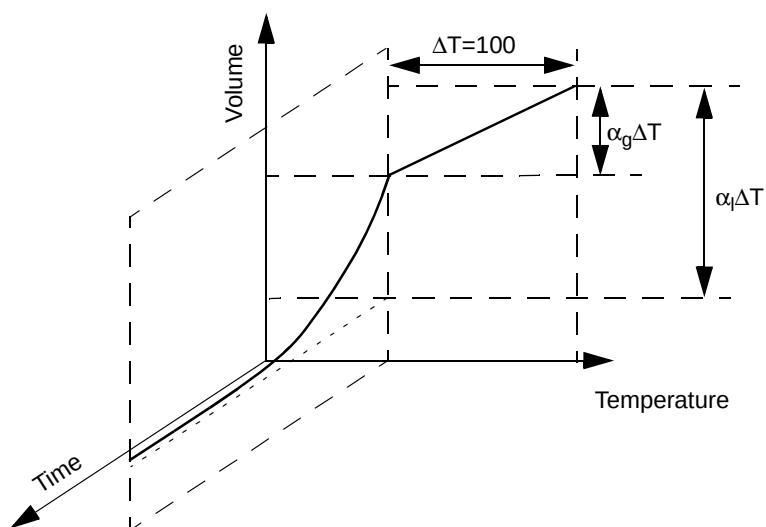


Figure 7-56 Volume-Temperature-Time Plot

Temperature Effects and Coefficient of Thermal Expansion

Experimental results indicate that a large number of material properties vary with temperatures. MSC.Marc accepts the temperature-dependent material properties shown in [Table 7-2](#) in the form of piecewise linear functions for different types of analysis.

Table 7-2 Temperature-Dependent Material Properties

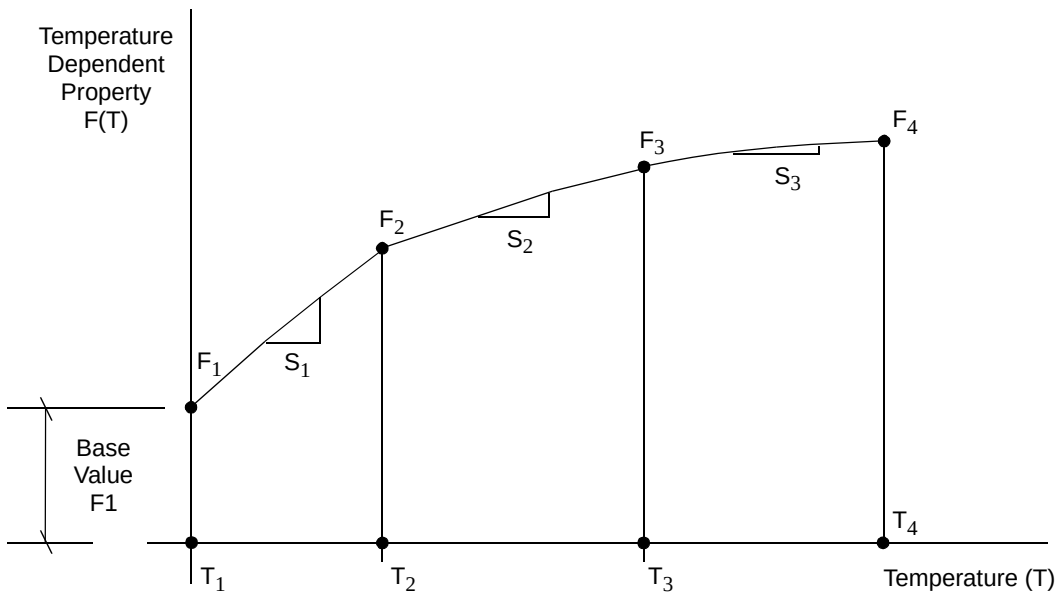
Analysis Type	Material Properties	
Stress Analysis	Modulus of elasticity (Young's Modulus)	$E(T)$
	Poisson's Ratio	$\nu(T)$
	Yield Stress	$\sigma_y(T)$
	Workhardening Slope	$h(T)$
	Coefficient of Thermal Expansion	$\alpha(T)$
	Mooney Constants	$C_{01}(T), C_{10}(T)$ $C_{11}(T), C_{20}(T)$ $C_{30}(T)$
Heat Transfer Analysis	Thermal Conductivity	$K(T)$
	Specific Heat	$C(T)$
	Emissivity	$\varepsilon(T)$
Couples Thermo-Electrical (Joule Heating Analysis)	Electric Resistivity	$\pi(T)$
Hydrodynamic Heating	Viscosity	$\mu(T)$

Please note that T is temperature in the above expressions. With the exception of heat transfer or Joule heating, the temperature is a state variable.

Piecewise Linear Representation

In MSC.Marc, the temperature variation of a material constant $F(T)$ is assumed to be a piecewise linear function of temperature. [Figure 7-57](#) illustrates this function. Input the piecewise linear function of temperature using the [ISOTROPIC](#), [ORTHOTROPIC](#), [TEMPERATURE EFFECTS](#), and [ORTHO TEMP](#) model definition options. Enter the base value or value at reference temperature (the lowest temperature that will occur during

your analysis), through the **ISOTROPIC**, and **ORTHOTROPIC** model definition options. Input either the slope-breakpoint data or the function-temperature data through the **TEMPERATURE EFFECTS** and **ORTHO TEMP** model definition options.



(1) Slope-Break Point Data

<u>Slope</u>	<u>Break Point</u>
$S_1 = (F_2 - F_1)/(T_2 - T_1)$	T_1
$S_2 = (F_3 - F_2)/(T_3 - T_2)$	T_2
$S_3 = (F_4 - F_3)/(T_4 - T_3)$	T_3

(2) Function-Variable Data

<u>Function</u>	<u>Variable</u>
F_1	T_1
F_2	T_2
F_3	T_3
F_4	T_4

Figure 7-57 Piecewise Linear Representation of Temperature-Dependent Material Properties

Temperature-Dependent Creep

In MSC.Marc, the temperature dependency of creep strain can be entered in two ways. The creep strain rate may be entered as a piecewise linear function. If the creep strain $\dot{\epsilon}^c$ can be expressed in the form of a power law

$$\dot{\epsilon}^c = AT^m \quad (7-243)$$

where A and m are two experimental constants, input the experimental constants through the [CREEP](#) model definition option.

For other temperature dependency, you must use the [CRPLAW](#) user subroutine for explicit creep and [UCRPLW](#) user subroutine for implicit creep to input the variation of creep strain with temperature.

Coefficient of Thermal Expansion

MSC.Marc always uses an instantaneous thermal expansion coefficient definition

$$d\epsilon_{ij}^{th} = \alpha_{ij}dT \quad \text{in general} \quad (7-244)$$

or

$$d\epsilon_{ij}^{th} = \alpha dT \delta_{ij} \quad \text{for the isotropic case} \quad (7-245)$$

In many cases, the thermal expansion data is given with respect to a reference temperature

$$\epsilon^{th} = \bar{\alpha}(T - T^0) \quad (7-246)$$

where α is a function of temperature:

$$\bar{\alpha} = \bar{\alpha}(T) \quad (7-247)$$

Clearly, in this case

$$d\epsilon^{th} = \left[\bar{\alpha} + \frac{d\bar{\alpha}}{dT}(T - T^0) \right] dT \quad (7-248)$$

so the necessary conversion procedure is:

1. Compute and plot Equation 7-246 in the form Equation 7-248

$$\alpha = \bar{\alpha} + \frac{d\bar{\alpha}}{dT}(T - T^0) \quad (7-249)$$

as a function of temperature.

2. Model Equation 7-249 in user subroutine ANEXP, or with piecewise linear slopes and breakpoints in the TEMPERATURE EFFECTS option.

The anisotropic coefficient of thermal expansion can be input through either the ORTHOTROPIC model definition option or the ANEXP user subroutine.

Time-Temperature-Transformation

Certain materials, such as carbon steel, exhibit a change in mechanical or thermal properties when quenched or air cooled from a sufficiently high temperature. At any stage during the cooling process, these properties are dependent on both the current temperature and the previous thermal history. The properties are influenced by the internal microstructure of the material, which in turn depends on the rate at which the temperature changes. Only in instances where the temperature is changed very gradually does the material respond in equilibrium, where properties are simply a function of the current temperature. In addition, during the cooling process certain solid-solid phase transformations can occur. These transformations represent another form of change in the material microstructure which can influence the mechanical or thermal properties. These transformations can be accompanied by changes in volume.

The occurrence of phase change is also dependent on the rate of cooling of the material. This relationship is shown in a typical cooling diagram (see [Figure 7-58](#)). The curves A, B, and C in [Figure 7-58](#) represent the temperature history of a structure that has been subjected to a different cooling rate. It is obvious that the structural material experiences phase changes at different times and temperatures, depending on the rate of cooling. Under cooling rate A, the material changes from phase 1 to phase 4 directly. The material undergoes three phase changes (phase 1 to phase 2 to phase 3 to phase 4) for both cooling rates B and C. However, the phase changes take place at different times and temperatures.

The Time-Temperature-Transformation ([TIME-TEMP](#)) option allows you to account for the time-temperature-transformation interrelationships of certain materials during quenching or casting analyses.

Use the [T-T-T](#) parameter to invoke the time-temperature-transformation. Input all the numerical data required for this option through the [TIME-TEMP](#) model definition option.

In a transient heat transfer analysis, the thermal properties which can be defined as a function of time and temperature are the thermal conductivity and the specific heat per unit reference mass. Here, the effects of latent heat or phase transformation can be included through the definition of the specific heat.

In a thermal stress analysis, the mechanical properties which can be defined as a function of time and temperature are the Young's modulus, Poisson's ratio, yield stress, workhardening slope, and coefficient of thermal expansion. The effects of volumetric change due to phase transformation can be included through the definition of the coefficient of thermal expansion.

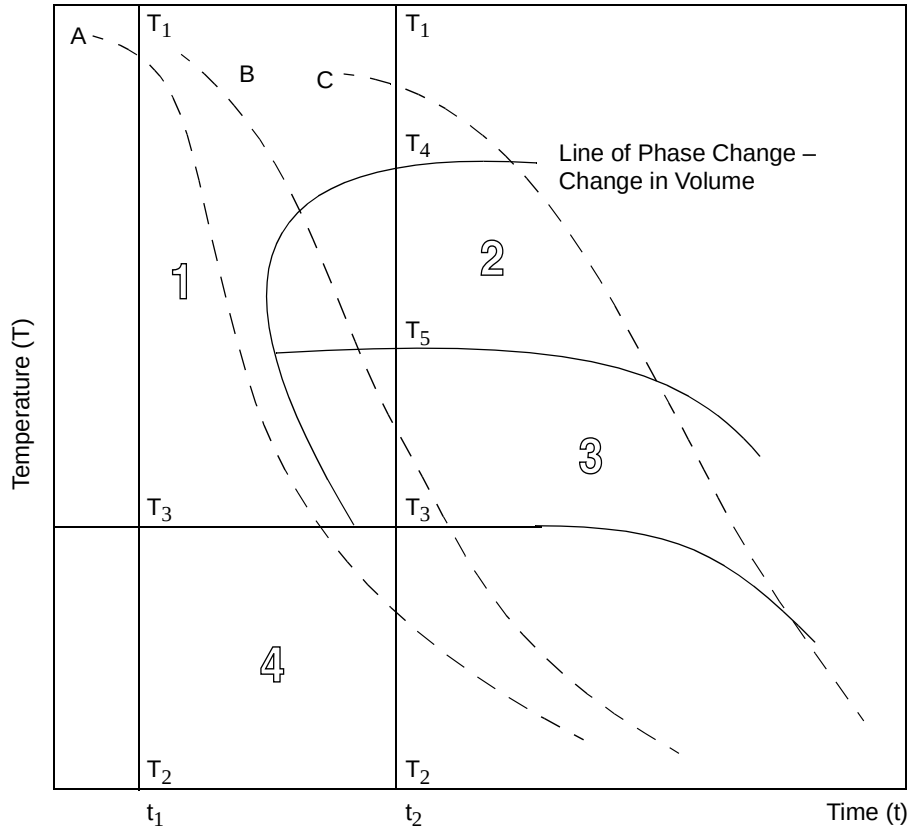


Figure 7-58 Simplified Cooling Transformation Diagram

Test data must be available in a tabular form for each property of each material group. For a given cooling rate, the value of a property must be known at discrete points over a range of temperatures. There can be several sets of these discrete points corresponding to measurements at several different cooling rates. The cooling tests must be of a specific type known as Newton Cooling; that is, the temperature change in the material is controlled such that

$$T(t) = A \exp(-at) + B \quad (7-250)$$

In addition, a minimum and a maximum temperature that bracket the range over which the **TIME-TEMP** option is meant to apply must also be given.

For the simulation of the cooling rate effect in finite element analysis, material properties of a structure can be assumed as a function of two variables: time and temperature. Two-dimensional interpolation schemes are used for the interpolation of properties.

Interpolation is based on making the time variable discrete. Stress analysis is carried out incrementally at discrete time stations and material properties are assumed to vary piecewise linearly with temperature at any given time. These temperature-dependent material properties are updated at each increment in the analysis. For illustration, at time t_1 , the material is characterized by the phase 1 and phase 4 behaviors at temperature ranges T_1 to T_3 , and T_3 to T_2 , respectively (see Figure 7-59). Similarly, at time t_2 , the material behavior must be characterized by all four phases, each in a different temperature range (that is, phase 1, T_1 to T_4 ; phase 2, T_4 to T_5 ; phase 3, T_5 to T_3 ; phase 4, T_3 to T_2). The selection of an interpolation scheme is generally dependent on the form of the experimental data. A linear interpolation procedure can be effectively used where the properties are expressed as a tabulated function of time and temperature.

During time-temperature-transformation, the change in volume in a stress analysis is assumed to take place in a temperature range ΔT . The change in volume is also assumed to be uniform in space, such that the effect of the volume change can be represented by a modification of the coefficient of thermal expansion. For a triangular distribution of $\alpha(T)$ in the temperature range ΔT , the value of the modified coefficient of thermal expansion is

$$\alpha_m = \frac{2}{\Delta T} [\sqrt[3]{1 + \phi} - 1] \quad (7-251)$$

where ϕ is the change in volume. A schematic of the modified $\alpha(T)$ is shown in Figure 7-59.

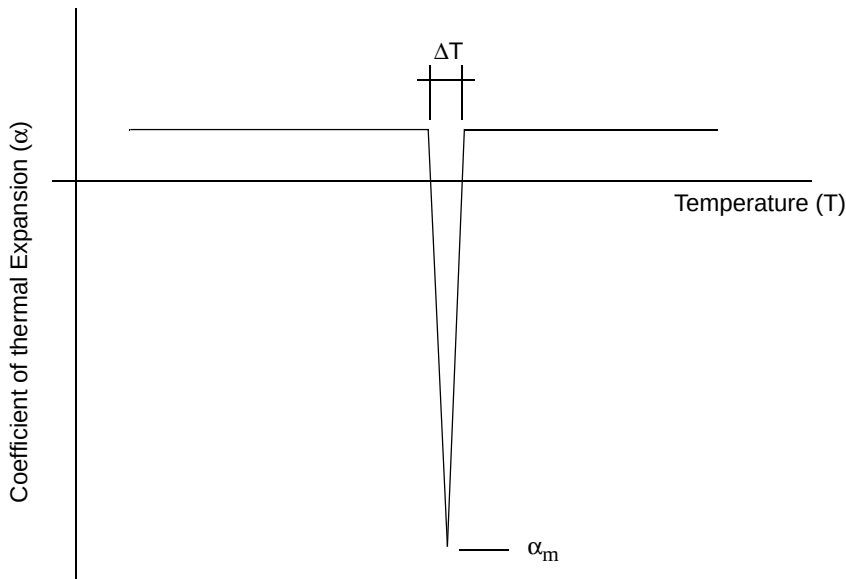


Figure 7-59 Modified Coefficient of Thermal Expansion for Short-Time Change in Volume

Low Tension Material

MSC.Marc can handle concrete and other low tension material. The [CRACK DATA](#) option assists in predicting crack initiation and in simulating tension softening, plastic yielding and crushing. This option can be used for the following:

- Elements with a one-dimensional stress-strain relation (beam and truss elements)
- Elements with a two-dimensional stress-strain relation (plane stress, plane strain, and shell elements)
- Three-dimensional elements (bricks)

Analytical procedures that accurately determine stress and deformation states in concrete structures are complicated by several factors. Two such factors are the following:

- The low strength of concrete in tension that results in progressive cracking under increasing loads
- The nonlinear load-deformation response of concrete under multiaxial compression

Because concrete is mostly used in conjunction with steel reinforcement, an accurate analysis requires consideration of the components forming the composite structure. Steel reinforcement bars are introduced as rebar elements.

Each rebar element must be input with a separate element number. The [REBAR](#) model definition option or [REBAR](#) user subroutine is used to define the orientation of the reinforcement rods.

Uniaxial Cracking Data

The cracking option is accessed through the [ISOTROPIC](#) option. Uniaxial cracking data can be specified using the [CRACK DATA](#) option or the [UCRACK](#) user subroutine. When the [CRACK DATA](#) option is used to specify uniaxial cracking data, the following must be specified: the critical cracking stress, the modulus of the linear strain softening behavior, and the strain at which crushing occurs. Material properties, such as Young's modulus and Poisson's ratio, are entered using the [ISOTROPIC](#) option and the [WORK HARD](#) option. This model is for a material which is initially isotropic; if the model is initially orthotropic, see [FAIL DATA](#) for an alternative cracking model. A typical uniaxial stress-strain diagram is shown in [Figure 7-60](#).

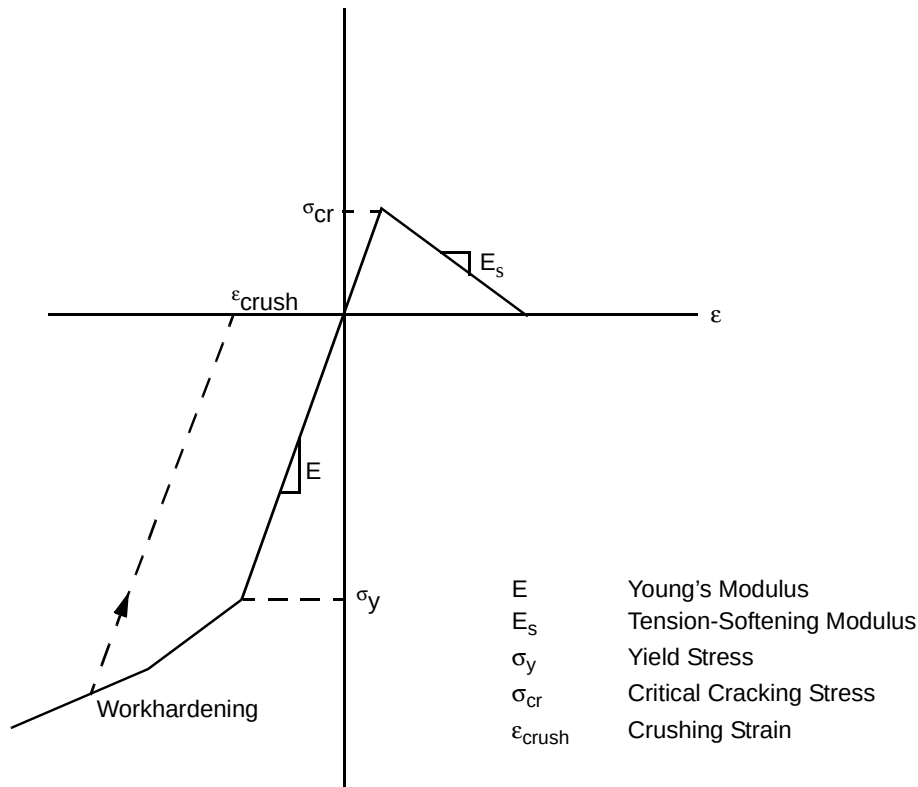


Figure 7-60 Uniaxial Stress-Strain Diagram

Low Tension Cracking

A crack develops in a material perpendicular to the direction of the maximum principal stress if the maximum principal stress in the material exceeds a certain value (see [Figure 7-61](#)). After an initial crack has formed at a material point, a second crack can form perpendicular to the first. Likewise, a third crack can form perpendicular to the first two. The material loses all load-carrying capacity across the crack unless tension softening is included.

Tension Softening

If tension softening is included, the stress in the direction of maximum stress does not go immediately to zero; instead the material softens until there is no stress across the crack. At this point, no load-carrying capacity exists in tension (see [Figure 7-61](#)). The softening behavior is characterized by a descending branch in the tensile stress-strain diagram, and it may be dependent upon the element size.

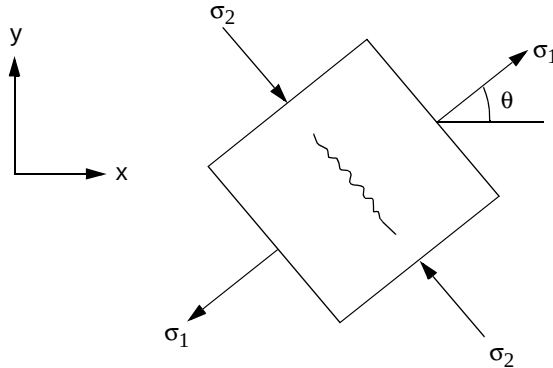


Figure 7-61 Crack Development

Crack Closure

After a crack forms, the loading can be reversed; therefore, the opening distance of a crack must be considered. In this case, the crack can close again, and partial mending occurs. When mending occurs, it is assumed that the crack has full compressive stress-carrying capability and that shear stresses are transmitted over the crack surface, but with a reduced shear modulus.

Crushing

As the compressive stress level increases, the material eventually loses its integrity, and all load-carrying capability is lost; this is referred to as crushing. Crushing behavior is best described in a multiaxial stress state by a crushing surface having the same shape as the yield surface. The failure criterion can be used for a two-dimensional stress state with reasonable accuracy. For many materials, experiments indicate that the crushing surface is roughly three times larger than the initial yield surface.

Analysis

The evolution of cracks in a structure results in the reduction of the load carrying capacity. The internal stresses need to be redistributed through regions that have not failed. This is a highly nonlinear problem and can result in the ultimate failure of the structure. The [AUTO INCREMENT](#) or [AUTO STEP](#) option should be used to control the applied load on the structure.

Soil Model

Soil material modeling is considerably more difficult than conventional metals, because of the nonhomogeneous characteristics of soil materials. Soil material usually consists of a large amount of random particles. Soils show unique properties when tested. The bulk modulus of soil increases upon pressing. Also, when the preconsolidation stress is exceeded, the stiffness reduces dramatically while the stiffness increases upon unloading. At failure, there is no resistance to shear, and stiff clays or dense sands are dilatant. Over the years, many formulations have been used, including linear elastic, nonlinear elastic, Drucker-Prager or Mohr Coulomb, and Cam-Clay and variations thereof. In MSC.Marc, the material models for available soil modeling are linear elasticity, nonlinear elasticity, and the Cam-Clay model.

Elastic Models

Linear elasticity is defined in the conventional manner, defining the Young's moduli and the Poisson's ratio in the **SOIL** option. The nonlinear elasticity model is implemented through the **HYPELA** or **HYPELA2** user subroutine. Some of the simplest models include the bilinear elasticity, where a different moduli is used during the loading and unloading path, or to represent total failure when a critical stress is obtained.

A more sophisticated elastic law is the hyperbolic model, where six constants are used. In this model, the tangent moduli are

$$E = \left[1 - \frac{R_f(1 - \sin\phi)(\sigma_1 - \sigma_3)}{2c \cos\phi + 2\sigma_3 \sin\phi} \right]^2 \left(\frac{\sigma_3}{pa} \right)^n \quad (7-252)$$

Two of the elastic models are the E-v and K-G variable elastic models. In the E-v model, Poisson's ratio is considered constant and

$$E = E_0 + \alpha_E p + \beta_E \tau \quad (7-253)$$

while, in the K-G model

$$K = K_0 + \alpha_K p \quad (7-254)$$

$$G = G_0 + \alpha_G p + \beta_G \tau \quad (7-255)$$

The key difficulty with the elastic models is that dilatancy cannot be represented.

Cam-Clay Model

The Cam-Clay model was originally developed by Roscoe, and then evolved into the modified Cam-Clay model of Roscoe and Burland. This model, which is also called the critical state model, is implemented in MSC.Marc.

The yield surface is an ellipse in the p, τ plane as shown in [Figure 7-62](#), and is defined by

$$F = \frac{\tau^2}{M_{cs}^2} - 2pp_c + p^2 = 0 \quad (7-256)$$

where p_c is the preconsolidation pressure, and M_{cs} is the slope of the critical state line.

The Cam-Clay model has the following no properties. At the intersection of the critical state line and the ellipse, the normal to the ellipse is vertical. Because an associated flow rule is used, all plastic strain at failure is distortional; the soil deforms at constant volume ([Figure 7-62](#)). The strain hardening and softening behavior are shown in [Figure 7-63](#). Also, if the preconsolidation pressure is large, the soil remains elastic for large stresses. The evolution of the preconsolidation pressure is

$$\dot{p}_c = -\theta p_c tr(\dot{\epsilon}^{pl}) \quad (7-257)$$

where

$$\theta = \frac{1+e}{\lambda-\kappa} \quad (7-258)$$

where

e is the void ratio

λ is the virgin compression index (see [Figure 7-64](#))

κ is the recompression index (see [Figure 7-64](#))

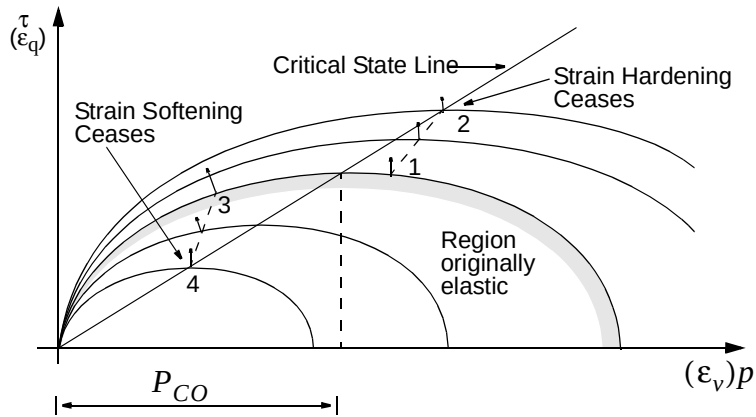


Figure 7-62 Modified Cam-Clay Yield Surface

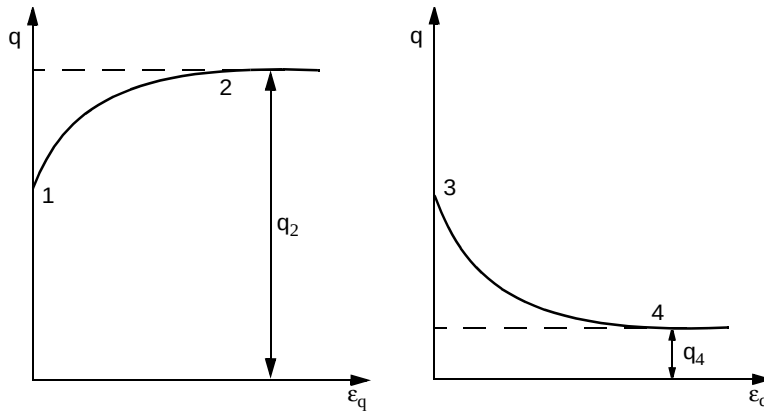


Figure 7-63 Strain Hardening and Softening Behavior

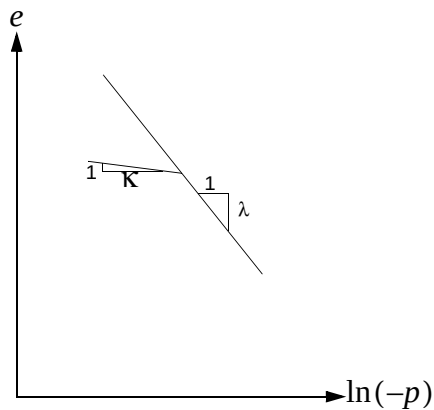


Figure 7-64 Response of Idealized Soil to Hydrostatic Pressure

The void ratio and the porosity are related by the expression

$$\phi = \frac{e}{1 + e} \quad (7-259)$$

In the modified Cam-Clay model, it is also assumed that the behavior is nonlinear elastic with a constant Poisson's ratio, the bulk modulus behave as:

$$K = -\frac{1 + e}{\kappa} p \quad (7-260)$$

Note that this implies that at zero hydrostatic stress, the bulk modulus is also zero. To avoid computational difficulties, a cutoff pressure of one percent of the preconsolidation pressure is used.

This constitutive law is implemented in MSC.Marc using a radial return procedure. It is available for either small or large strain analysis. When large displacements are anticipated, you should use the [UPDATE](#) parameter.

The necessary parameters for the Cam-Clay soil model can be obtained by the following experiments:

1. Hydrostatic test: determines volumetric elastic bulk modulus, yield stress, and the virgin and recompression ratio of soils.
2. Shear-box test: determines the slope of critical state line and shear modulus of the soil. However, the tests have to be calibrated with numerical simulations to get the necessary constants.
3. Triaxial shear test: the most comprehensive experimental information and obviates the need for the first two tests and obtains all the necessary constants listed above.

Evaluation of Soil Parameters for the Critical State Soil Model

To illustrate how to extract soil parameters, namely M , κ , and λ , a hypothetical data set for a normally consolidated clay is presented in Figures 7-65 to 7-71. Now we shall use these data to show the procedure of determination of parameters for the critical state model. The data presented here pertain to conventional triaxial conditions.

Figures 7-65 and 7-66 show the stress-strain relations for constant pressure tests under different initial conditions $p_0=10$ (69), 20 (139), and 30 (207) psi (kPa), respectively. Here the mean effective pressure of the soil sample is kept constant throughout the

test. The last data point shown on the deviatoric stress (q) versus axial strain (ϵ_1) plot for any test is considered as the ultimate condition for that test. The void ratio values at the beginning and the end of each test are given in Table 7-3.

Table 7-3 Test Values for Cam Clay Model

Test	Initial Pressure, p_0 (effective) (psi)	Initial void Ratio, e_0	Final Pressure, p_f (effective) (psi)	Final Void Ratio, e_1
p -constant	10	1.080	10	0.980
p -constant	20	0.959	20	0.860
p -constant	30	0.889	30	0.787
Drained	10	1.080	15	0.908
Drained	20	0.959	30	0.787
Drained	30	0.889	45	0.716
Undrained	10	1.080	5.55	1.080
Undrained	20	0.959	11.09	0.959
Undrained	30	0.889	16.64	0.889

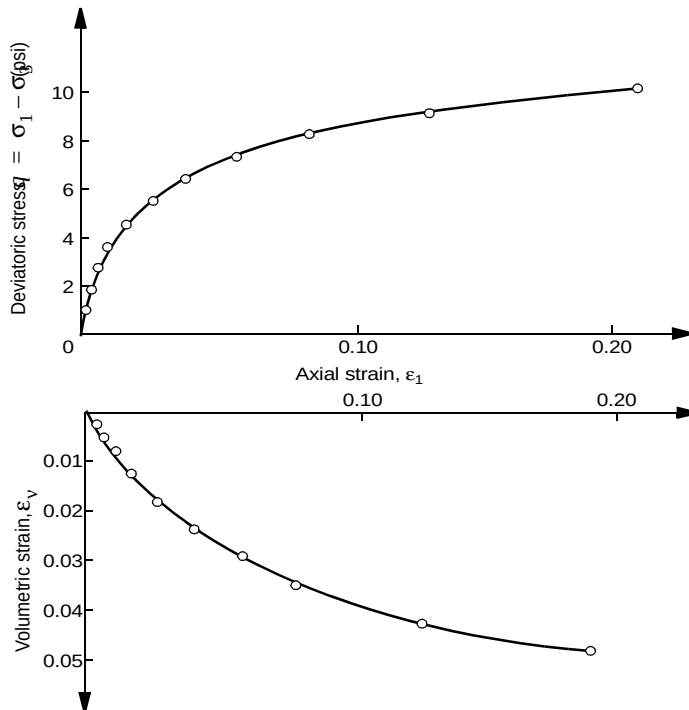


Figure 7-65 Constant Pressure Test: $p_0 = 10$ psi

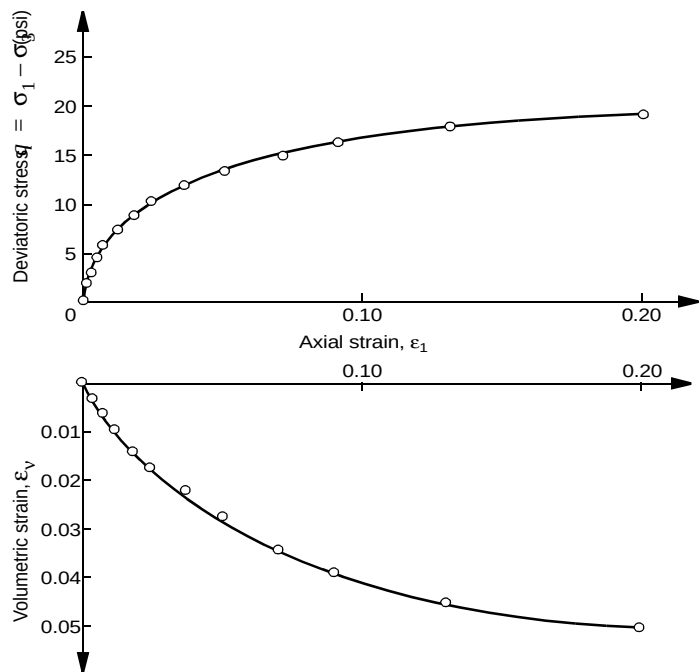


Figure 7-66 Constant Pressure Test: $p_0 = 20$ psi

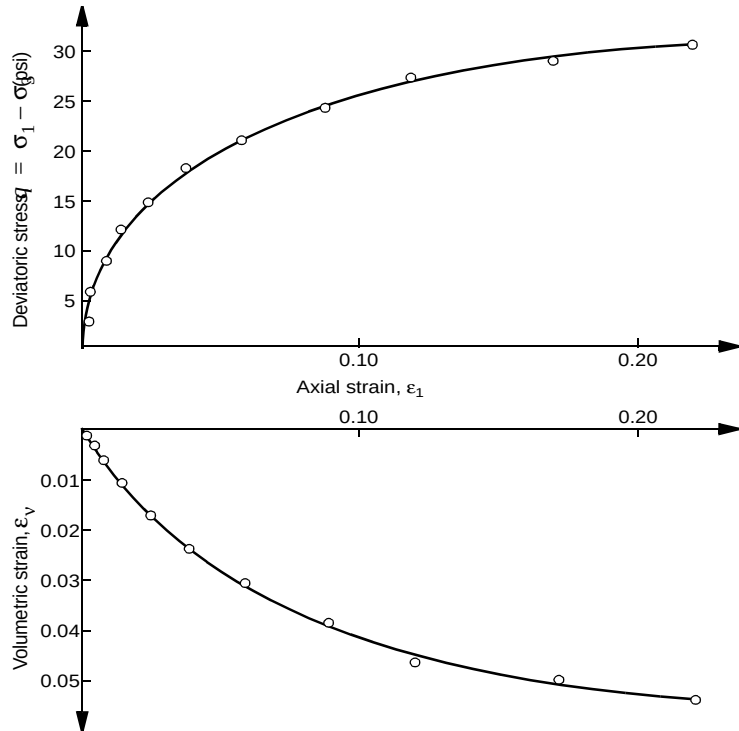


Figure 7-67 Constant Pressure Test: $p_0 = 30$ psi

Figures 7-68 to 7-70 show stress-strain relation plots for three fully drained tests performed at initial pressures 10, 20, and 30 psi. Here, there is no pore pressure development, and the effective mean pressure increases during the test. The void ratio values at the beginning and end of each test are given in Table 7-3.

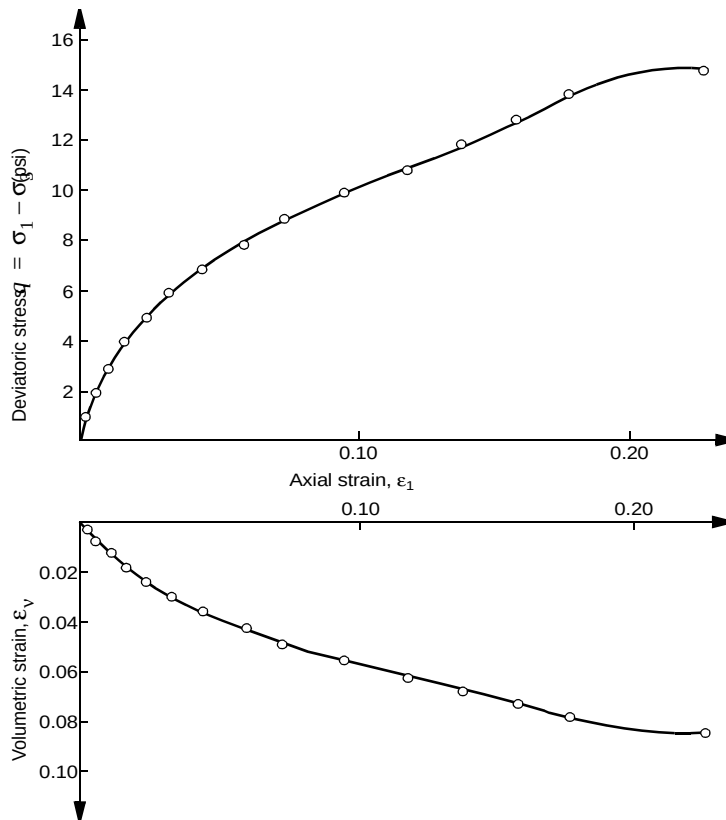


Figure 7-68 Drained Test: $p_0 = 10$ psi

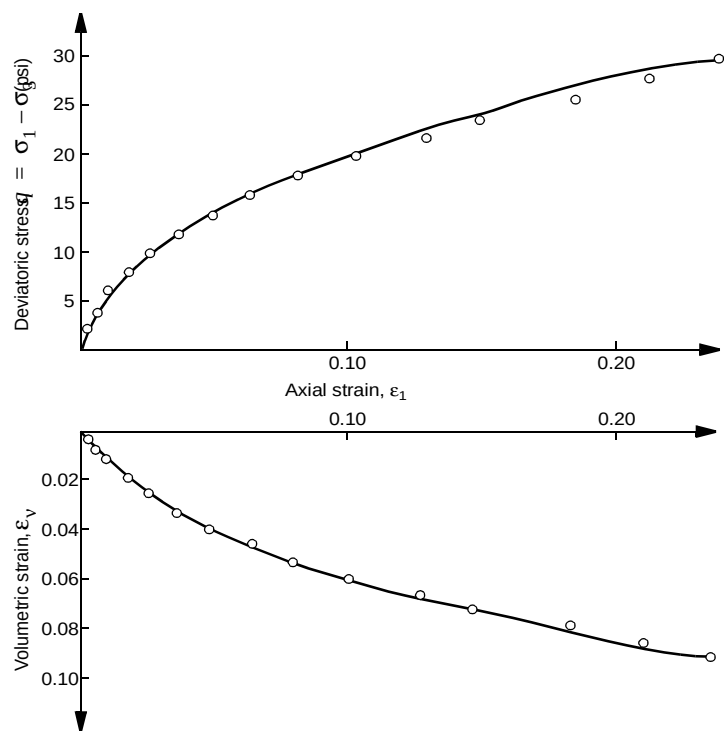


Figure 7-69 Drained Test: $p_0 = 20$ psi

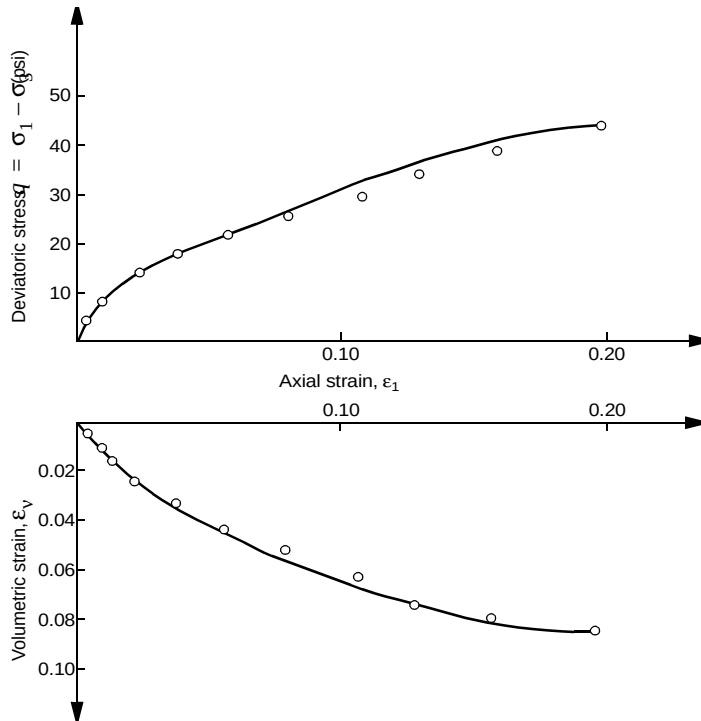


Figure 7-70 Drained Test: $p_0 = 30$ psi

Figures 7-71 to 7-73 show stress-strain behavior under undrained conditions. Here there is no change in volume of the sample. However, fluid pore pressure are developed during the test, thereby reducing the effective stresses until the ultimate (failure) state is reached.

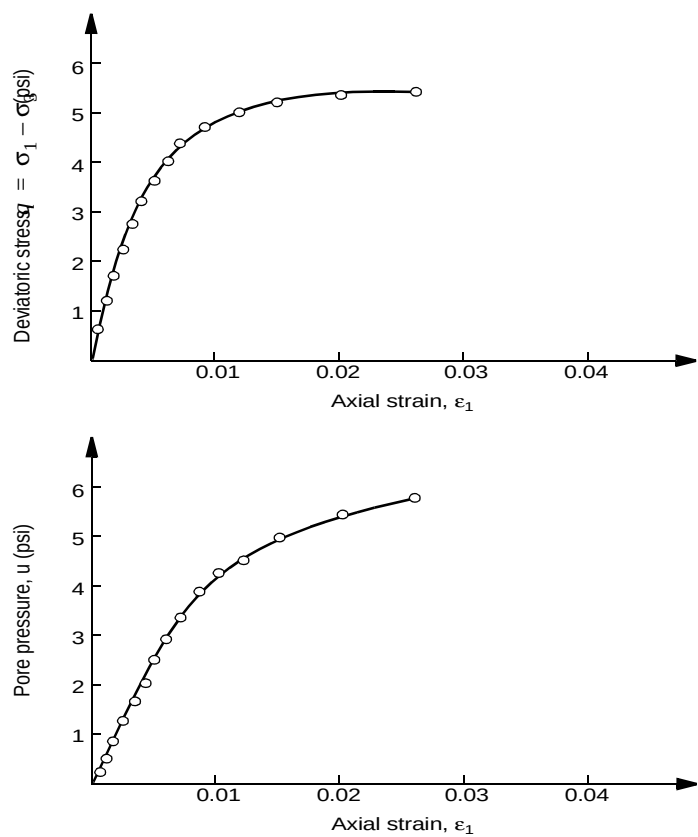


Figure 7-71 Undrained Test: $p_0 = 10$ psi

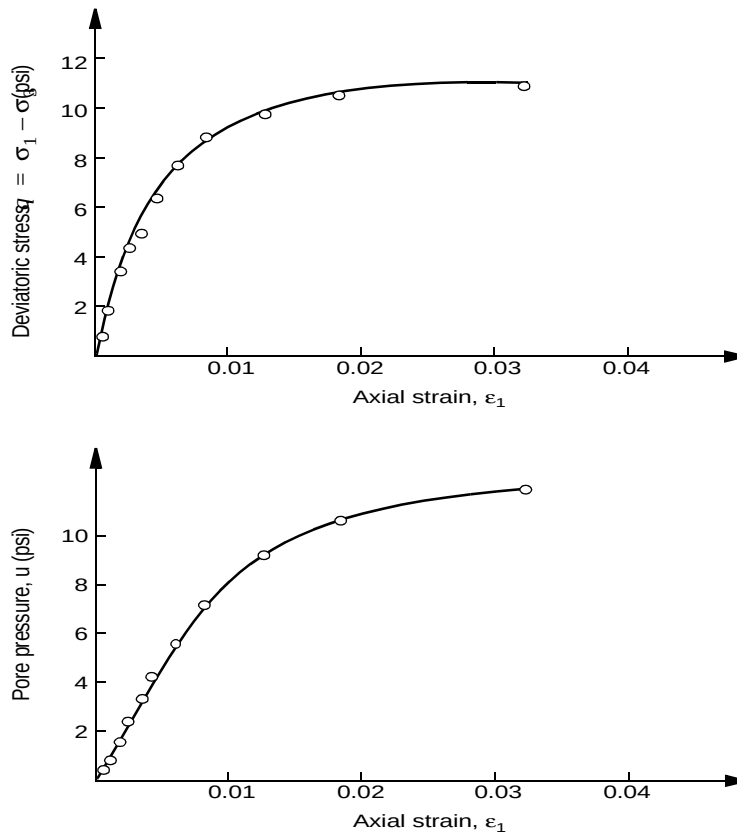


Figure 7-72 Undrained Test: $p_0 = 20$ psi

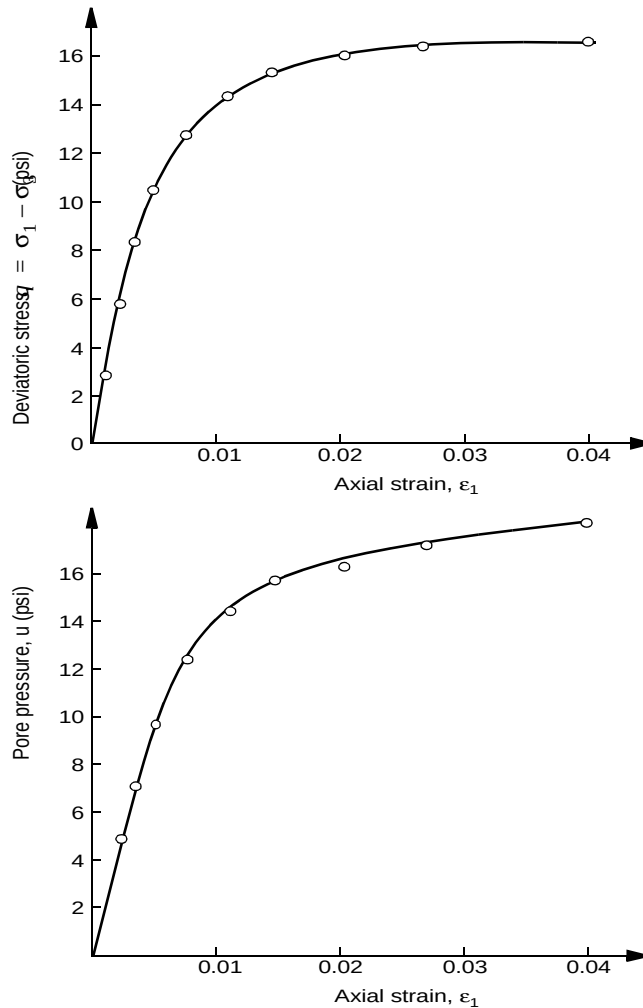


Figure 7-73 Undrained Test: $p_0 = 30$ psi

Figure 7-74 shows the variation of void ratio with mean pressure which is obtained from a hydrostatic compression (HC) test with three load reversals (that is, unloading-reloading cycles). In this figure, void ratio is plotted to a linear scale while the hydrostatic stress is plotted to a logarithmic (base 10) scale. In fact, this is the usual way of presenting one-dimensional consolidation or hydrostatic test results in geotechnical engineering practices.

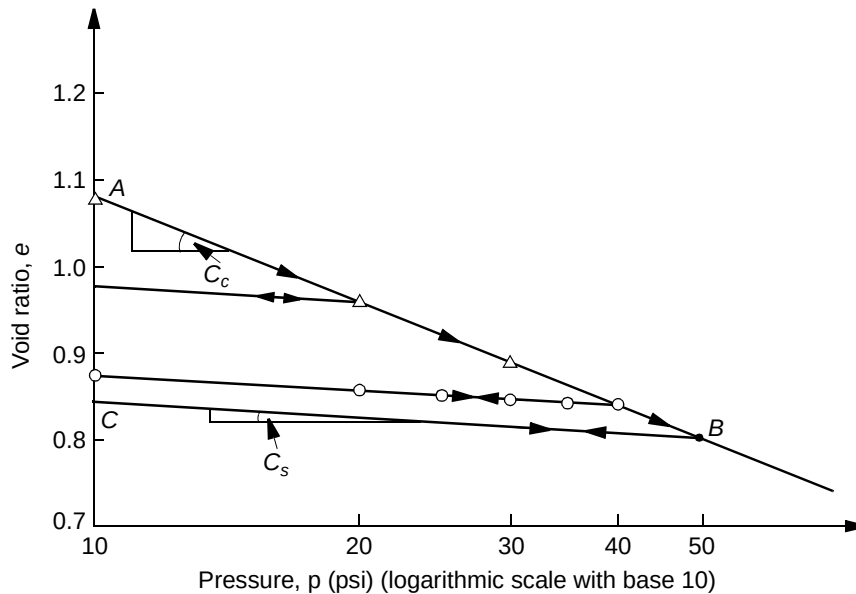


Figure 7-74 Hydrostatic Compression Test: $C_c = 0.04$, $C_s = 0.06$

Determination of parameter, M : The parameter M is the slope of the critical state line on a p - q plot. To determine its value, the values of p and q at ultimate conditions for each test are plotted as shown in Figure 7-75; the ultimate condition for each test is assumed to be the last point plotted in Figures 7-65 to 7-73. At the ultimate conditions, the sample undergoes excessive deformations under constant deviatoric stress, and hence it could be taken as the asymptotic stress to the curve. The slope of the critical state line (Figure 7-75) is calculated as 1.0 for the soil above. That is, $M = 1.0$.

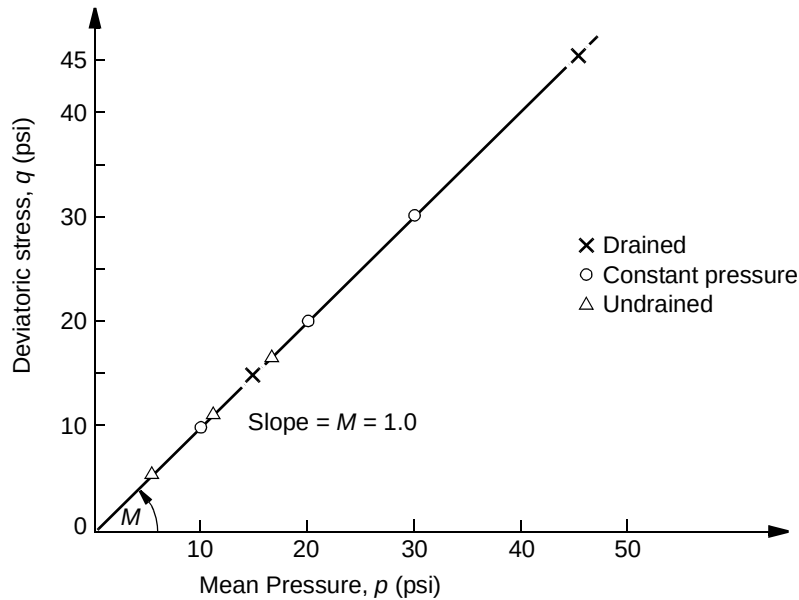


Figure 7-75 Critical State Line in q - p (psi)

Determination of parameters λ and κ : The values of gamma and kappa can be related to the commonly known quantities such as compression index (C_c) and swelling index (C_s). The compression index, C_c , is defined as the slope of virgin loading line on e - $\log_{10}p$ plot while the swelling index, C_s , is defined as the unloading-reloading curves on the same plot. Usually, the compression index and swelling index are defined with respect to a one-dimensional consolidation test. However, it can be shown that the e - $\ln(p)$ curve for any constant stress ratio test, that is, for constant q/p ratio, is parallel to that obtained from a hydrostatic test, (Figure 7-76).

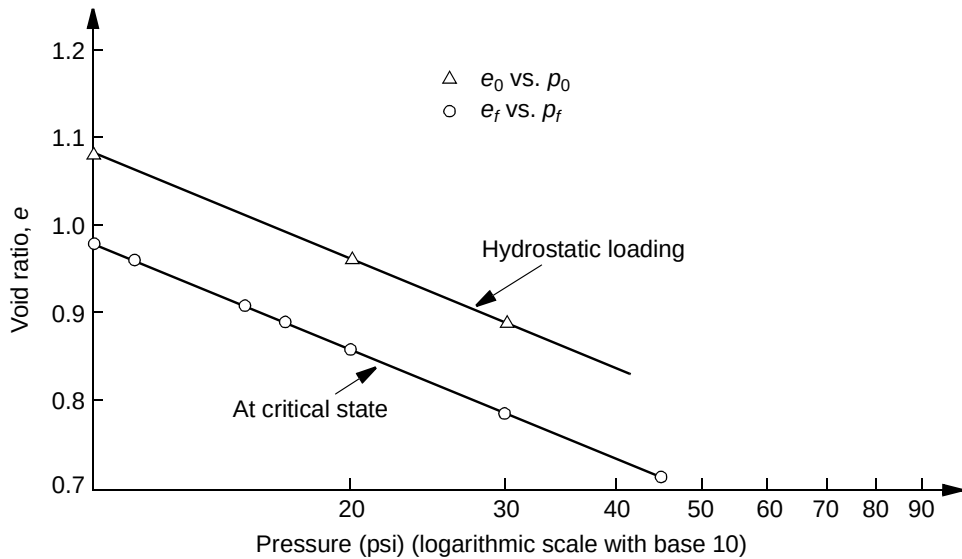


Figure 7-76 Isotropic Consolidation (Data from 5-8)

In fact, one-dimensional hydrostatic test is parallel to that obtained under critical state conditions. The values of gamma and kappa can be related to C_c and C_s as follows. The virgin compression line can be expressed as

$$e - e_0 = C_c \log_{10} \left(\frac{p}{p_0} \right) \quad (7-261)$$

or

$$e - e_0 = \lambda \ln \left(\frac{p}{p_0} \right) \quad (7-262)$$

and the swelling line (unloading-reloading) can be expressed as

$$e - e_0 = C_s \log_{10} \left(\frac{p}{p_0} \right) \quad (7-263)$$

or

$$e - e_0 = \kappa \ln \left(\frac{p}{p_0} \right) \quad (7-264)$$

Therefore, comparing Equation 7-261 and Equation 7-262, we have

$$\lambda = \frac{C_c}{\ln 10} = \frac{C_c}{2.303} \quad (7-265)$$

and comparing Equation 7-263 and Equation 7-264 yields

$$\kappa = \frac{C_s}{\ln 10} = \frac{C_s}{2.303} \quad (7-266)$$

The value of C_c can be computed by considering two points, A and B, in Figure 7-74 as

$$\begin{aligned} C_c &= \frac{e_A - e_B}{\log p_B - \log p_A} \\ &= \frac{1.08 - 0.08}{\log 50 - \log 10} \\ &= 0.40 \end{aligned} \quad (7-267)$$

Here the subscript denotes the value at that point. The swelling index, C_s , can be computed by considering points B and C of the same figures

$$\begin{aligned} C_s &= \frac{e_C - e_B}{\log p_B - \log p_C} \\ &= \frac{0.842 - 0.80}{\log 50 - \log 10} \\ &= 0.06 \end{aligned} \quad (7-268)$$

Hence, the values of gamma and kappa can be computed from Equation 7-265 and Equation 7-266 as

$$\lambda = \frac{0.40}{2.303} = 0.174 \quad (7-269)$$

Damage Models

In many structural applications, the finite element method is used to predict failure. This is often performed by comparing the calculated solution to some failure criteria, or by using classical fracture mechanics. Previously, we discussed two models where the actual material model changed due to some failure, see [Progressive Composite Failure on page 22](#) and the previous section on [Low Tension Material on page 133](#). In this section, the damage models appropriate for ductile metals and elastomeric materials will be discussed.

Ductile Metals

In ductile materials given the appropriate loading conditions, voids will form in the material, grow, then coalesce, leading to crack formation and potentially, failure. Experimental studies have shown that these processes are strongly influenced by hydrostatic stress. Gurson studied microscopic voids in materials and derived a set of modified constitutive equations for elastic-plastic materials. Tvergaard and Needleman modified the model with respect to the behavior for small void volume fractions and for void coalescence.

In the modified Gurson model, the amount of damage is indicated with a scalar parameter called the void volume fraction f . The yield criterion for the macroscopic assembly of voids and matrix material is given by:

$$F = \left(\frac{\bar{\sigma}}{\sigma_y} \right)^2 + 2q_1 f^* \cosh\left(\frac{q_2 \sigma_{kk}}{2\sigma_y} \right) - [1 + (q_1 f^*)^2] = 0 \quad (7-270)$$

as seen in [Figure 7-77](#).

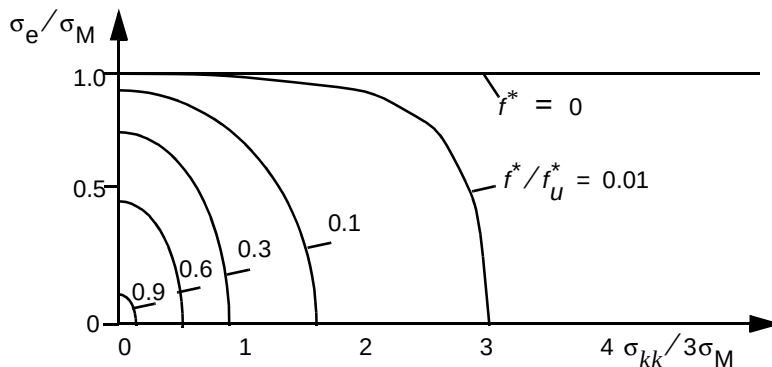


Figure 7-77 Plot of Yield Surfaces in Gurson Model

The parameter q_1 was introduced by Tvergaard to improve the Gurson model at small values of the void volume fraction. For solids with periodically spaced voids, numerical studies [Ref. 10] showed that the values of $q_1 = 1.5$ and $q_2 = 1$ were quite accurate.

The evolution of damage as measured by the void volume fraction is due to void nucleation and growth. Void nucleation occurs by debonding of second phase particles. The strain for nucleation depends on the particle sizes. Assuming a normal distribution of particle sizes, the nucleation of voids is itself modeled as a normal distribution in the strains, if nucleation is strain controlled. If void nucleation is assumed to be stress controlled in the matrix, a normal distribution is assumed in the stresses. The original Gurson model predicts that ultimate failure occurs when the void volume fraction f , reaches unity. This is too high a value and, hence, the void volume fraction f is replaced by the modified void volume fraction f^* in the yield function.

The parameter f^* is introduced to model the rapid decrease in load carrying capacity if void coalescence occurs.

$$\begin{aligned} f^* &= f & \text{if } f \leq f_c \\ f^* &= f_c + \left(\frac{f_u^* - f_c}{f_F - f_c} \right) (f - f_c) & \text{if } f > f_c \end{aligned} \quad (7-271)$$

where f_c is the critical void volume fraction, and f_F is the void volume at failure, and $f_u^* = 1/q_1$. A safe choice for f_F would be a value greater than $(1/q_1)$ namely, $f_F = 1.1/q_1$. Hence, you can control the void volume fraction, f_F , at which the solid loses all stress carrying capability.

Numerical studies show that plasticity starts to localize between voids at void volume fractions as low as 0.1 to 0.2. You can control the void volume fraction f_c , beyond which void-void interaction is modeled by MSC.Marc. Based on the classical studies, a value of $f_c = 0.2$ can be chosen.

The existing value of the void volume fraction changes due to the growth of existing voids and due to the nucleation of new voids.

$$\dot{f} = \dot{f}_{growth} + \dot{f}_{nucleation} \quad (7-272)$$

The growth of voids can be determined based upon compressibility of the matrix material surrounding the void.

$$\dot{f}_{growth} = (1 - f) \dot{\epsilon}_{kk}^p \quad (7-273)$$

As mentioned earlier, the nucleation of new voids can be defined as either strain or stress controlled. Both follow a normal distribution about a mean value.

In the case of strain controlled nucleation, this is given by

$$\dot{f}_{nucleation} = \frac{f_N}{S\sqrt{2\pi}} \exp\left[-\frac{1}{2}\left(\frac{\epsilon_m^p - \epsilon_n}{S}\right)^2\right] \dot{\epsilon}_m^p \quad (7-274)$$

where f_N is the volume fraction of void forming particles, ϵ_n the mean strain for void nucleation and S the standard deviation.

In the case of stress controlled nucleation, the rate of nucleation is given by:

$$\dot{f}_{nucleation} = \frac{f_N}{S\sqrt{2\pi}} \exp\left[-\frac{1}{2}\left(\frac{\bar{\sigma} + \frac{1}{3}\sigma_{kk} - \sigma_n}{S}\right)^2\right] * \left[\dot{\bar{\sigma}} + \frac{1}{3}\dot{\sigma}_{kk}\right] \quad (7-275)$$

If the second phase particle sizes in the solid are widely varied in size, the standard deviation would be larger than in the case when the particle sizes are more uniform. The MSC.Marc user can also input the volume fraction of the nucleating second phase void nucleating particles in the input deck, as the variable f_N .

A typical set of values for an engineering alloy is given by Tvergaard for strain controlled nucleation as

$$\epsilon_n = 0.30 ; f_N = 0.04 ; S = 0.01 . \quad (7-276)$$

It must be remarked that the determination of the three above constants from experiments is extremely difficult. The modeling of the debonding process must itself be studied including the effect of differing particle sizes in a matrix. It is safe to say that such an experimental study is not possible. The above three constants must necessarily be obtained by intuition keeping in mind the meaning of the terms.

When the material reaches 90 percent of f_F , the material is considered to be failed. At this point, the stiffness and the stress at this element are reduced to zero.

Elastomers

Under repeated application of loads, elastomers undergo damage by mechanisms involving chain breakage, multi-chain damage, micro-void formation, and micro-structural degradation due to detachment of filler particles from the network entanglement. Two types of phenomenological models namely, discontinuous and continuous, exists to simulate the phenomenon of damage.

1. Discontinuous Damage:

The discontinuous damage model simulates the “Mullins’ effect” as shown in [Figure 7-78](#).

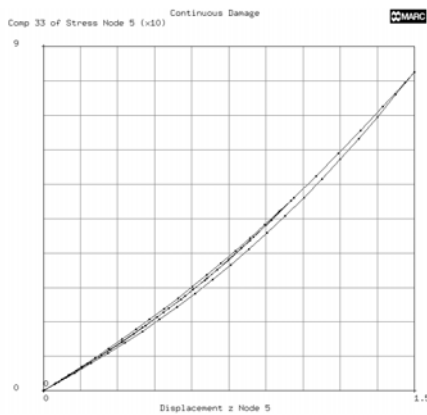


Figure 7-78 Discontinuous Damage

This involves a loss of stiffness below the previously attained maximum strain. The higher the maximum attained strain, the larger is the loss of stiffness. Upon reloading, the uniaxial stress-strain curve remains insensitive to prior behavior at strains above the previously attained maximum in a cyclic test. Hence, there is a progressive stiffness loss with increasing maximum strain amplitude. Also, most of the stiffness loss takes place in the few earliest cycles provided the maximum strain level is not increased. This phenomenon is found in both filled as well as natural rubber although the higher levels of carbon black particles increase the hysteresis and the loss of stiffness. The free energy, W , can be written as:

$$W = K(\alpha, \beta)W^0 \quad (7-277)$$

where W^0 is the nominal strain energy function, and

$$\alpha = \max(W^0) \quad (7-278)$$

determines the evolution of the discontinuous damage. The reduced form of Clausius-Duhem dissipation inequality yields the stress as:

$$S = 2K(\alpha, \beta) \frac{\partial W^0}{\partial C} \quad (7-279)$$

Mathematically, the discontinuous damage model has a structure very similar to that of strain space plasticity. Hence, if a damage surface is defined as:

$$\Phi = W - \alpha \leq 0 \quad (7-280)$$

The loading condition for damage can be expressed in terms of the Kuhn-Tucker conditions:

$$\Phi \leq 0 \quad \dot{\alpha} \geq 0 \quad \dot{\alpha} \Phi = 0 \quad (7-281)$$

The consistent tangent can be derived as:

$$C = 4 \left[K \frac{\partial^2 W^0}{\partial C \partial C} + \frac{\partial K}{\partial W^0} \frac{\partial W^0}{\partial C} \otimes \frac{\partial W^0}{\partial C} \right] \quad (7-282)$$

2. Continuous Damage:

The continuous damage model can simulate the damage accumulation for strain cycles for which the values of effective energy is below the maximum attained value of the past history as shown in [Figure 7-79](#).

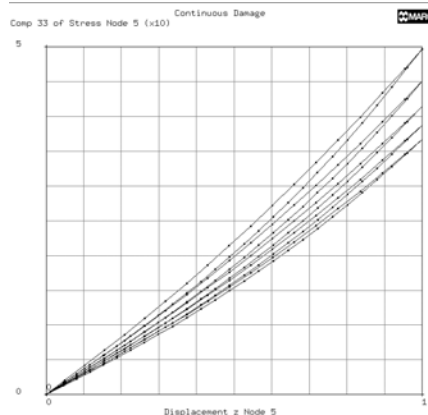


Figure 7-79 Continuous Damage

This model can be used to simulate fatigue behavior. More realistic modeling of fatigue would require a departure from the phenomenological approach to damage. The evolution of continuous damage parameter is governed by the arc length of the effective strain energy as:

$$\beta = \int_0^t \left| \frac{\partial}{\partial s'} W^0(s') \right| ds' \quad (7-283)$$

Hence, β accumulates continuously within the deformation process.

The Kachanov factor $K(\alpha, \beta)$ is implemented in MSC.Marc through both an additive as well as a multiplicative decomposition of these two effects as:

$$K(\alpha, \beta) = d^\infty + \sum_{n=1}^2 d_n^\alpha \exp\left(-\frac{\alpha}{\eta_n}\right) + \sum_{n=1}^2 d_n^\beta \exp\left(-\frac{\beta}{\lambda_n}\right) \quad (7-284)$$

$$K(\alpha, \beta) = d^\infty + \sum_{n=1}^2 d_n \exp\left(-\frac{\alpha + \delta_n \beta}{\eta_n}\right) \quad (7-285)$$

You specify the phenomenological parameters d_n^α , d_n^β , η_n , λ_n , d_n , δ_n and d^∞ . If d^∞ is not defined, it is automatically determined such that, at zero values of α and β , the Kachanov factor $K = 1$. If, according to [Equation 7-283](#) or [Equation 7-284](#) the value of K exceeds 1, K is set back to 1.

The above damage model is available for deviatoric behavior and is flagged by means of the [OGDEN](#) and [DAMAGE](#) model definition options. If, in addition, viscoelastic behavior is desired, the [VISCELOGDEN](#) option can be included. Finally, the user subroutine, [UELDAM](#), can be used to define damage functions different from [Equation 7-281](#) to [Equation 7-284](#).

The parameters required for the continuous or discontinuous damage model can be obtained using the experimental data fitting option in MSC.Marc Mentat.

Nonstructural Materials

In addition to stress analysis, MSC.Marc can be used for heat transfer, coupled thermo-electrical heating (Joule heating), coupled electrical-thermal-mechanical analysis (Joule-mechanical), hydrodynamic bearing, fluid/solid interaction, electrostatic, magnetostatic, electromagnetic, piezoelectric, acoustic, and fluid problems. Material properties associated with these analyses and MSC.Marc options that control these analyses are described below.

Heat Transfer Analysis

In heat transfer analysis, use model definition options, [ISOTROPIC](#) and [ORTHOTROPIC](#), to input values of thermal conductivity, specific heat, and mass density. If the latent heat effect is to be included in the analysis, the value of latent heat and associated solidus and liquidus temperatures must be entered through the [TEMPERATURE EFFECTS](#) model definition option. Both the thermal conductivity and specific heat can be dependent on temperatures. The mass density must be constant throughout the analysis. In addition, the [ANKOND](#) user subroutine can be used for the input of anisotropic thermal conductivity. For radiation analysis, you must enter the Stefan-Boltzmann constant in the [RADIATION](#) parameter and the emissivity through the [ISOTROPIC](#) option. The emissivity can be temperature dependent.

Thermo-Electrical Analysis

In addition to thermal conductivity, specific heat and mass density, the value of electric resistivity must be entered using the model definition options, [ISOTROPIC](#), and [ORTHOTROPIC](#), in a coupled thermo-electrical analysis. Input the variation of electric resistivity with temperature through the [TEMPERATURE EFFECTS](#) option.

Electrical-Thermal-Mechanical Analysis

Material properties for coupled electrical-thermal-mechanical analysis are the same for stress analysis in addition to those of joule heating analysis (thermal conductivity, specific heat and mass density and electric resistivity).

Hydrodynamic Bearing Analysis

In a hydrodynamic bearing analysis, the model definition options, [ISOTROPIC](#) and [ORTHOTROPIC](#), can be used for entering both the viscosity and specific mass. Define the temperature dependency of viscosity through the [TEMPERATURE EFFECTS](#) option.

Fluid/Solid Interaction Analysis – Added Mass Approach

In a fluid/solid analysis, the density of a fluid is given in the first field of the **ISOTROPIC** option. The fluid is assumed to be nonviscous and incompressible.

Electrostatic Analysis

In an electrostatic analysis, the permittivity can be defined through either the **ISOTROPIC** or **ORTHOTROPIC** options. In addition, the **UEPS** user subroutine can be used for the input of anisotropic permittivity.

Magnetostatic Analysis

In a magnetostatic analysis, the permeability can be defined through either the **ISOTROPIC** or **ORTHOTROPIC** options. A nonlinear relationship can be entered via the **B-H RELATION** option. The **UMU** user subroutine can be used for the input of isotropic permeability.

Electromagnetic Analysis

In an electromagnetic analysis, the permittivity, permeability and conductivity can be defined using either the **ISOTROPIC** or **ORTHOTROPIC** options. An assumption made is that the permittivity is a constant (does not vary with time) in the analysis. A nonlinear permeability can be entered via the **B-H RELATION** option. The user subroutines, **UEPS**, **UMU**, and **USIGMA**, can be used for the input of anisotropic permittivity, permeability, and conductivity, respectively. Both the dependent and harmonic analysis can be performed.

Piezoelectric Analysis

In a piezoelectric analysis, the material properties contain a mechanical part, an electrostatic part, and a part connecting these two. The material properties for the mechanical part are the same as for an elastic stress analysis. The material properties for the electrostatic part, the permittivity, and the material properties for the piezoelectric coupling can be defined through the **PIEZOELECTRIC** option. The coefficients for the piezoelectric coupling can be either stress based or strain based.

Acoustic Analysis

In an acoustic analysis of a cavity with rigid boundaries, the bulk modulus and the relative density of the medium can be entered through the **ISOTROPIC** option.

The **ACOUSTIC** parameter is used to indicate that a coupled acoustic-structural analysis is performed. In addition to the **CONTACT** option, the model definition options **ACOUSTIC** and **REGION** are used to define the material properties of the acoustic medium and to set which elements correspond to the solid and the fluid region.

Fluid Analysis

In the fluid analysis, viscosity and density can be defined using the **ISOTROPIC** option. In addition, conductivity and specific heat are defined for coupled fluid-thermal analysis. Non-Newtonian fluid behavior can be defined using the **STRAIN RATE** option. The **TEMPERATURE EFFECTS** option can be used to define the temperature dependence of the properties above.

References

1. Auricchio, F. and Taylor, R.L., "Shape-memory alloy: modeling and numerical simulations of the finite-strain superelastic behavior", *Comput. Methods Appl. Mech. Engrg.*, Vol. 143, pp.175-194 (1997).
2. Auricchio, F., "A robust integration-algorithm for a finite-strain shape-memory-alloy superelastic model", *Int. J. Plasticity*, Vol.17, pp.971-990 (2001).
3. Arruda, E. M. and Boyce, M. C. "A three-dimensional constitutive model for the large stretch behaviour of rubber elastic materials", *J. Mech. Phys. Solids*, Vol.41, No. 2, 1993.
4. Buyukozturk, O., "Nonlinear Analysis of Reinforced Concrete Structures", *Computers and Structures*, 7, 149-156 (1977).
5. Gent, A. N., "A new constitutive relation for rubber", *Rubber Chem. Tech.*, 69, 1996.
6. Barlat, F., Lege, D.J. and Brem, J.C., "A six-component yield function for anisotropic metals", *Int. J. Plasticity*, 7, 693-712 (1991).
7. Chung, K. and Shah, K., "Finite element simulation of sheet metal forming for planar anisotropic metals", *Int. J. Plasticity*, 8, 453-476 (1992).
8. Yoon, J.W., Yang, D.Y. and Chung, K. and Barlat, F., "A general elasto-plastic finite element formulation based on incremental deformation theory for planar anisotropy and its application to sheet metal forming", *Int. J. Plasticity*, 15, 35-67 (1999).
9. Yoon, J.W., Barlat, F., Chung, K., Pourboghrat, F. and Yang, D.Y., "Earing predictions based on asymmetric nonquadratic yield function", *Int. J. Plasticity*, 16, 1075-1104 (2000).
10. "Current Recommended Constitutive Equations for Inelastic Design Analysis of FFTF Components." ORLN-TM-360Z, October 1971.
11. "Finite Element Calculation of Stresses in Glass Parts Undergoing Viscous Relaxation", *J.Am.Ceram.Soc.* Vol. 70[2], pp. 90-95, 1987.
12. Mooney, M. J. *Appl Phys.*, Vol. II, p. 582, 1940.
13. Naghdi, P. M. "Stress-Strain Relations in Plasticity and Thermoplasticity." *In Plasticity, Proceedings of the Second Symposium on Naval Structural Mechanics*, edited by E. H. Lee and P. S. Symonds. Pergamon Press, 1960.

14. Narayanaswamy, O.S., "A Model of Structural Relaxation in Glass", *J.Am.Ceram.Soc.*, Vol. 54[10], pp. 491-498, 1971.
15. Prager, W. *Introduction to Mechanics to Continua*. New York: Dover Press, 1961.
16. "Proposed Modification to RDT Standard F9-5T Inelastic Analysis Guidelines." ORNL, October 1978.
17. Rivlin, R. S. *Phil Trans Roy Soc (A)*, Vol. 240, 459, 1948.
18. Rivlin, R. S. *Phil Trans Roy Soc (A)*, Vol. 241, 3-79, 1948.
19. Timoshenko, S. P., and J. N. Goodier. *Theory of Elasticity*. Third Ed. New York: McGraw-Hill, 1970.
20. Simo, J. C. "On a Fully Three-dimensional Finite Strain Viscoelastic Damage Model: Formulation and Computational Aspects," *Computer Methods in Applied Mechanics and Engineering*, Vol. 60, 1987, pp. 153-173.
21. Tvergaard, V., "Influence of Voids on Shear Band Instabilities under Plane Strain Conditions", *Int. J. Fracture*, Vol. 17, pp. 389-407, 1981.
22. Tvergaard, V., "Material Failure by Void Coalescence in Localized Shear Bands", in *Int. J. Solids Struct.*, Vol. 18, No. 8, pp. 652-672, 1982.
23. Chaboche, J. L., "Constitutive Equations for Cyclic Plasticity and Cyclic Viscoplasticity", *International Journal of Plasticity*, Vol. 5, pp. 247-302, 1989
24. van den Boogard, A.H., "Implicit integration of the Perzyna viscoplastic material model", TNO Building and Construction Research, The Netherlands, TNO-report, 95-NM-R711, 199

CHAPTER

8

Contact

- Numerical Procedures
- Definition of Contact Bodies
- Numbering of Contact Bodies
- Motion of Bodies
- Detection of Contact
- Implementation of Constraints
- Friction Modeling
- Separation
- Coupled Analysis
- Thermal Contact
- Element Considerations
- Dynamic Impact

- [Global Adaptive Meshing and Rezoning](#)
- [Adaptive Meshing](#)
- [Result Evaluation](#)
- [Tolerance Values](#)
- [Workspace Reservation](#)
- [Mathematical Aspects of Contact](#)
- [References](#)

The simulation of many physical problems requires the ability to model the contact phenomena. This includes analysis of interference fits, rubber seals, tires, crash, and manufacturing processes among others. The analysis of contact behavior is complex because of the requirement to accurately track the motion of multiple geometric bodies, and the motion due to the interaction of these bodies after contact occurs. This includes representing the friction between surfaces and heat transfer between the bodies if required. The numerical objective is to detect the motion of the bodies, apply a constraint to avoid penetration, and apply appropriate boundary conditions to simulate the frictional behavior and heat transfer. Several procedures have been developed to treat these problems including the use of Perturbed or Augmented Lagrangian methods, penalty methods, and direct constraints. Furthermore, contact simulation has often required the use of special contact or gap elements. MSC.Marc allows contact analysis to be performed automatically without the use of special contact elements. A robust numerical procedure to simulate these complex physical problems has been implemented in MSC.Marc.

Numerical Procedures

Lagrange Multipliers

In performing contact analysis, you are solving a constrained minimization problem where the constraint is the ‘no penetration’ constraint. The Lagrange multiplier technique is the most elegant procedure to apply mathematical constraints to a system. Using this procedure, if the constraints are properly written, overclosure or penetration does not occur. Unfortunately, Lagrange multipliers lead to numerical difficulties with the computational procedure as their inclusion results in a nonpositive definite mathematical system. This requires additional operations to insure an accurate, stable solution which leads to high computational costs. Another problem with this method is that there is no mass associated with the Lagrange multiplier degree of freedom. This results in a global mass matrix which cannot be inverted. This precludes the use of Lagrange multiplier techniques in explicit dynamic calculations which are often used in crash simulations. The Lagrange multiplier technique has often been implemented in contact procedures using special interface elements such as the MSC.Marc gap element (type 12). This facilitates the correct numerical procedure, but puts a restriction on the amount of relative motion that can occur between bodies. The use of interface elements requires an apriori knowledge of where contact occurs. This is unachievable in many physical problems such as crash analysis or manufacturing simulation.

Penalty Methods

The penalty method or its extension, the Augmented Lagrangian method, is an alternative procedure to numerically implement the contact constraints. Effectively, the penalty procedure constrains the motion by applying a penalty to the amount of penetration that occurs. The penalty approach can be considered as analogous to a nonlinear spring between the two bodies. Using the penalty approach, some penetration occurs with the amount being determined by the penalty constant or function. The choice of the penalty value can also have a detrimental effect on the numerical stability of the global solution procedure. The penalty method is relatively easy to implement and has been extensively used in explicit dynamic analysis although it can result in an overly stiff system for deformable-to-deformable contact since the contact pressure is assumed to be proportional to the pointwise penetration. The pressure distribution is generally oscillatory.

Hybrid and Mixed Methods

In the hybrid method, the contact element is derived from a complementary energy principle by introducing the continuity on the contact surface as a constraint and treating the contact forces as additional elements. Mixed methods, based on perturbed Lagrange formulation, usually consist of pressure distribution interpolation which is an order less than the displacement field, have also been used to alleviate the difficulties associated with the pure Lagrange method.

Direct Constraints

Another method for the solution of contact problems is the direct constraint method. In this procedure, the motion of the bodies is tracked, and when contact occurs, direct constraints are placed on the motion using boundary conditions – both kinematic constraints on transformed degrees of freedom and nodal forces. This procedure can be very accurate if the program can predict when contact occurs. This is the procedure that is implemented in MSC.Marc through the [CONTACT](#) option. No special interference elements are required in this procedure and complex changing contact conditions can be simulated since no apriori knowledge of where contact occurs is necessary. The details of the implementation are presented later.

Definition of Contact Bodies

There are two types of contact bodies in MSC.Marc – deformable and rigid. Deformable bodies are simply a collection of finite elements as shown below.

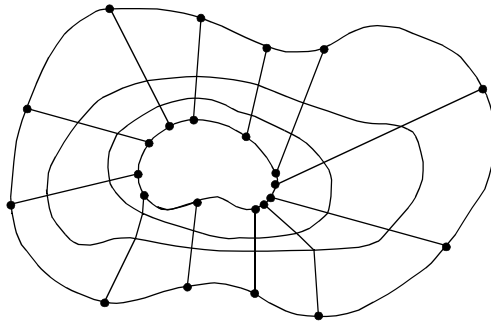


Figure 8-1 Deformable Body

This body has three key aspects to it:

1. The elements which make up the body.
2. The nodes on the external surfaces which might contact another body or itself. These nodes are treated as potential contact nodes.
3. The edges (2-D) or faces (3-D) which describe the outer surface which a node on another body (or the same body) might contact. These edges/faces are treated as potential contact segments.

Note that a body can be multiply connected (have holes in itself). It is also possible for a body to be composed of both triangular elements and quadrilateral elements in 2-D or tetrahedral elements and brick elements in 3-D. Beam elements and shells are also available for contact. One should not mix continuum elements, shells, and/or beams in the same contact body.

Each node and element should be in, at most, one body. The elements in a body are defined using the **CONTACT** option. It is not necessary to identify the nodes on the exterior surfaces as this is done automatically. The algorithm used is based on the fact that nodes on the boundary are on element edges or faces that belong to only one element. Each node on the exterior surface is treated as a potential contact node. In many problems, it is known that certain nodes never come into contact; in such cases, the **CONTACT NODE** option can be used to identify the relevant nodes. As all nodes on free surfaces are considered contact nodes, if there is an error in the mesh generation such that internal holes or slits exist, undesirable results can occur.

Note: These problems can be visualized by using MSC.Marc Mentat to create an outline plot and fixed by using the sweep command in MSC.Marc Mentat.

The potential segments composed of edges or faces are treated in potentially two ways. The default is that they are considered as piece-wise linear (PWL). As an alternative, a cubic spline (2-D) or a Coons surface (3-D) can be placed through them. The **SPLINE** option is used to activate this procedure. This improves the accuracy of the calculation of the normal.

Rigid bodies are composed of curves (2-D) or surfaces (3-D) or meshes with only thermal elements in coupled problems. The most significant aspect of rigid bodies is that they do not deform. Deformable bodies can contact rigid bodies, but contact between rigid bodies is not considered.

They can be created either in CAD systems and transferred through MSC.Marc Mentat into MSC.Marc, created within MSC.Marc Mentat, or created directly through the MSC.Marc input. There are several different types of curves and surfaces that can be entered including:

Table 8-1 Geometrical Entities Used in Modeling Contact

2-D	3-D
line	4-node patch
circular arc	ruled surface
spline	surface of revolution
NURBS	Bezier
	poly-surface
	cylinder
	sphere
	NURBS
	trimmed NURBS

Within the MSC.Marc Mentat GUI, all curves or surfaces are mathematically treated as NURBS curves or surfaces. This allows the greatest level of generality. Within the analysis, these rigid curves or surfaces can be treated in two ways – discrete piecewise linear lines (2-D) or patches (3-D), or as analytical NURBS curves or surfaces. When the discrete approach is used, all geometric primitives are subdivided into straight line segments or bilinear patches. You have control over the density of these subdivisions to approximate a curved line or surface within a desired degree of accuracy. This subdivision is also relevant when the corner conditions (see [Detection of Contact on page 8-13](#)) is determined. Too few subdivisions might lead to a very bad surface description and too many subdivisions might lead to a more expensive analysis due to more complex contact checking. When using the analytical description, the number of subdivisions given is less relevant. It should be large enough to roughly visualize

the shape of the NURBS. In 3-D, a too large number is automatically reduced so that the analysis cost is not influenced too much. The treatment of the rigid bodies as NURBS surfaces is advantageous because it leads to greater accuracy in the representation of the geometry and a more accurate calculation of the surface normal. Additionally, the variation of the surface normal is continuous over the body which leads to a better calculation of the friction behavior and a better convergence.

To create a rigid body, you can either read in the curve and surface geometry created from a CAD system or create the geometry in MSC.Marc Mentat or directly enter it in MSC.Marc. You then use the **CONTACT** option to select which geometric entities are to be a part of the rigid body. An important consideration for a rigid body is the definition of the interior side and the exterior side. For two-dimensional analysis, the interior side is formed by the right-hand rule when moving along the body.

This is easily visualized with MSC.Marc Mentat by activating ID CONTACT (2-D).

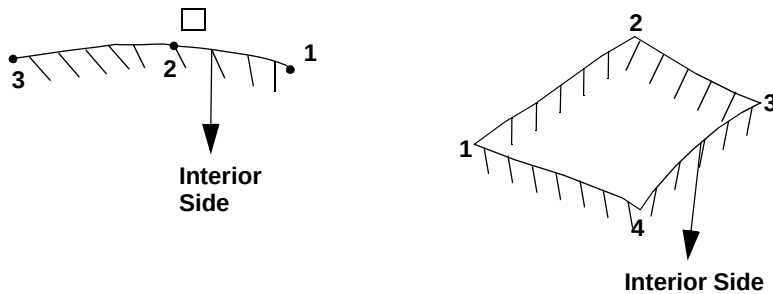


Figure 8-2 Orientation of Rigid Body Segments

For three-dimensional analysis, the interior side is formed by the right-hand rule along a patch. The interior side is visualized in MSC.Marc Mentat as the front surface (pink); whereas, the exterior side is visualized in MSC.Marc Mentat as the back surface (gold).

It is not necessary for rigid bodies to define the complete body. Only the bounding surface needs to be specified. You should take care, however, that the deforming body cannot slide out of the boundary curve in 2-D (**Figure 8-3**). This means that it must always be possible to decompose the displacement increment into a component normal and a component tangential to the rigid surface.

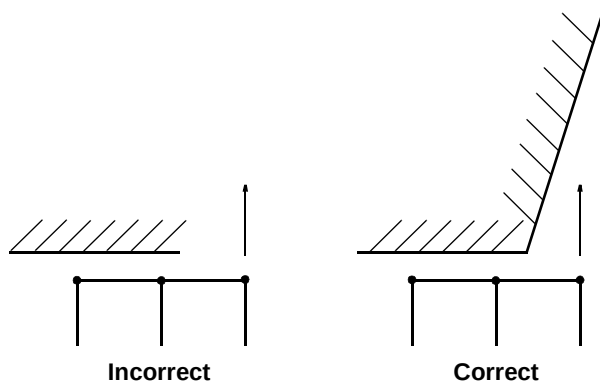


Figure 8-3 Deformable Surface Sliding Out of Rigid Surface

Numbering of Contact Bodies

When defining contact bodies for a deformable-to-deformable analysis, it is important to define them in the proper order. As a general rule, a body with a finer mesh should be defined before a body with a coarser mesh.

Note: For problems involving adaptive meshing or automated remeshing, care must be taken to satisfy this rule before as well as after the mesh change.

If one has defined a body numbering which violates the general rule, or if the rule is violated upon remeshing, then a CONTACT TABLE [model](#) or [history](#) definition option can be used to modify the order in which contact is established. This order can be directly user-defined or decided by the program. In the latter case, the order is based on the rule that if two deformable bodies might come into contact, searching is done for nodes of the body having the smallest element edge length. It should be noted that this implies single-sided contact for this body combination, as opposed to the default double-sided contact.

Motion of Bodies

The motion of deformable bodies is prescribed using the conventional methods of applying displacements, forces, or distributed loads to the bodies. It is advantageous to not apply displacements or point loads to nodes which might come into contact with other rigid bodies. If a prescribed displacement is to be imposed, it is better to introduce another rigid body and apply the motion to the rigid body. Symmetry surfaces are treated as a special type of bodies which have the property of being frictionless and where the nodes are not allowed to separate.

There are four ways to prescribe the motion of rigid surfaces:

- Prescribed velocity
- Prescribed position
- Prescribed load
- Prescribed scaling

Associated with the rigid body is a point labeled the centroid. When the first two methods are chosen, you define the translational motion of this point, and the angular motion about an axis through this point. The direction of the axis can be defined for three-dimensional problems. For two-dimensional problems, it is a line normal to the plane. For complex time-dependent behavior, the **MOTION** user subroutine can be used to prescribe the motion as an alternative to the input. The motion during a time increment is considered to be linear. The position is determined by an explicit, forward integration of the velocities based upon the current time step. A time increment must always be defined even if a static, rate-independent analysis is performed.

When load controlled rigid bodies are used, two additional nodes, called the control nodes, are associated with each rigid body. In 2-D problems, the first node has two translational degrees of freedom (corresponding to the global x- and y-direction) and the second node has one rotational degree of freedom (corresponding to the global z-direction). In 3-D problems, the first node has three translational degrees of freedom (corresponding to global x-, y-, and z-direction) and the second node has three rotational degrees of freedom (corresponding to the global x-, y-, and z-direction). In this way, both forces and moments can be applied to a body by using the **POINT LOAD** option for the control nodes. Alternatively, one may prescribe one or more degrees of freedom of the control nodes by using the **FIXED DISP** or **DISP CHANGE** options. Generally speaking, load-controlled bodies can be considered as rigid bodies with three (in 2-D) or six (in 3-D) degrees of freedom. The prescribed position and prescribed velocity methods (see [Figure 8-4](#)) have less computational costs than the prescribed load method (see [Figure 8-5](#)).

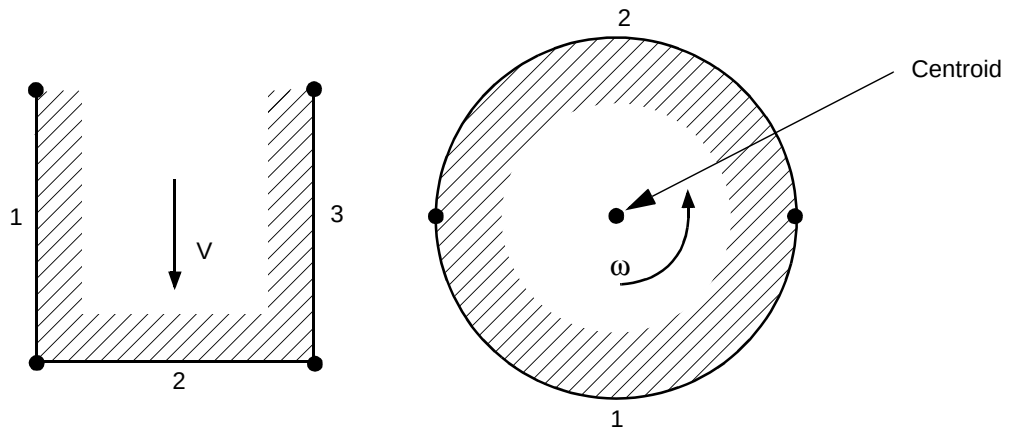


Figure 8-4 Velocity Controlled Rigid Surface

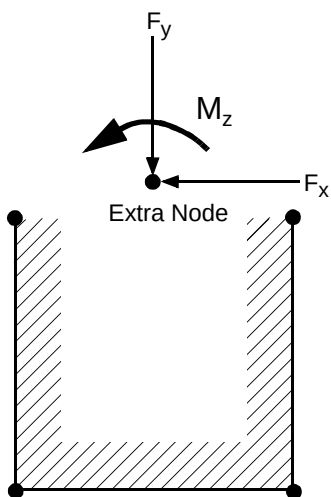


Figure 8-5 Load Controlled Rigid Surface

If the second control node is not specified, the rotation of the body is prescribed to be zero.

It should be noted that, irrespective of the coordinates of the control nodes, MSC.Marc positions the control nodes in the center of rotation of the body.

It is also possible to either expand or contract the size of the rigid body through the use of user subroutine [UGROWRIGID](#). If no rotations are applied, the scale factors can be different in the x-, y-, and z-directions.

Initial Conditions

At the beginning of the analysis, bodies should either be separated from one another or in contact. Bodies should not penetrate one another unless the objective is to perform an interference fit calculation. Rigid body profiles are often complex, making it difficult for you to determine exactly where the first contact is located. Before the analysis begins (increment zero), if a rigid body has a nonzero motion associated with it, the initialization procedure brings it into first contact with a deformable body. No motion or distortion occurs in the deformable bodies during this process. In a coupled thermal mechanical analysis, no heat transfer occurs during this process. If more than one rigid body exists in the analysis, each one with a nonzero initial velocity is moved until it comes into contact. Because increment zero is used to bring the rigid bodies into contact only, you should not prescribe any loads (distributed or point) or prescribed displacements initially. For multistage contact analysis (often needed to simulate manufacturing processes), the [APPROACH](#) and [SYNCHRONIZED](#) options in conjunction with the [CONTACT TABLE](#) and [MOTION CHANGE](#) options allow you to model contact bodies so that they just come into contact with the workpiece.

In assembly analysis, it is possible that multiple bodies are initially in contact. Because of mesh discretization, it is possible that the contact is not perfect, which would result in inducing stresses due to overclosure. The [CONTACT TABLE](#) option may be used to specify that the initial contact should be stress free. In such cases, the coordinates of the nodes are relocated on the surface so that initial stresses do not occur.

Detection of Contact

During the incremental procedure, each potential contact node is first checked to see whether it is near a contact segment. The contact segments are either edges of other 2-D deformable bodies, faces of 3-D deformable bodies, or segments from rigid bodies. By default, each node could contact any other segment including segments on the body that it belongs to. This allows a body to contact itself. To simplify the computation, it is possible to use the [CONTACT TABLE](#) option to indicate that a particular body will or will not contact another body. This is often used to indicate that a body will not contact itself. During the iteration process, the motion of the node is checked to see whether it has penetrated a surface by determining whether it has crossed a segment.

Because there can be a large number of nodes and segments, efficient algorithms have been developed to expedite this process. A bounding box algorithm is used so that it is quickly determined whether a node is near a segment. If the node falls within the bounding box, more sophisticated techniques are used to determine the exact status of the node.

During the contact process, it is unlikely that a node exactly contacts the surface. For this reason, a contact tolerance ([Figure 8-6](#)) is associated with each surface.

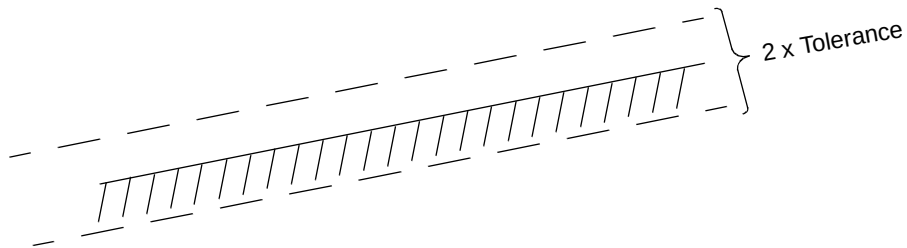


Figure 8-6 Contact Tolerance

If a node is within the contact tolerance, it is considered to be in contact with the segment. The contact tolerance is calculated by the program as the smaller of 5% of the smallest element side or 25% of the smallest (beam or shell) element thickness. It is also possible for you to define the contact tolerance through the input.

During an increment, if node A moves from $A^{(t)}$ to $A^{(\text{trial})}(t + \Delta t)$, where $A^{(\text{trial})}(t + \Delta t)$ is beyond the contact tolerance, the node is considered to have penetrated (Figure 8-7). In such a case, depending upon which numerical implementation is used, either the increment is divided into subincrements as discussed in “Mathematical Aspects of Contact” in this chapter, the increment is reduced in size, an iterative procedure is invoked, or the penetration is neglected in this increment.

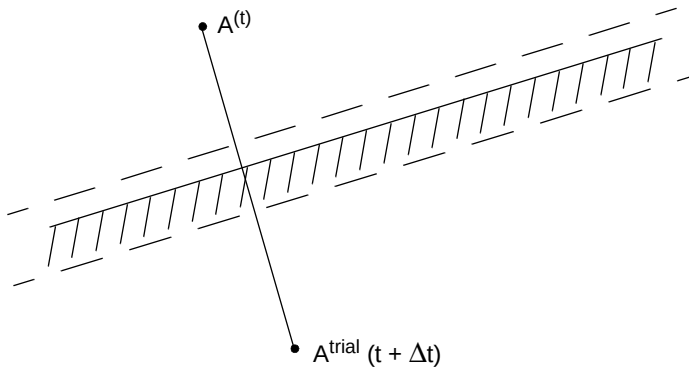


Figure 8-7 Trial Displacement with Penetration

The size of the contact tolerance has a significant impact on the computational costs and the accuracy of the solution. If the contact tolerance is too small, detection of contact and penetration is difficult which leads to higher costs. Penetration of a node happens in a shorter time period leading to more recycles due to iterative penetration checking or to more increment splitting and increases the computational costs. If the contact tolerance is too large, nodes are considered in contact prematurely, resulting in a loss of accuracy or more recycling due to separation. Furthermore, the accepted solution might have nodes that “penetrate” the surface less than the error tolerance, but more than desired by the user. The default error tolerance is recommended.

Many times, areas exist in the model where nodes are almost touching a surface (for example, in rolling analysis close to the entry and exit of the rolls). In such cases, the use of a biased tolerance area with a smaller distance on the outside and a larger distance on the inside is advised. This avoids the close nodes from coming into contact and separating again and is accomplished by entering a bias factor. The bias factor should be given a value between 0.0 and 0.99. The default is 0.0 or no bias. Also, in analyses involving frictional contact, a bias factor for the contact tolerance is

recommended. The outside contact area is $(1 - \text{bias})$ times the contact tolerance on the inside contact area $(1 + \text{bias})$ times the contact tolerance ([Figure 8-8](#)). The bias factor recommended value is 0.95.

In some instances, you might wish to influence the decision regarding the deformable segment a node contacts (or does not contact). This can be done using the [EXCLUDE](#) option.

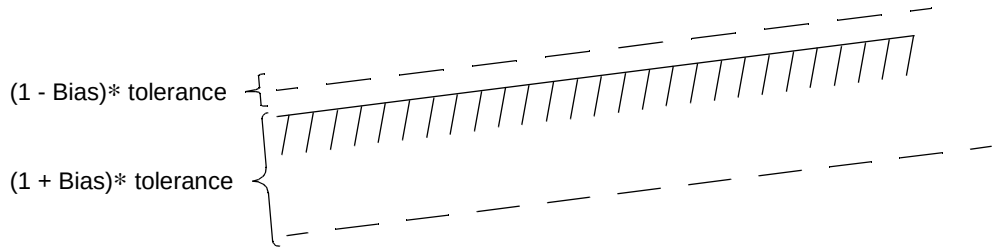


Figure 8-8 Biased Contact Tolerance

Shell Contact

A node on a shell makes contact when the position of the node plus or minus half the thickness projected with the normal comes into contact with another segment ([Figure 8-9](#)). In 2-D, this can be shown as:

$$x_1 = A + n \, t/2$$

$$x_2 = A - n \, t/2$$

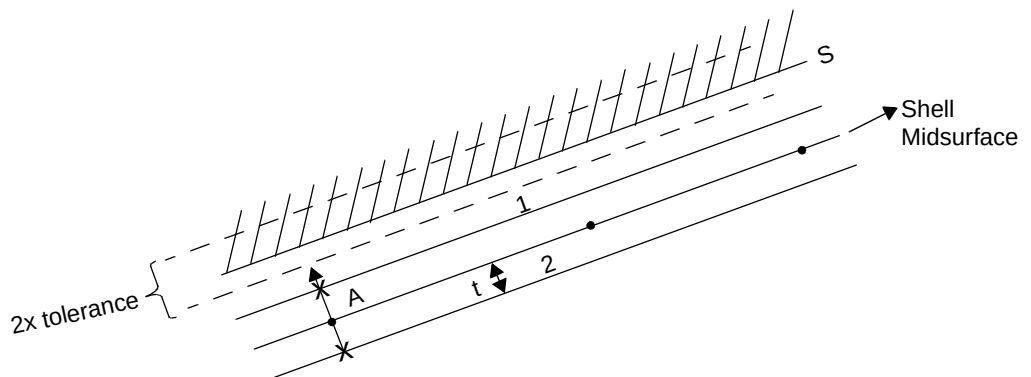


Figure 8-9 Default Shell Contact

If point x or y falls within the contact tolerance distance of segment S , node A is considered in contact with the segment S . Here x_1 and x_2 are the position vectors of a point on the surfaces 1 and 2 on the shell, A is the position vector of a point (node in a discretized model) on the midsurface of the shell, n is the normal to the midsurface, and t is the shell thickness.

As the shell has finite thickness, the node (depending on the direction of motion) can physically contact either the top surface, bottom surface, or mathematically contact can be based upon the midsurface. You can control whether detection occurs with either both surfaces, the top surface, the bottom surface, or the middle surface. In such cases, either two or one segment will be created at the appropriate physical location. Note that these segments will be dependent, not only on the motion of the shell, but also the current shell thickness ([Figure 8-10](#)).

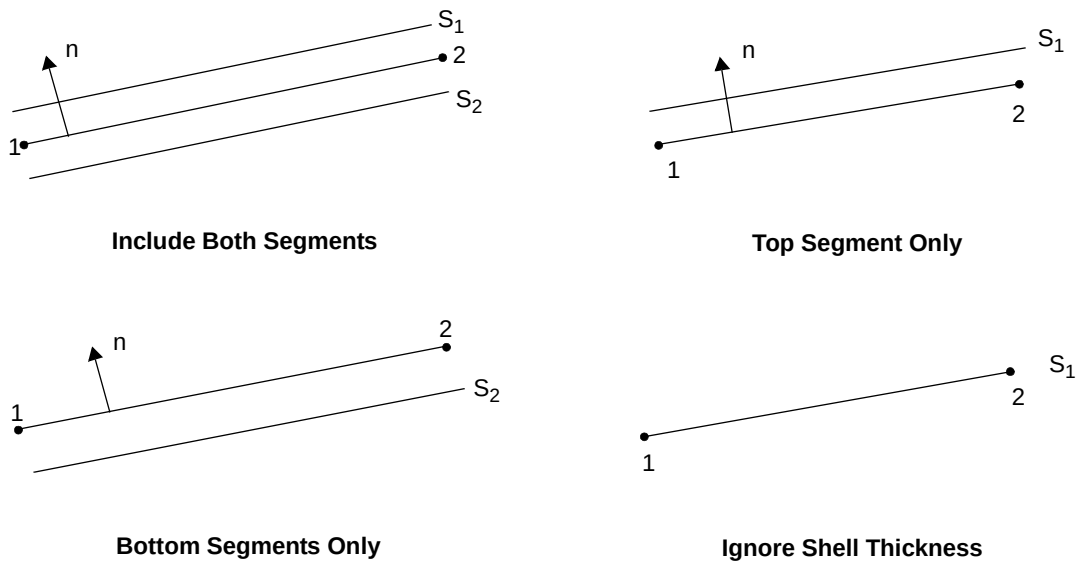


Figure 8-10 Selective Shell Contact

S_1 , S_2 are segments associated with shell consisting of node 1 and 2.

Neighbor Relations

When a node is in contact with a rigid surface, it tends to slide from one segment to another. In 2-D, the segments are always continuous and so are the segment numbers. Hence, a node in contact with segment n slides to segment $n - 1$ or to segment $n + 1$ (Figure 8-11). This simplifies the implementation of contact.

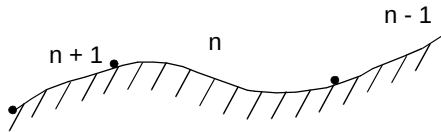


Figure 8-11 Neighbor Relationship (2-D)

In 3-D, the segments are often discontinuous (Figure 8-12). This can be due to the subdivision of matching surfaces or, more likely, the CAD definition of the underlying surface geometry.

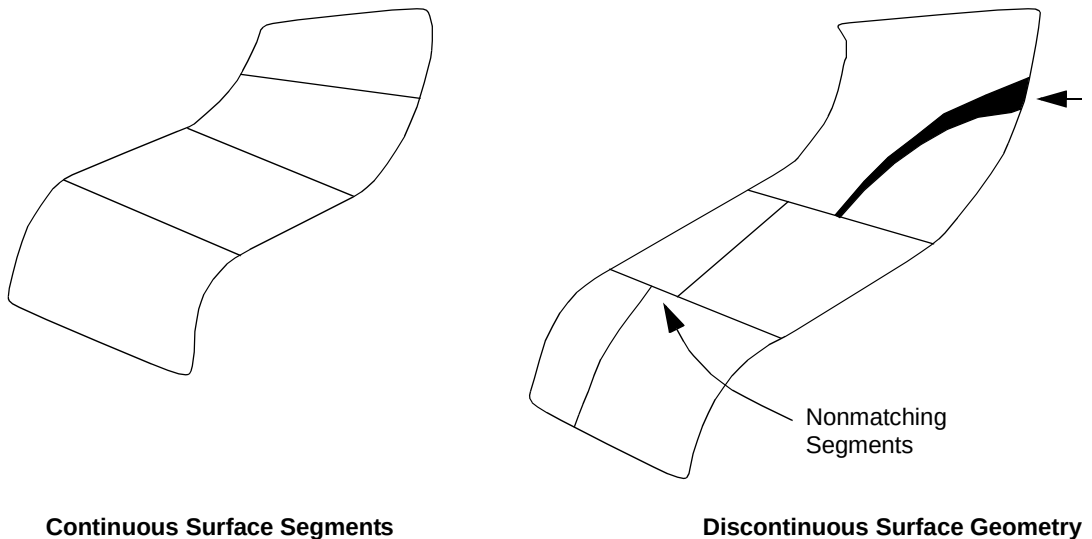


Figure 8-12 Neighbor Relationship (3-D)

Continuous surface geometry is highly advantageous as a node can slide from one segment to the next with no interference (assuming the corner conditions are satisfied). Discontinuous surface geometry results in additional operations when a node slides off a patch and cannot find an adjacent segment. Hence, it is advantageous to use geometry clean-up tools to eliminate small sliver surfaces and make the surfaces both physically continuous and topologically contiguous.

Implementation of Constraints

For contact between a deformable body and a rigid surface, the constraint associated with no penetration is implemented by transforming the degrees of freedom of the contact node and applying a boundary condition to the normal displacement. This can be considered solving the problem:

$$\begin{bmatrix} K_{\hat{a}\hat{a}} & K_{\hat{a}b} \\ K_{b\hat{a}} & K_{bb} \end{bmatrix} \begin{Bmatrix} u_{\hat{a}} \\ u_b \end{Bmatrix} = \begin{Bmatrix} f_{\hat{a}} \\ f_b \end{Bmatrix}$$

where $\hat{a}$ represents the nodes in contact which have a local transformation, and b represents the nodes not in contact and, hence, not transformed. Of the nodes transformed, the displacement in the normal direction is then constrained such that $\delta u_{\hat{a}n}$ is equal to the incremental normal displacement of the rigid body at the contact point.

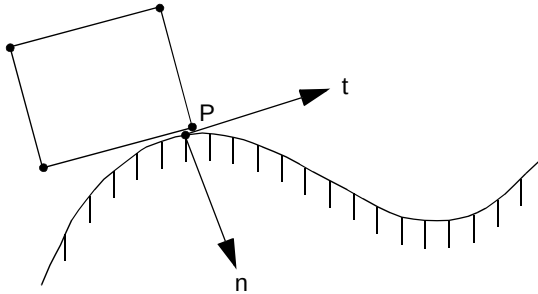


Figure 8-13 Transformed System (2-D)

As a rigid body can be represented as either a piecewise linear or as an analytical (NURBS) surface, two procedures are used. For piecewise linear representations, the normal is constant until node P comes to the corner of two segments as shown in [Figure 8-14](#). During the iteration process, one of three circumstances occur. If the angle α is small ($-\alpha_{\text{smooth}} < \alpha < \alpha_{\text{smooth}}$), the node P slides to the next segment. In such a case, the normal is updated based upon the new segment. If the angle α is large ($\alpha > \alpha_{\text{smooth}}$ or $\alpha < -\alpha_{\text{smooth}}$) the node separates from the surface if it is a convex corner, or sticks if it is a concave corner. The value of α_{smooth} is important in

controlling the computational costs. A larger value of α_{smooth} reduces the computational costs, but might lead to inaccuracies. The default values are 8.625° for 2-D and 20° for 3-D. These can be reset using the **PARAMETERS** option in the model definition or history definition section.

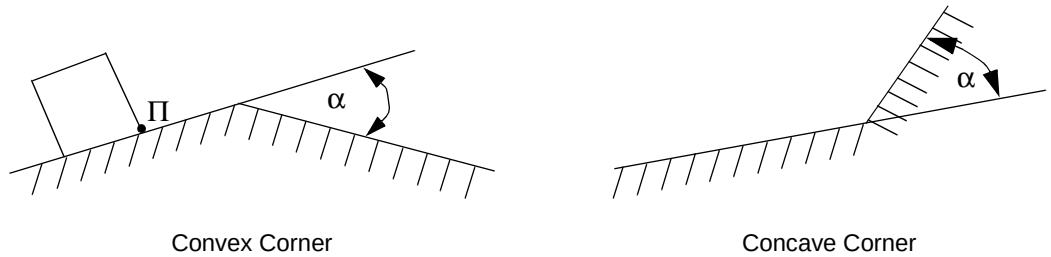


Figure 8-14 Corner Conditions (2-D)

In 3-D, these corner conditions are more complex. A node (P) on patch A slides freely until it reaches the intersection between the segments. If it is concave, the node first tries to slide along the line of intersection before moving to segment B. This is the natural (lower energy state) of motion.

These corner conditions also exist for deformable-to-deformable contact analysis. Because the bodies are continuously changing in shape, the corner conditions (sharp convex, smooth or sharp concave) are continuously being re-evaluated.

When a rigid body is represented as an analytical surface, the normal is recalculated at each iteration based upon the current position. This leads to a more accurate solution, but can be more costly because of the NURBS evaluation.

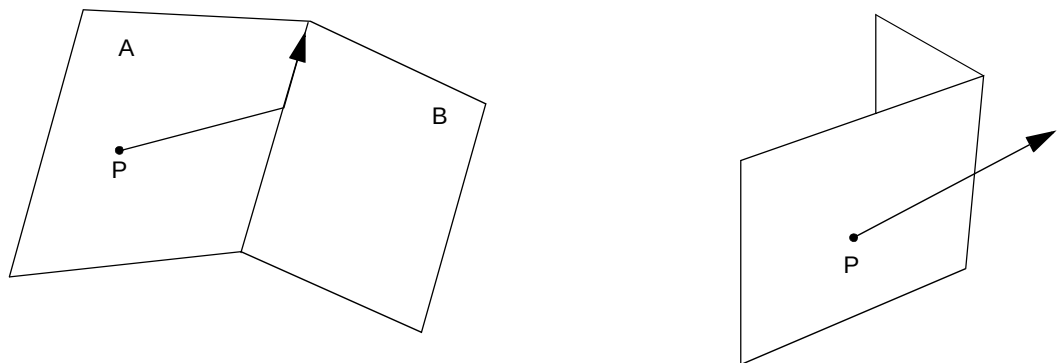


Figure 8-15 Corner Conditions (3-D)

When a node of a deformable body contacts a deformable body, a multipoint constraint (called tying) is automatically imposed. Recalling that the exterior edges (2-D) or faces (3-D) of the other deformable bodies are known, a constraint expression is formed. For 2-D analysis using lower-order elements, the number of retained nodes is three – two from the edge and the contacting node itself. For 3-D analysis, the number of retained nodes is five – four from the patch and the contacting node itself. When using higher-order elements and true quadratic contact, the number of retained nodes for 2-D becomes four, for 3-D, (hexahedrals) nine, and for 3-D (tetrahedrals) seven. The constraint equation is such that the contacting node should be able to slide on the contacted segment, subject to the current friction conditions. This leads to a nonhomogeneous, nonlinear constraint equation. In this way, a contacting node is forced to be on the contacted segment. This might introduce undesired stress changes, since a small gap or overlap between the node and the contacted segment will be closed (note that using the Single Step Houbolt operator in dynamics, forcing a node to be on the contacted segment is switched off by default, but can be activated via the [PARAMETERS](#) option). During initial detection of contact (increment 0), the stress-free projection option avoids those stress changes for deformable contact by adapting the coordinates of the contacting nodes such that they are positioned on the contacted segment. This stress-free projection can be activated using [CONTACT TABLE](#). A similar option exists for glued contact; however, in this case, overlap will not be removed.

During the iteration procedure, a node can slide from one segment to another, changing the retained nodes associated with the constraint. A recalculation of the bandwidth is automatically made. Because the bandwidth can radically change, the bandwidth optimization is also automatically performed.

A node is considered sliding off a contacted segment if it passes the end of the segment over a distance more than the contact tolerance. As mentioned earlier, the node separates from the contacted body if this happens at a convex corner. For deformable contact, this tangential tolerance at convex corners can be enlarged by using the delayed sliding off option activated via [CONTACT TABLE](#).

Friction Modeling

Friction is a complex physical phenomena that involves the characteristics of the surface such as surface roughness, temperature, normal stress, and relative velocity. The actual physics of friction continues to be a topic of research. Hence, the numerical modeling of the friction has been simplified to two idealistic models.

The most popular friction model is the Adhesive Friction or Coulomb Friction model. This model is used for most applications with the exception of bulk forming such as forging. The Coulomb model is:

$$\sigma_{fr} \leq -\mu \sigma_n \cdot t$$

where

σ_n is the normal stress

σ_{fr} is the tangential (friction) stress

μ is the friction coefficient

t is the tangential vector in the direction of the relative velocity

$$t = \frac{\mathbf{v}_r}{|\mathbf{v}_r|}$$

$\mathbf{v}_r$ is the relative sliding velocity.

The Coulomb model is also often written with respect to forces

$$f_t \leq -\mu f_n \cdot t$$

where

f_t is the tangential force

f_n is the normal reaction

Quite often in contact problems, neutral lines develop. This means that along a contact surface, the material flows in one direction in part of the surface and in the opposite direction in another part of the surface. Such neutral lines are, in general, not known a priori.

For a given normal stress, the friction stress has a step function behavior based upon the value of v_r or Δu .

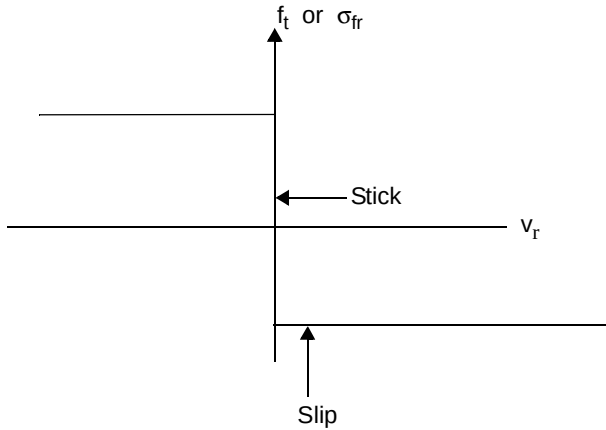


Figure 8-16 Coulomb Friction Model

This discontinuity in the value of σ_{fr} can result in numerical difficulties so a modified Coulomb friction model is implemented:

$$\sigma_{fr} \leq -\mu \sigma_n \frac{2}{\pi} \arctan\left(\frac{v_r}{RVCNST}\right) \cdot t$$

Physically, the value of $RVCNST$ is the value of the relative velocity when sliding occurs. The value of $RVCNST$ is important in determining how closely the mathematic model represents the step function. A very large value of $RVCNST$ results in a reduced value of the effective friction. A very small value results in poor convergence. It is recommended that the value of $RVCNST$ be 1% or 10% of a typical relative sliding velocity, v_r . Because of this smoothing procedure, a node in contact always has some slipping.

Besides the numerical reasons, this 'ever slipping node' model has a physical basis. Oden and Pires pointed out that for metals, there is an elasto-plastic deformation of asperities at the microscopic level (termed as 'cold weld') which leads to a nonlocal and nonlinear frictional contact behavior. The arctan representation of the friction model is a mathematical idealization of this nonlinear friction behavior.

When the Coulomb model is used with the stress based model, the integration point stresses are first extrapolated to the nodal points and then transformed so a direct component is normal to the contacted surface. The tangential stress is then evaluated and a consistent nodal force is calculated.

For shell elements, since $\sigma_n \equiv 0$ a nodal force based Coulomb model is used:

$$f_t = -\mu f_n \cdot \frac{2}{\pi} \arctan\left(\frac{v_r}{RVCNST}\right) \cdot t$$

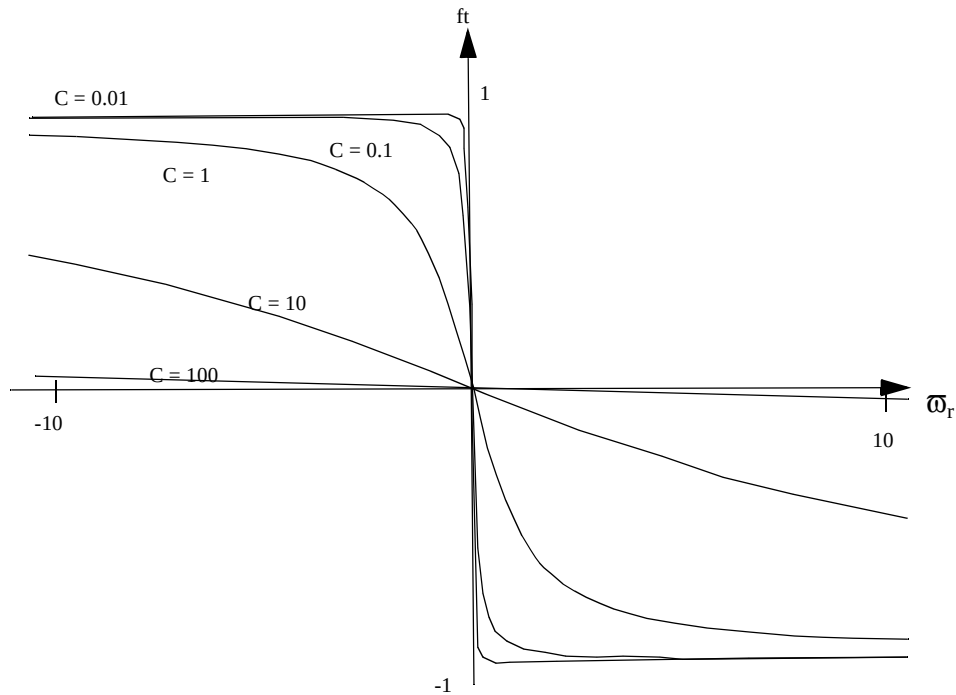


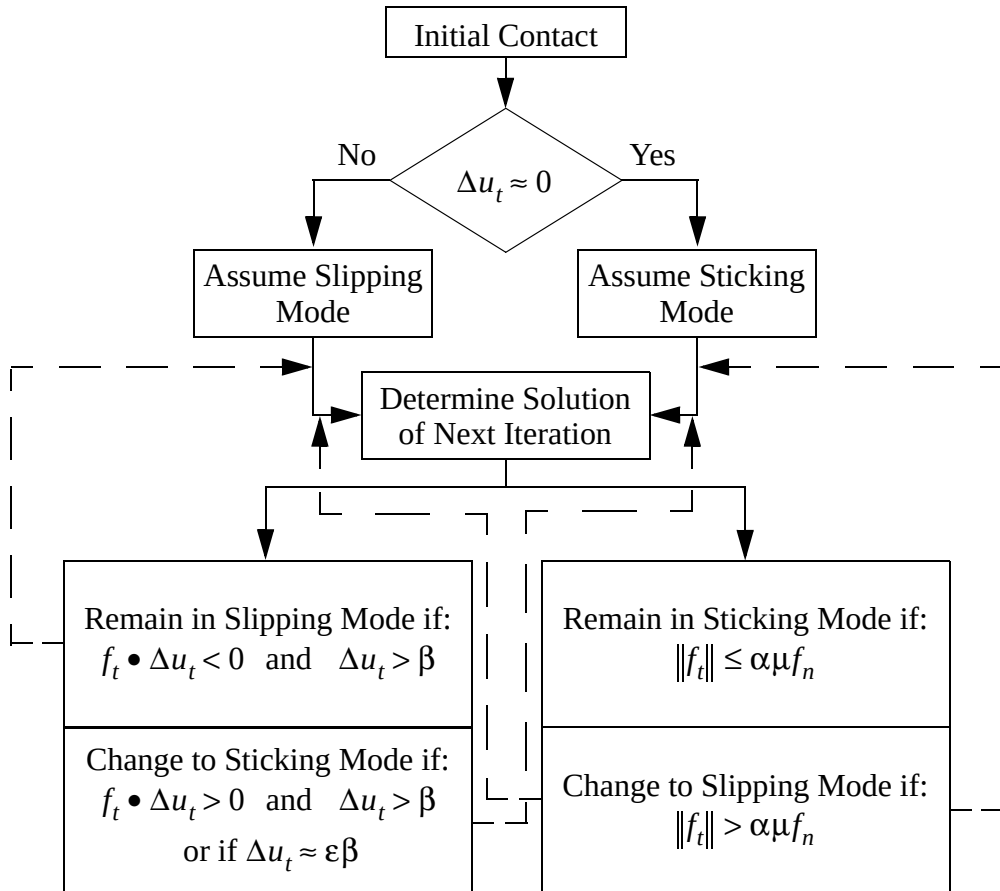
Figure 8-17 Stick-slip Approximation ($f_n = 1$, $C = RVCNST$)

This nodal forced based model should not be used if a nonlinear friction coefficient is to be used, as this nonlinearity is, in general, dependent upon the stress, not the force. This model can also be used for continuum elements.

The Coulomb friction model can also be utilized as a true stick-slip model. In this procedure, a node completely sticks to a surface until the tangential force reaches the critical value μf_n . Also, to model the differences in static versus dynamic friction coefficients, an overshoot parameter, α , can be used in the calculations.

The stick-slip model is always based upon the nodal forces.

When using the stick-slip procedure, the program flow is:



Note that this procedure requires additional testing to determine if the stick-slip condition has converged. It requires that

$$1 - e \leq \frac{f_t}{f_t^p} \leq 1 + e$$

where f_t^p is the tangential force in the previous iteration.

This additional testing on the convergence of the friction forces is not required when the smooth/continuous model is used.

The friction model can be represented as shown in [Figure 8-18](#).

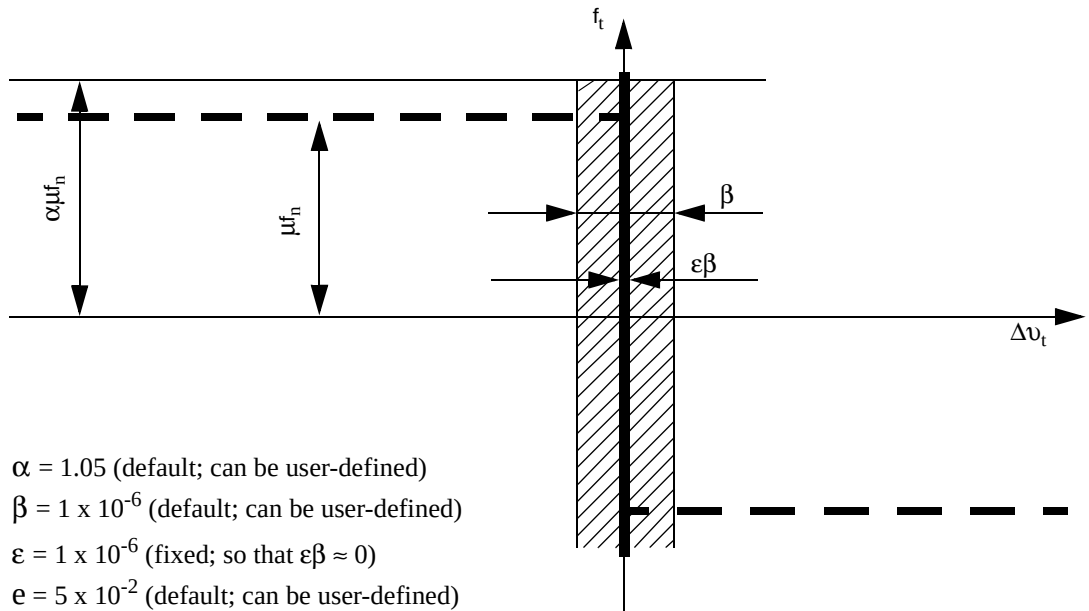


Figure 8-18 Stick-Slip Friction Parameters

Coulomb friction is a highly nonlinear phenomena dependent upon both the normal force and relative velocity. Because the Coulomb friction model is an implicit function of the velocity or displacement increment, the numerical implementation of friction has two components: a force contribution and a contribution to the stiffness matrix. The stiffness is calculated based upon:

$$K_{ij} = \frac{\partial f_{t_i}}{\partial v_{r_j}}$$

This later contribution, if fully implemented would lead to a nonsymmetric system. Because of the additional computational costs – both in terms of memory and CPU costs, the contribution to the stiffness matrix is symmetrized.

When the stress based friction model is used, the following steps are taken.

1. Extrapolate the physical stress, equivalent stress, and temperature from the integration points to the nodes using the conventional element shape functions.
2. Calculate the normal stress.
3. Calculate the relative sliding velocity. At the beginning of an increment, the previously calculated relative sliding velocity is used as the starting point. When a node first comes into contact, it is assumed that it is first sticking, so the relative sliding velocity is zero.
4. Numerically integrate the friction forces and the stiffness contribution.

For the case of deformable-deformable contact, loads equal in magnitude and opposite in direction are applied to the body that is contacted. Each of these loads is extrapolated to the closest boundary nodes. With this procedure, it is guaranteed that all friction forces applied are in self equilibrium.

The Coulomb friction model often does not correlate well with experimental observations when the normal force/stress becomes large. If the normal stress becomes large, the Coulomb model might predict that the frictional shear stresses increase to a level that can exceed the flow stress or the failure stress of the material. As this is not physically possible, the choices are either to have a nonlinear coefficient of friction or to use the cohesive, shear based friction model.

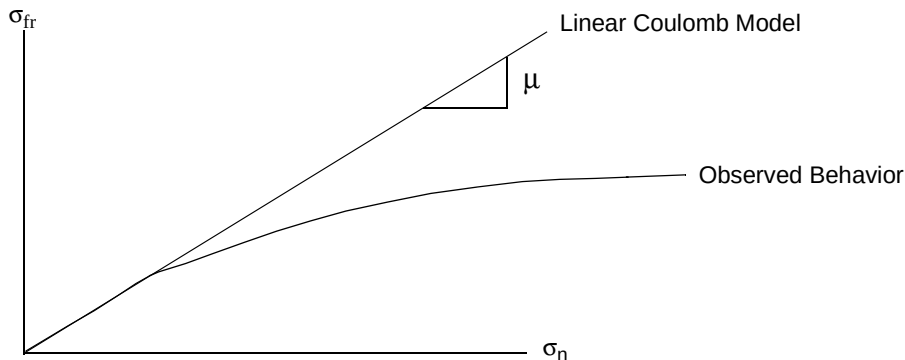


Figure 8-19 Linear Coulomb Model Versus Observed Behavior

The shear based model states that the frictional stress is a fraction of the equivalent stress in the material:

$$\sigma_{fr} \leq -m \frac{\bar{\sigma}}{\sqrt{3}} t$$

Again, this model is implemented using an arctangent function to smooth out the step function:

$$\sigma_{fr} \leq -m \frac{\bar{\sigma}}{\sqrt{3}} \cdot \frac{2}{\pi} \arctan\left(\frac{v_r}{RVCNST}\right) \cdot t$$

This model is available for all elements using the distributed load approach.

When a node contacts a rigid body, the coefficient of friction associated with the rigid body is used. When a node contacts a deformable body, the average of the coefficients for the two bodies are used. The [CONTACT TABLE](#) option can be used if complex situations occur.

Recalling that friction is a complex physical phenomena, due to variations in surface conditions, lubricant distribution, and lubricant behavior, relative sliding, temperature, geometry, and so on.

It was decided to implement the above two friction models, and to allow you to extend them, if necessary, by means of user the [UFRIC](#) subroutine. In such a routine, you provide the friction coefficient or the friction factor as

$$\mu = \mu(x, f_n, T, v_r, \sigma_y)$$

or

$$m = m(x, f_n, T, v_r, \sigma_y)$$

x – position of the point at which friction is being calculated

f_n – normal force at the point at which friction is being calculated

T – temperature at the point at which friction is being calculated

v_r – relative sliding velocity between point at which friction is being calculated and surface

σ_y – flow stress of the material

Glue Model

A special type of friction model is the glue option, which imposes that there is no relative tangential motion. The glue motion is activated through the **CONTACT TABLE** option.

A novel application of contact is to join two dissimilar meshes. In such a case, by specifying a very large separation force and that the glue motion is activated, the constraint equations are automatically written between the two meshes.

Separation

After a node comes into contact with a surface, it is possible for it to separate in a subsequent iteration or increment. Mathematically, a node should separate when the reaction force between the node and surface becomes tensile or positive. Physically, you could consider that a node should separate when the tensile force or normal stress exceeds the surface tension. Rather than use an exact mathematical definition, you can enter the force or stress required to cause separation.

Separation can be based upon either the nodal forces or the nodal stresses. The use of the nodal stress method is recommended as the influence of element size is eliminated.

When using higher-order elements and true quadratic contact, the separation criteria must be based upon the stresses as the equivalent nodal forces oscillate between the corner and midside nodes.

In many analysis, contact occurs but the contact forces are small; for example, laying a piece of paper on a desk. Because of the finite element procedure, this could result in numerical chattering. MSC.Marc has some additional contact control parameters that can be used to minimize this problem. As separation results in additional iterations (which leads to higher costs), the appropriate choice of parameters can be very beneficial.

When contact occurs, a reaction force associated with the node in contact balances the internal stress of the elements adjacent to this node. When separation occurs, this reaction force behaves as a residual force (as the force on a free node should be zero). This requires that the internal stresses in the deformable body be redistributed. Depending on the magnitude of the force, this might require several iterations.

You should note that in static analysis, if a deformable body is constrained only by other bodies (no explicit boundary conditions) and the body subsequently separates from all other bodies, it would then have rigid body motion. For static analysis, this would result in a singular or nonpositive definite system. This problem can be avoided by appropriate boundary conditions.

Release

A special case of separation is the intentional release of all nodes from a rigid body. This is often used in manufacturing analysis to simulate the removal of the workpiece from the tools. After the release occurs in such an analysis, there might be a large redistribution of the loads. It is possible to gradually reduce the residual force to zero, which improves the stability, and reduces the number of iterations required. The **RELEASE** history definition option allows the release (separation) of all the nodes in contact with a particular surface at the beginning of the increment. The rigid body should be moved away using the **MOTION CHANGE** option or deactivated using the **CONTACT TABLE** option to ensure that the nodes do not inadvertently recontact the surface they were released from.

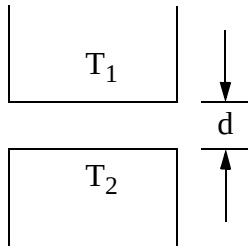
Coupled Analysis

Contact analysis has several consequences on performing a coupled analysis. When performing a coupled contact analysis between multiple deformable bodies, each body deforms due to mechanical and thermal loads and undergoes heat conduction. When the bodies come into contact, there is heat flux across the surfaces. You need to provide a coefficient of heat transfer between the surfaces. This is often quite significant as a hot workpiece might come into contact with a cold tool set. Additionally, if friction is present, heat is generated based upon:

$$q = f_{fr} \cdot v_r \cdot M_{eq}$$

where M_{eq} is the mechanical equivalent of heat. Half of the heat generated due to friction is contributed to each body. If the model allows a body to be rigid, two options are available. In the first case, the rigid body is considered to have a constant temperature (heat source or heat sink). Heat flux is exchanged between the rigid body and the deformable body. In the second case, it is required to perform a heat transfer analysis in the rigid body. In this case, the rigid body must be constructed of finite elements, but they are chosen as heat transfer types. Deformation does not occur in this body, but a heat transfer analysis is performed. This is computationally more efficient than performing a coupled analyses where all bodies are deformable. Symmetry bodies have different characteristics as they are frictionless (hence, no heat generated) and because there is no heat flux across a symmetry plane.

Heat fluxes (FILMS) are automatically created on all the boundaries of the deformable-bodies. Three types of heat fluxes can be distinguished, where each flux has its own physical characteristic. The type of flux MSC.Marc uses, depends on the distance between the contacting bodies (see [Figure 8-20](#)). We define the distance, $d_{contact}$, as the distance below which the two bodies are touching each other. For a distance smaller than $d_{contact}$, the first flux type is used. The second flux type is used when the distance is between $d_{contact}$ and d_{near} , where d_{near} is called the near contact distance. This distance represents a small gap between the contact bodies, and should not be chosen larger than the smallest element size. If this distance is not set, this type of flux is disregarded. The third flux type is used when the distance is larger than d_{near} .

**Figure 8-20** Contact Distance

If $d < d_{contact}$, the bodies touch and conduction takes place between the two contacting bodies. The heat flux is written as:

$$q = H_{TC}(T_2 - T_1)$$

where,

T_1 is the surface temperature,

H_{TC} is the film coefficient between the two surfaces,

T_2 is the temperature of the same contact location, as obtained from interpolation of nodal temperatures of the body being contacted.

If $d_{contact} < d < d_{near}$, a small gap between the bodies exists. The heat flux that exists through this gap can be decomposed in the following physical processes: convection, natural convection, radiation, and a distance dependent term. This flux is given by:

$$q = H_{CV}(T_2 - T_1) + H_{NC}(T_2 - T_1)^{B_{NC}} + \sigma \epsilon f(TA_2^4 - TA_1^4) + \left\{ H_{CT} \left(1 - \frac{d}{d_{near}} \right) + H_{BL} \frac{d}{d_{near}} \right\} (T_2 - T_1)$$

where,

H_{CV} is the convection coefficient for near field behavior.

H_{NC} is the natural convection coefficient for near field behavior.

- B_{CN} is the exponent associated with natural convection.
- $\sigma \epsilon f$ are the coefficients for radiation, the Stefan-Boltzman coefficient, the emissivity, and the viewfactor, respectively. The Stefan-Boltzmann constant is given via the [PARAMETERS](#) option.
- H_{BL} is the separation distance dependent heat transfer coefficient.
- TA_2 $T_2 + T_{AB^sREF}$ is the temperature in absolute scale. The offset between the user temperature and the absolute temperature is given in the [PARAMETERS](#) option.

If $d > d_{near}$, the bodies are not touching and, at both boundaries, a heat flux to the environment exists in the following form,

$$q = H_{CTVE}(T_2 - T_{SINK})$$

where,

H_{CTVE} is the heat transfer coefficient to the environment

T_{SINK} is the environment sink temperature

When a deformable-body contacts a rigid-body, the coefficients associated with the rigid-body are used. When two deformable bodies are in contact, the average value of the film coefficient specified for the bodies is used. This only works for the coefficients discussed in the first and third heat flux type. A better way to prescribe the coefficients is to use the [CONTACT TABLE](#) option. With this option, all the previously discussed coefficients can be set for each interface separately. For the second heat flux type, this is the only way to specify the coefficients. As with all other coupled problems, heat generated by plastic deformation can be calculated and applied as a volumetric flux. The heat generated by friction is also calculated and applied as a surface flux.

Three user subroutines are available to facilitate the creation of more sophisticated boundary flux definitions (such as radiation and convections with variations in space, temperature, pressure, etc.). [UHTCON](#) allows you to specify a film coefficient while the surface is in contact ($d < d_{contact}$), [UHTNRC](#) allows you to specify a film coefficient while the surface is almost in contact ($d_{contact} < d < d_{near}$), and [UHTCOE](#) allows you to specify a film coefficient while the surface is free ($d > d_{near}$).

Thermal Contact

A thermal contact analysis is comparable to a coupled analysis except that in this case the mechanical part is omitted, resulting in a computationally more efficient analysis. The two types of rigid bodies that are described in the coupled analysis are allowed. So one is a rigid which is considered to have a constant temperature (heat source or heat sink), and the other is a finite element body with thermal properties. The three types of heat fluxes as described in [Coupled Analysis](#) can also be used here.

The coefficients are prescribed in the same way as in a [Coupled Analysis](#) where it is advised to use the [CONTACT TABLE](#) option.

The three user subroutines [UHTCON](#), [UHTNRC](#), and [UHTCOE](#) as previously discussed in [Coupled Analysis](#) can also be used in a thermal contact analysis.

Element Considerations

MSC.Marc allows contact with almost all of the available elements, but the use of certain elements has a consequence on the analysis procedure. Contact analysis can be performed with all of the structural continuum elements, either lower order or higher order, including those of the Herrmann (incompressible) formulation, except axisymmetric elements with twist. Friction modeling is available in all of these elements except the semi-infinite elements. Higher-order isoparametric elements use shape functions which, when the elements are loaded by a (for example) uniform pressure, lead to equivalent nodal loads that oscillate between the corner and midside nodes. This has a detrimental effect on determining contact separation and two procedures have been implemented to eliminate this problem:

1. *Linearized Contact:* The midside nodes on the exterior surface are automatically tied to the corner nodes in this case. This effectively results in a linear variation of both the geometry and the displacement on the exterior element edges. Hence, the elements with edges on the exterior do not behave as full bi-quadratic (2-D) or tri-quadratic (3-D) elements. All elements in the interior of the body behave in the conventional higher-order manner, but the constraints on the exterior easily cause the behavior of the complete structure to be too stiff; while in the area of contact, the stress distribution might be irregular.
2. *True Quadratic Contact:* No special constraints introduced on the exterior surface other than coming from contact in this case. Both the midside and the corner nodes may come into contact and when contact is established with another deformable body consisting of quadratic elements, a constraint equation corresponding to the complete quadratic shape function is automatically incorporated. Since the above mentioned oscillating nodal loads cannot be used for separation, the decision whether or not a node should separate is based on the contact normal stress rather than the contact normal force.

In many manufacturing and rubber analyses, the lower-order elements behave better than the higher-order elements because of their ability to represent the large distortion; hence, these lower-order elements are recommended.

The constraints imposed on the nodal degrees of freedom are dependent upon the type of element.

1. When a node of a continuum element comes into contact, the translational degrees of freedom are constrained.

2. When a node of a shell element comes into contact, the translational degrees of freedom are constrained and no constraint is placed on the rotational degrees of freedom. The exception to this is when a shell contacts a symmetry surface. In this case, the rotation about the element edge is also constrained.

Additionally, beams and shells contact is governed by the rules outlined below.

2-D Beams

All nodes on beams are potential contact nodes. Beam elements can be used in contact in two modes.

1. The two-dimensional beams can come into contact with rigid bodies composed of curves in the same x-y plane. The normal is based upon the normal of the rigid surface.
2. The two-dimensional beams can come into contact with deformable bodies either of continuum elements or other beam elements. As the beams are in two dimensions, they do not intersect one another.

3-D Beams

Three-dimensional beam elements can be used in contact in three modes.

1. The nodes of the beams can come into contact with rigid bodies composed of surfaces. The normal is based upon the normal of the rigid surface.
2. The nodes of the beam and truss elements can also come into contact with the faces of three-dimensional continuum elements or shell elements. The normal is based upon the normal of the element face.
3. The three-dimensional beam and truss elements can also come into contact with other beam or truss elements (beam-to-beam contact).

The first two modes are activated by default if contact bodies consisting of beam or truss elements are defined using the **CONTACT** option. The third mode must be activated explicitly by additionally switching on the beam-to-beam contact flag on the **CONTACT** option.

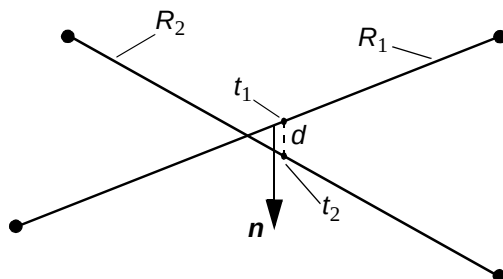


Figure 8-21 Beam-to-Beam Contact

In the beam-to-beam contact model, a beam or truss element is viewed as a straight cylinder with a circular cross-section. The radius of the cross-section, the *contact radius* of the element, must be entered via the [GEOMETRY](#) option. Contact is detected between two beam or truss elements if the associated cylinders touch each other; that is, if the distance d between the points t_1 on the first element and t_2 on the second element that are closest to each other is within the contact tolerance of the sum of the contact radii R_1 and R_2 of the elements,

$$R_1 + R_2 - (1 + \text{Bias}) * \text{tolerance} < d < R_1 + R_2 + (1 - \text{Bias}) * \text{tolerance} .$$

The points t_1 and t_2 are the contacting points of the elements. The normal vector $\mathbf{n}$ is the direction of the relative position of t_2 with respect to t_1 (see [Figure 8-21](#)).

If two beam or truss elements are in contact, a multipoint constraint (tying) in the direction of the normal vector is automatically imposed to ensure that the distance between the beams remains equal to the sum of the contact radii. The constraint equation involves the displacements of the beginning and ending nodes of both elements. The tied node in that equation is automatically selected, taking into account the location of the contacting points on the elements, any boundary conditions applied to the nodes and any contact between the nodes and rigid surfaces or faces of continuum or shell elements.

During the iteration procedure, the contacting points of two beam or truss elements can change if the elements slide with respect to each other. In addition, the points in contact can move from one element to another. In that case, the nodes involved in the multipoint constraint is automatically updated. During sliding, friction may be taken into account. Since the normal stress in the contact points is not available, only Coulomb friction is supported for beam-to-beam contact. The glue model, which imposes that there is no relative tangential motion, is also available.

The limitations of the beam-to-beam contact model are:

1. The contact radius must be the same for all elements in a contact body.
2. A contact body cannot contain branches, that is, every element in the contact body must have a unique successor and predecessor.

Shell Elements

All nodes on shell elements are potential contact nodes. As the midside nodes of shell elements are automatically tied, the high-order shell element (type 22) has no benefit. Shell elements can contact either rigid bodies, continuum elements, or other shell elements. Shell-shell contact involves a more complex analysis because it is necessary to determine which side of the shell contact occurs.

Dynamic Impact

The capability is available for both implicit time integration via the Newmark-beta or Houbolt operators, and the explicit time integration via the (fast) central difference operator.

The Newmark-beta and the Single Step Houbolt procedure have the capability to allow variable time steps and, when using the user-defined fixed time step procedure, the time step is split by the algorithm to satisfy the contact conditions. Since the central difference operator is constrained to small time steps by the stability requirement, time splitting is not used by the contact algorithm.

For most dynamic impact problems, the Single Step Houbolt method is recommended, as this procedure possesses high-frequency dissipation. This is often necessary to avoid numerical problems by contact-induced high-frequency oscillations. If the other dynamic operators are used, it is recommended that numerical damping be used during the analysis.

In dynamic analysis, the requirement of energy conservation is supplemented with the requirement of momentum conservation. In addition to the constraints placed upon the displacements, additional constraints are placed on the velocity and acceleration of the nodal points in contact, except for the Single Step Houbolt method.

When a node contacts a rigid surface, it is given the velocity and acceleration of the rigid surface in the normal direction. The rigid surfaces are treated as if they have infinite mass, hence, infinite momentum.

Global Adaptive Meshing and Rezoning

In manufacturing simulations the objective is to deform the workpiece from some initial (simple) shape to a final, often complex shape. This deformation of the material, results in mesh distortion. For this reason, it is often required to perform a rezoning/remeshing step. At this point, a new mesh is created; the current state of deformation, strains, stresses, etc. is transferred to the new mesh, and the analysis is continued. This has several consequences for contact analysis.

1. When manual rezoning is used, it is necessary to redefine the elements associated with the deformable body that is being rezoned. This is done through the [CONTACT CHANGE](#) option.
2. If a contact tolerance was not defined by you, a new contact tolerance is calculated based upon the new mesh. This distance can be smaller than the previously calculated tolerance, leading to more iterations.
3. After the new mesh is created, the program redetermines the potential contact nodes. In the current release of MSC.Marc, it is necessary to give an upper bound to this number to insure sufficient memory allocation. Additionally, as a rezoning analysis increases the number of nodes, elements, and boundary conditions, it is necessary to reserve space for these features from the beginning of the analysis. This is done through the [SIZING](#) or [ADAPTIVE](#) parameter.
4. At the first increment after remeshing, MSC.Marc determines which nodes are in contact. If the new mesh does not reflect the exterior surface of the old model, it is possible that a region that was previously in contact is no longer in contact. This is usually not so serious; as most likely, the node subsequently comes into contact. A more significant problem is if a node in the new mesh is created which, in fact, has penetrated another body. If the newly created node is beyond another body's surface by more than the contact tolerance, it is possible that a poor contact analysis results.

If rezoning occurs, rigid surfaces must be kept the same and cannot be changed.

MSC.Marc auto-rezoning feature, activated by the parameter [REZONING](#),1 for 2-D problems and [REZONING](#),2 for 3-D problems, automatically creates new meshes and performs rezoning to resolve mesh distortion problems. In such cases, points (1) and (4) are treated automatically.

Adaptive Meshing

Contact between a deformable-body and a rigid body is insured such that the nodes do not penetrate the rigid surface. It is possible that an edge of an element penetrates a rigid surface, especially where high curvature is present because of the finite element discretization. The use of the adaptive mesh generation procedure can be used to reduce these problems.

In addition to the traditional error criteria such as Zienkiewicz-Zhu or maximum stress, etc., you can request that the mesh be adaptively refined due to contact. In such a case, when a node comes into contact, the elements associated with that node are refined. This results in a greater number of elements and nodes on the exterior region where contact occurs. This can lead to a substantial improvement in the accuracy of the solution.

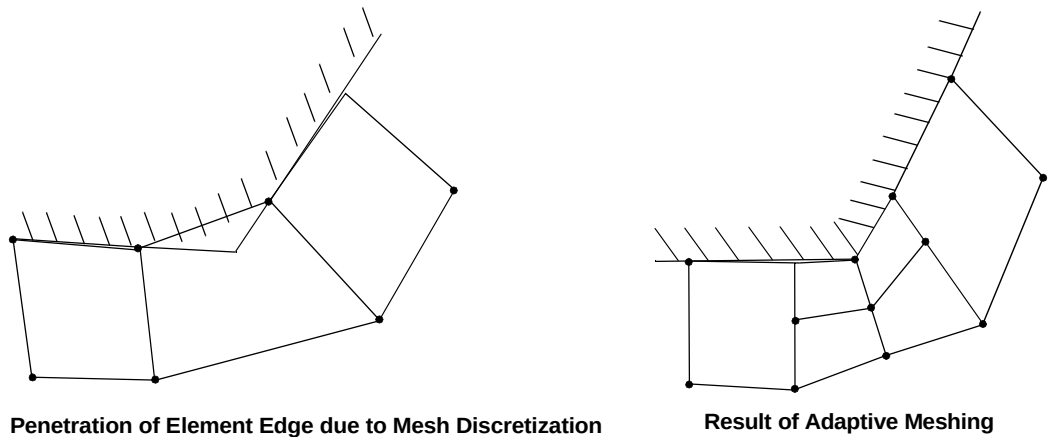


Figure 8-22 Contact Closure Condition

The adaptive meshing procedure has consequences to the contact procedure similar to the rezoning procedure.

1. If a contact tolerance was not defined by you, a new contact tolerance is calculated based upon the new mesh. This distance can be smaller than the previously calculated tolerance, leading to more iterations.
2. After the new mesh is created, the program redetermines the potential contact nodes. In the current release of MSC.Marc, it is necessary to give an upper bound to this number to insure sufficient memory allocation. Additionally, as a rezoning analysis increases the number of nodes, elements, and boundary conditions, it is necessary to reserve space for these features from the beginning of the analysis. This is done through the **ADAPTIVE** parameter.

Result Evaluation

The MSC.Marc post file contains the results for both the deformable bodies and the rigid bodies. In performing a contact analysis, you can obtain three types of results. The first is the conventional results from the deformable body. This includes the deformation, strains, stresses, and measures of inelastic behavior such as plastic and creep strains. In addition to reaction forces at conventional boundary conditions, you can obtain the contact forces and friction forces imparted on the body by rigid or other deformable bodies. By examining the location of these forces, you can observe where contact has occurred, but MSC.Marc also allows you to select the contact status as a post file variable:

A value of 0 means that a node is not in contact.

A value of 0.5 means the node is in near thermal contact.

A value of 1 means that a node is in contact.

A value of 2 means the node is on a cyclic symmetry boundary.

It is also possible to obtain the resultant force following from contact on the deformable bodies and the resultant force and moment on the rigid bodies. The moment is taken about the user-defined centroid of the rigid body. The time history of these resultant forces are of significant issues in many engineering analysis. Of course, if there is no resultant force on a rigid body, it implies that body is not in contact with any deformable body.

Finally, if the additional print is requested, the output file reflects information on when a node comes into contact, what rigid body/segment is contacted, when separation occurs, when a node contacts a sharp corner, the displacement in the local coordinate system, and the contact force in the local coordinate system. For large problems, this can result in a significant amount of output.

The motion of the rigid bodies can be displayed as well as the deformable bodies. Rigid bodies which are modeled using the piecewise linear approach are displayed as line segments for flat patches. When the rigid surfaces are modeled as analytical surfaces, the visualization appears as piecewise linear in older versions (pre-K7/Mentat 3.1) or as trimmed NURBS in subsequent versions.

Tolerance Values

On the third data block for contact, five tolerances can be set for determination of the contact behavior. Not entering any values here means that MSC.Marc calculates values based on the problem specification.

Relative Sliding Velocity Between Surfaces Below Which Friction Forces Drop

As discussed in the [Friction Modeling](#) section, the equations of friction are smoothed internally in the program to avoid numerical instabilities. The equations are inequalities whenever two contacting surfaces stick to each other and equalities whenever the surfaces slide (or slip). Thus, the character of contact constraints change depending on whether there is sticking or slipping. The smoothing procedure consist of modifying it in such a way, that there is always slip; the amount is a function of the relative velocity and a constant RVCNST. The value of this constant must be specified here. It actually means, that if we specify a small value in comparison to the relative velocity, the jump behavior is better approximated, but numerical instabilities can be expected. A large value means, that we need a large relative velocity before we get the force at which the slip occurs.

It is suggested to use values between 0.1 and 0.01 times a typical surface velocity.

Distance Below Which a Node is Considered Touching a Surface

In each step, it is checked whether a (new) node is in contact with other surfaces. This is determined by the distance between the nodes and the surfaces. Since the distance is a calculated number, there are always roundoff errors involved. Therefore, a contact tolerance is provided such that if the distance calculated is below this tolerance, a node is considered in contact. A too large value means that a high number of body nodes are considered to be in contact with the surface and are consequently all moved to the surface, which can be unrealistic in some applications. A too small value of this number means that the applied deformation increment is split into a high number of increments, thus increasing the cost of computation.

The tolerance must be provided by the analyst or can be calculated by MSC.Marc. In general, the contact tolerance should be a small number compared to the geometrical features of the configuration being analyzed. The value calculated by MSC.Marc is determined as 1/20 of the smallest element size for solid elements or 1/4 of the thickness of shell elements. If both shell and continuum elements are present, the default is based upon the smaller of the two values.

Tolerance on Nodal Reaction Force on Nodal Stress Before Separation Occurs

If a tensile force occurs at a node which is in contact with a surface, the node should separate from the surface. Rather than using any positive value, a threshold value can be specified. This number should theoretically be zero. However, because a small positive reaction might be due only to errors in equilibrium, this threshold value avoids unnecessary separations. A too small value of this force results in alternating separation and contact between the node and the surface. A too large value, of course, results in unrealistic contact behavior.

MSC.Marc calculates this value as the maximum residual force in the structure. Note that the maximum residual force can be specified in the [CONTROL](#) option. Default for this value, a 10 percent of the maximum reaction force is used. Consequently, if locally high reaction forces at a particular point are present, the separation force is large as well. In most cases, however, the default value is a good measure.

If you indicate that separation is to be based upon stresses, a value of the separation stress should be entered. The default value is the maximum residual force at node n divided by the contact area of node n .

Workspace Reservation

In the definition of the contact behavior, a number of maximum workspace numbers must be set for the workspace reservation in MSC.Marc. It would, theoretically, be possible to automate this as well, but since in a restart analysis with mesh rezoning or adaptive meshing for instance, it might be necessary to increase the number of elements of the deforming body, this is not possible for all cases. Therefore, you must specify some numbers here. Too large values of the numbers means that the internal workspace for MSC.Marc is not optimally used. Too small numbers are indicated at the start of the analysis by MSC.Marc, but if it occurs in a restart, it can mean that the complete analysis has to be repeated.

The numbers which have to be set are the following:

- number of bodies
This is immediately clear from the surface definition.
- upper bound to the number of boundary nodes of any deformable surface
This number is, of course, closely related to the previous number for deformable surfaces.

In cases where the maximums are difficult to determine beforehand, it is always possible to perform the analysis with a **STOP** parameter. MSC.Marc tells you the numbers which are to be used. Again, be careful for possible restarts with mesh rezoning or adaptive meshing.

Mathematical Aspects of Contact

Contact can be defined as finding the displacement of points A and B such that

$$(u_A - u_B) \cdot n < \text{TOL}$$

where A is on one body and B is on another body, n is the direction cosine of a vector between the two points, and TOL is the closure distance.

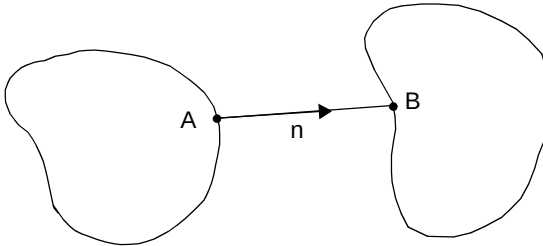


Figure 8-23 Normal Gap Between Potentially Contacting Bodies

Lagrange Multiplier Procedure

In the Lagrange Multiplier Procedure, the variational procedure is augmented by the constraint. The virtual work principle leads to the traditional systems equation:

$$Ku = f$$

The constraint conditions can be expressed as

$$Cu = 0$$

Through the minimization of the augmented functional

$$\Psi = \frac{1}{2}u^T Ku - u^T f + \lambda^T Cu$$

you obtain

$$\begin{bmatrix} K & C^T \\ C & 0 \end{bmatrix} \begin{Bmatrix} u \\ \lambda \end{Bmatrix} = \begin{Bmatrix} f \\ 0 \end{Bmatrix}$$

This equation can be solved simultaneously for both the displacement (u) and the Lagrange multiplier (λ). You observe the introduction of Lagrange multipliers results in a zero on the diagonal. Hence, even well posed physical problems no longer have positive definite systems. This often results in numerical difficulties and restricts the type of linear equation solver that can be used.

Direct Constraint Procedure

MSC.Marc divides contact problems into two domains; the first is when a deformable body makes contact with a rigid surface and the second is when a deformable body makes contact with another deformable body or itself.

Two-dimensional or three-dimensional bodies are treated conceptually in the same manner.

Deformable-Rigid Contact

In such a problem, a target node on the deformable body has no constraint while contact does not occur. Once contact is detected, the degrees of freedom are transformed to a local system and a constraint is imposed such that

$$\Delta u_{normal} = v \cdot n$$

where v is the prescribed velocity of the rigid surface. This local transformation is continuously updated to reflect sliding of point A along the rigid surface.

If the glue option is activated, an additional displacement constraint is formed as

$$\Delta u_{tangential} = v \cdot t$$

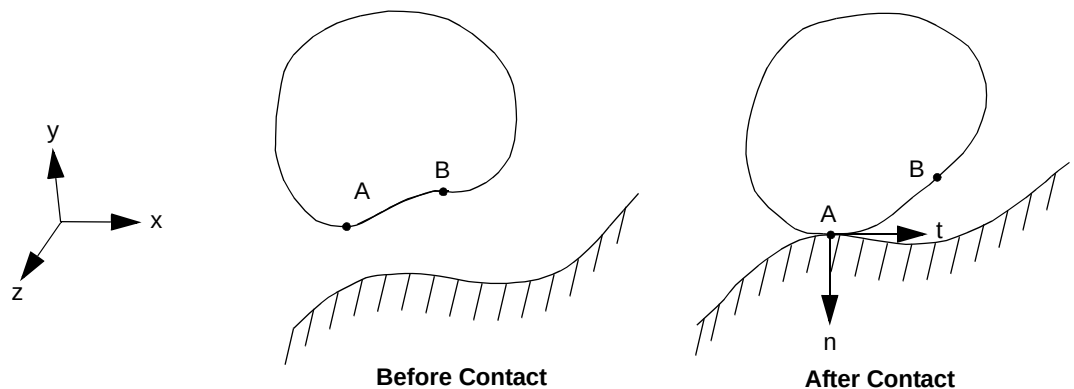


Figure 8-24 Contact Coordinate System

The determination of when contact occurs and the calculation of the normal vector are critical to the numerical simulation. In MSC.Marc, three procedures are used for detecting penetration and imposing contact. In [Figure 8-25](#), node A is not in contact with the rigid body at the beginning of the increment (n). After time step Δt , node A penetrates rigid surface if no constraint is imposed in step $n + 1$.

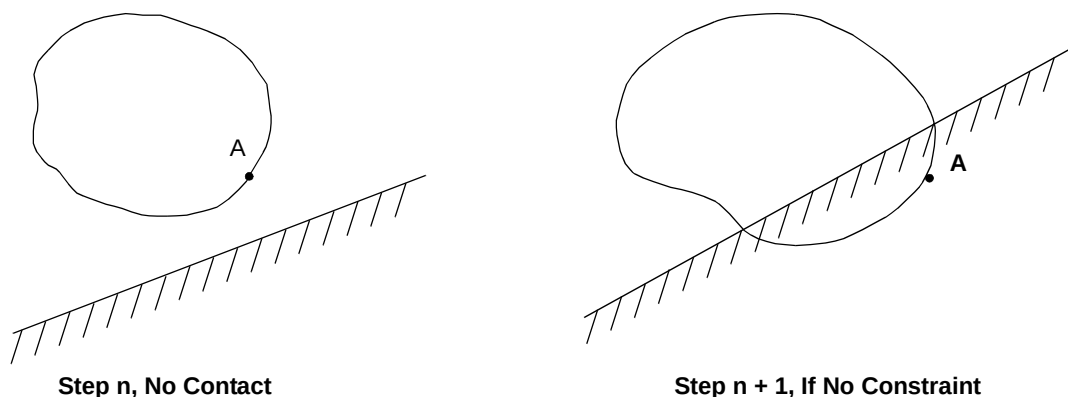


Figure 8-25 Contact Constraint Requirement

This is clearly not an acceptable solution. The first two procedures discussed below are used when the fixed time step procedure is used. In the first procedure, the time step Δt is divided into two periods or subincrements ([Figure 8-26](#)). In the first subincrement, node A is unconstrained such that its motion is unimpeded. In the second subincrement, the motion is constrained. The numerical procedure now needs to find the time period when contact first occurs. This is a nonlinear mathematical problem and the time step is chosen by linearizing the displacement increment. This results in potential penetration or only approximately reaching the contact surface.

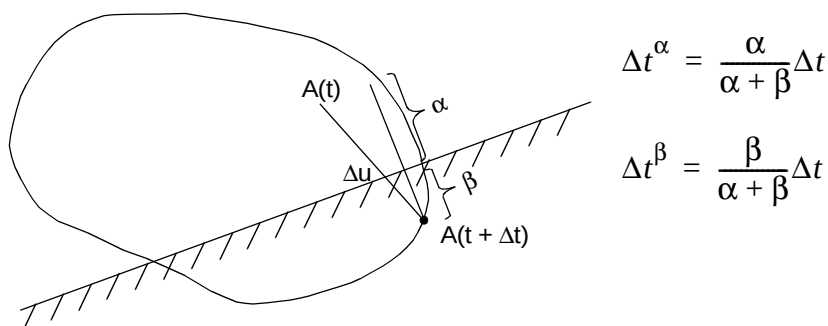


Figure 8-26 Strategy for Increment Splitting

In this approach, during each subincrement, node A is unconstrained for time period Δt^α and constrained for time period Δt^β .

In the second approach labeled iterative penetration checking, the change of the contact condition is treated within the Newton-Raphson iteration loop. In this method, the incremental displacements are scaled such that contact occurs. This effectively brings node A back onto the rigid surface.

The residuals are then calculated and the boundary conditions are updated such that node A is constrained and the iteration procedure is continued such that both equilibrium is achieved and the contact conditions are satisfied.

The third procedure is when the adaptive time stepping procedure **AUTO STEP** is used. MSC.Marc reduces the time step in the increment such that the node just comes into contact. In the next increment, the node is given the appropriate contact constraint and a new time step is determined based upon the convergence criteria.

Analytical Contact

Utilizing the analytical representation of rigid surfaces influences the numerical procedure in several ways:

1. The rigid surface is represented by a nonuniform rational B-spline surface (NURBS) which provides a precise mathematical form capable of representing the common analytical shapes – circle, conic curve, free-form curves, surfaces of revolution, and sculptured surfaces.

The surface can be expressed in simple mathematical form to model complex multiple surfaces with advantage of continuity in first and second derivative. Those analytical forms are expressed in four-dimensional homogeneous coordinate space by:

$$P(u, v) = \frac{\sum_{i=1}^{n+1} \sum_{j=1}^{m+1} B_{i,j} h_{i,j} N_{i,k}(u) M_{j,l}(v)}{\sum_{i=1}^{n+1} \sum_{j=1}^{m+1} h_{i,j} N_{i,k}(u) M_{j,l}(v)}$$

For curves, it simplifies to:

$$P(u) = \frac{\sum_{i=1}^{n+1} B_i h_i N_{i,k}(u)}{\sum_{i=1}^{n+1} h_i N_{i,k}(u)}$$

where the B are 4-D homogeneous defining polygon vertices, $N_{i,k}$ and $M_{j,l}$ are nonrational B-spline basis functions, $h_{i,j}$ is homogeneous coordinates. For given parameters u and v in local system, the location (x , y , z) in three-dimensional space, first derivative and second derivative (if required), are calculated. Given a point with x , y and z Cartesian coordinates, there is no explicit mathematical form to find out the parameters u and v in the local space, so they are obtained using an iterative procedure.

2. The location of the closest point on the contact surface corresponding to node A needs to be determined. This process, determined analytically when using flat patches, requires an iterative approach for rigid bodies modeled with NURBS.

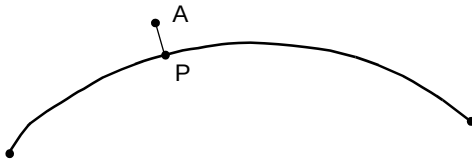


Figure 8-27 Closest Point Projection Algorithm

3. Once the point P is known, the normal is calculated based upon the NURBS. This normal is used in a manner consistent with the PWL procedure.
4. Because the normal can change from iteration to iteration, the imposition of the kinematic boundary conditions is continuously changing.

In the PWL approach:

$$(\Delta u_n)^0 = \mathbf{v} \cdot \mathbf{n}$$

$$(\delta u_n)^i = 0$$

In the analytical approach:

$$(\Delta u_n)^0 = \mathbf{v} \cdot \mathbf{n}_0$$

$$(\delta u_n)^i = \mathbf{v} \cdot \mathbf{n}_1 - \mathbf{v} \cdot \mathbf{n}_0$$

where the subscripts represent the iteration number.

5. In the PWL approach, when a node slides from one segment to another, the corner condition logic is activated. The advantages of analytical surfaces is that this logic is not necessary until you come to the end of the spline. This reduces the amount of iterations required.
6. The friction calculation is dependent upon the surface normal and tangent. When using the analytical approach, the friction calculation includes the effect of changes in the direction of the normal vector from iteration to iteration. This improves the accuracy and convergence behavior.
7. Because the normal is continuous, the calculation of nodal forces is more accurate. This allows for the determination of nodal separation to occur beginning with the second iteration as opposed to when equilibrium convergence has been achieved using the PWL approach.

Solution Strategy for Rigid Contact

1. At the start of an increment, all boundary nodes are checked for contact with surfaces and flagged if so.
2. Next, transformations and imposed displacements are determined for each touching node. With the knowledge of the time increment and surface velocity, the configuration of the surface at the end of the increment is found. Then, the distance from the node starting position to the current surface segment is determined and is the probable normal displacement to be imposed. If there is a previous solution for displacement increments, these are used to estimate the tangential displacement increment (Figure 8-28). If this estimate still puts the node at the end of the increment in the same surface segment, then this one is used to determine the

transformation matrix between local and global coordinates, and the normal displacement is accepted. Otherwise, the procedure is repeated for the adjacent segment. If a concave corner is hit, the boundary node is fixed to such corner. If the node goes out of a sharp convex corner, it is immediately released from the surface.

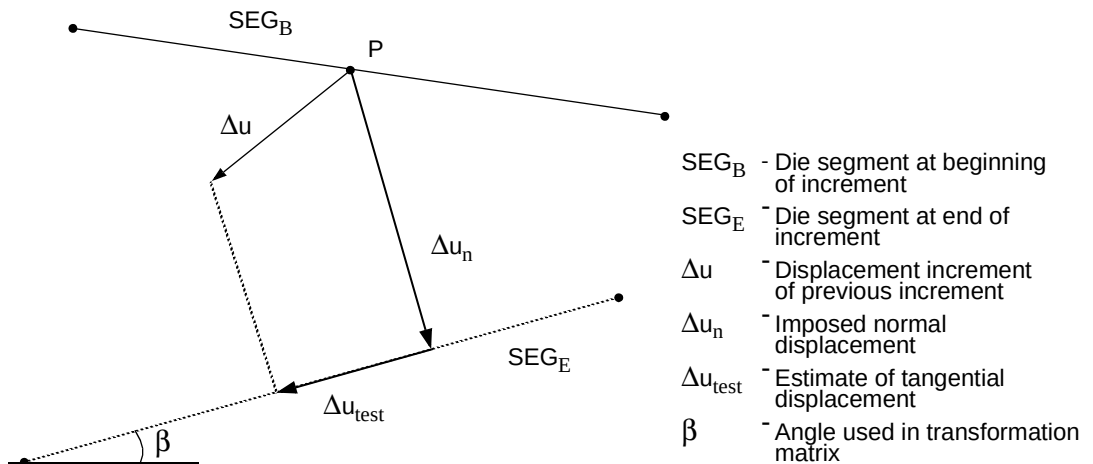


Figure 8-28 Node P Already In Contact (2-D)

3. One increment of the problem is solved iteratively and values for displacement increments are calculated. [Step 2](#) is repeated at each iteration for possible changes in surface segment.
4. Once convergence is obtained, nodal forces are checked. Whenever a positive normal force (or stress) is detected which is larger than the tolerance, a node is released from the surface segment at which such force (or stress) corresponds and [Step 3](#) is repeated. This tolerance can be modified using the user subroutine [SEPFOR](#) (or [SEPSTR](#)).
5. A check is made on all free boundary nodes to determine whether the newly calculated displacement increments puts them inside any surface ([Figure 8-29](#)). If that is the case and the fixed time step procedure is used, the increment is reduced in such a way, that the first node is barely in contact. The process is restarted at [Step 3](#) whenever there is a reduction in increment size.

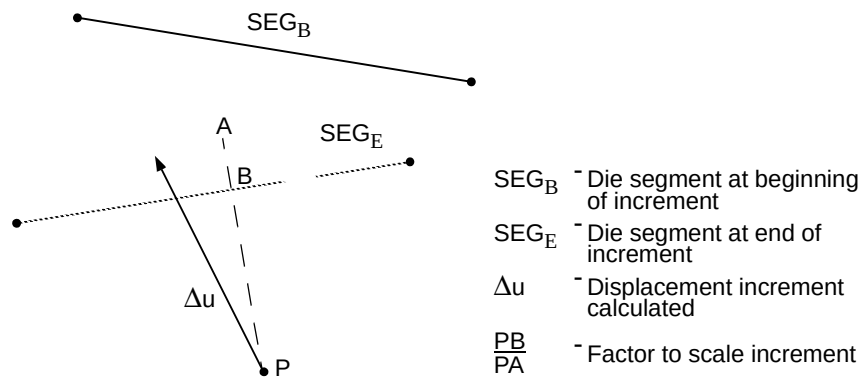


Figure 8-29 New Node in Contact P

- 6. Whenever there is a reduction in increment size and the loading history parameters request fixed increments, the current increment is considered split into two and the remainder is executed next.
- 7. Go to next increment ([Step 1](#)).

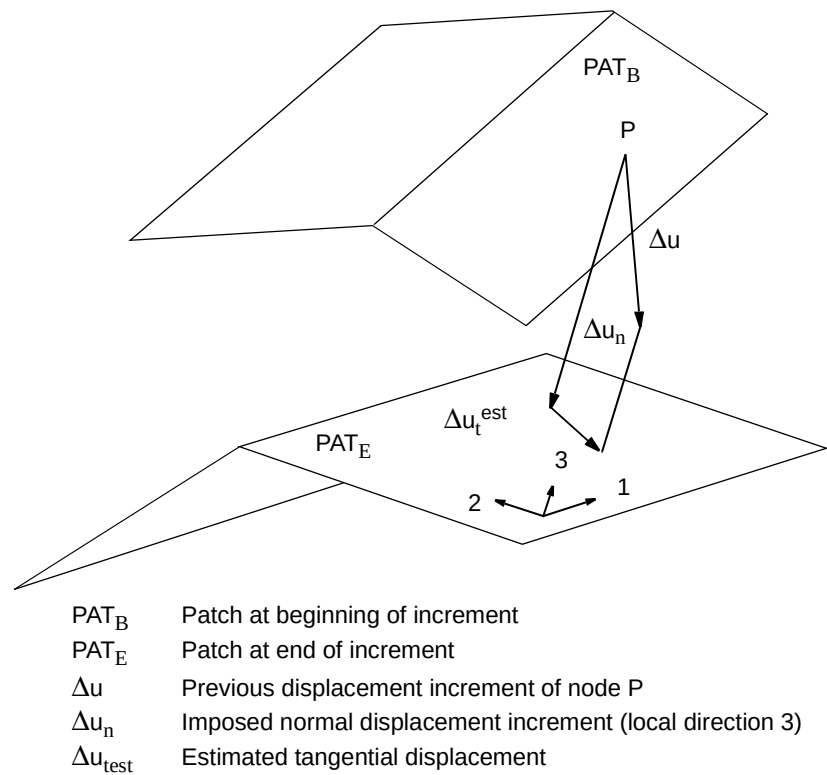


Figure 8-30 Node P Already in Contact (3-D)

Deformable-Deformable Contact

When a node contacts a deformable body, a tying relation is formed between the contacting nodes and the nodes on the other body (Figure 8-31). This constraint relationship uses information regarding the normal to the surface. For lower-order elements, this normal is calculated based upon the piecewise linear representation of the element face. This has the consequences that the constraint relation is not accurate because it is constant over the complete segment. As a node slides from one face to another, there is a discontinuity in the normal which leads to potential numerical difficulties. When higher-order elements are used, a normal is created based upon the true quadratic surface.

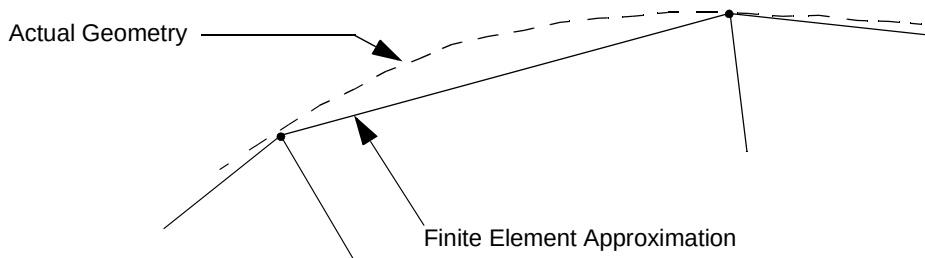


Figure 8-31 Deformable Contact; Piecewise Linear Geometry Description

For lower-order elements, you can fit a smooth surface through the finite elements. For 2-D, a cubic spline is used. This spline is calculated based upon the tangent and position of the nodes on the segment. This gives a more accurate representation of the actual physical geometry and a more accurate calculation of the normal. You must identify where actual discontinuities exist. You can use the [SPLINE](#) option to define the area of the deformable body in 2-D or 3-D where the smoothness is required.

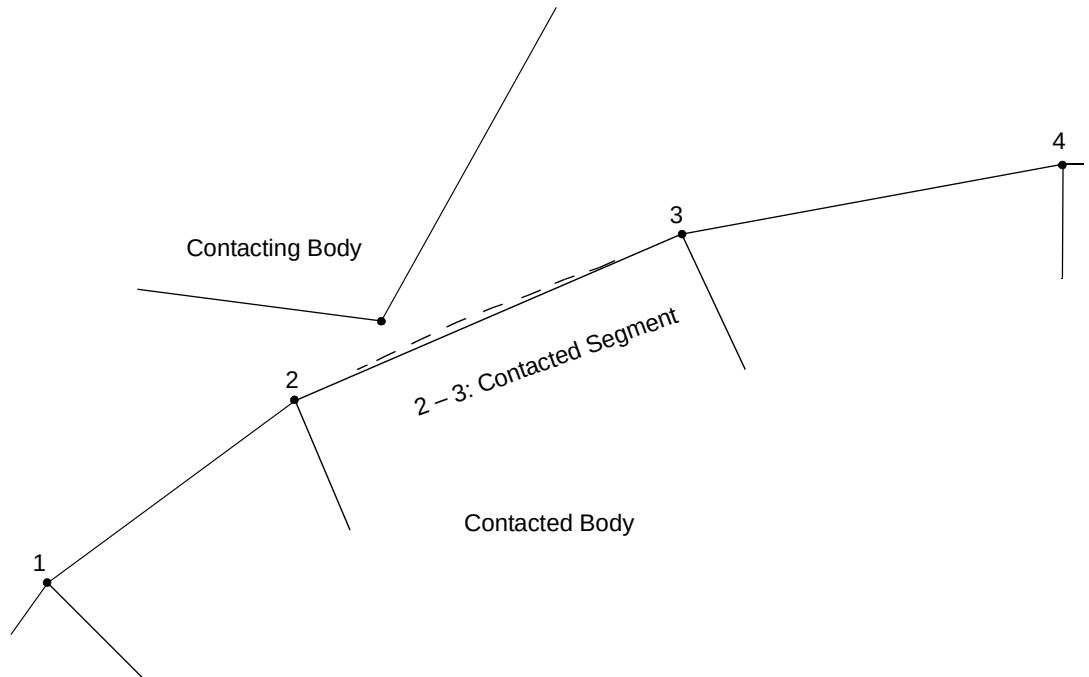


Figure 8-32 Improved Geometry Description of Contacted Deformable Body (2-D case)

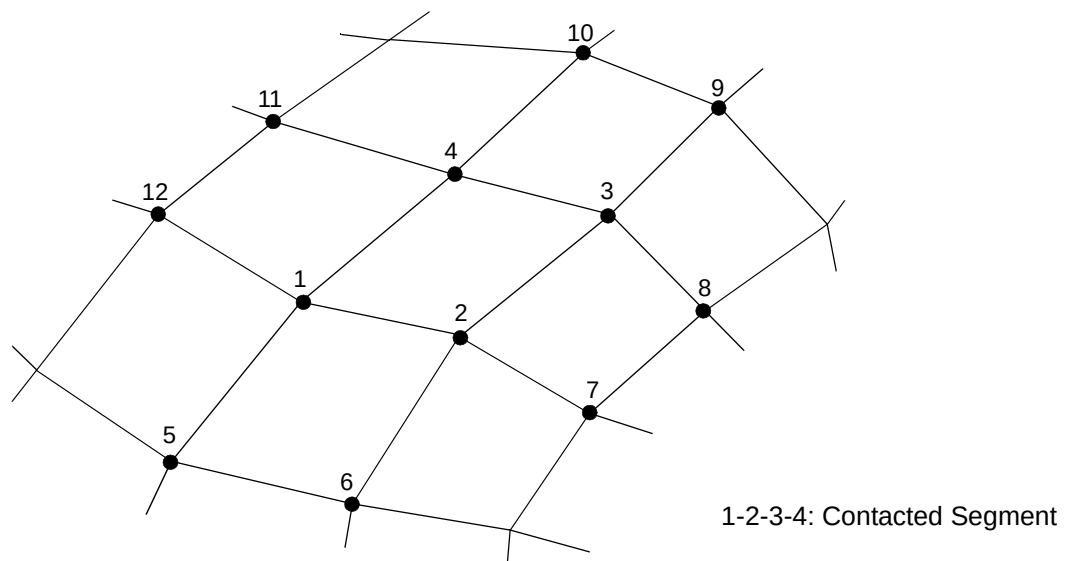


Figure 8-33 Contacted Body

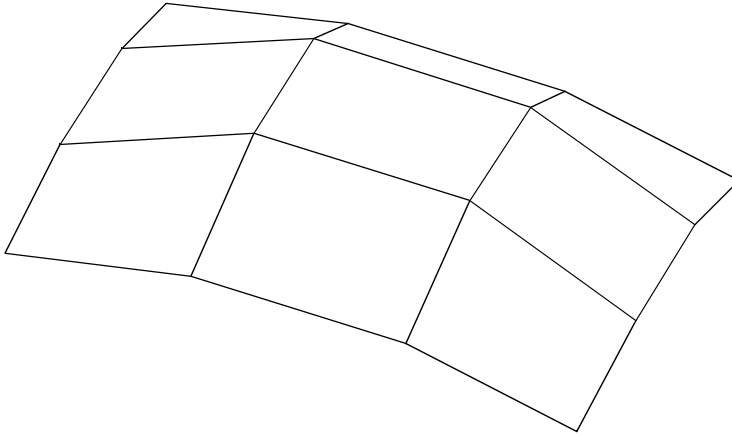


Figure 8-34 Extracted Faces

For 3-D problems, a Coons surface is used to improve the geometric representation in calculating the normal.

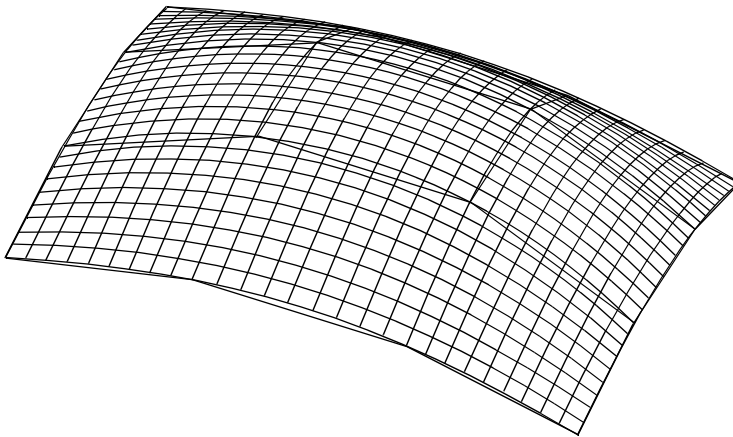


Figure 8-35 Internal Representation Based on Coons Surfaces

Solution Strategy for Deformable Contact

1. Upon initiation, the program determines the boundary nodes on each body. For 2-D, these nodes are ordered in sequence in order to describe the boundary in a counterclockwise manner. Linear geometrical entities are then created out of subsequent pairs of nodes, to define a surface profile as in a rigid surface. If higher-order elements are used in full quadratic mode,

quadratic entities are created from the three nodes. For 3-D using lower-order elements, 4-node patches are created. For 3-D higher-order hexahedral elements, a quadratic surface is created.

2. Once contact between a node and a deformable surface is detected, a tie is activated. The tying matrix is such that the contacting node follows the surface of the surface (Figure 8-36); it can slide along it or be stuck according to the general contact conditions.
3. Contact occurs between all deformable bodies unless the **CONTACT TABLE** option is used. In deformable contact, there is no master or slave body; each body is checked against every other body.
4. During the iteration process of finding an incremental solution, a node can slide from one segment to another, changing the retained nodes of its tie. A recalculation of the bandwidth is, therefore, made every iteration. From increment (or subincrement) to increment, the number of nodes in contact can change. In such cases, an optimization of the bandwidth is automatically available to cope with the possible drastic changes in bandwidth that a new tie produces.

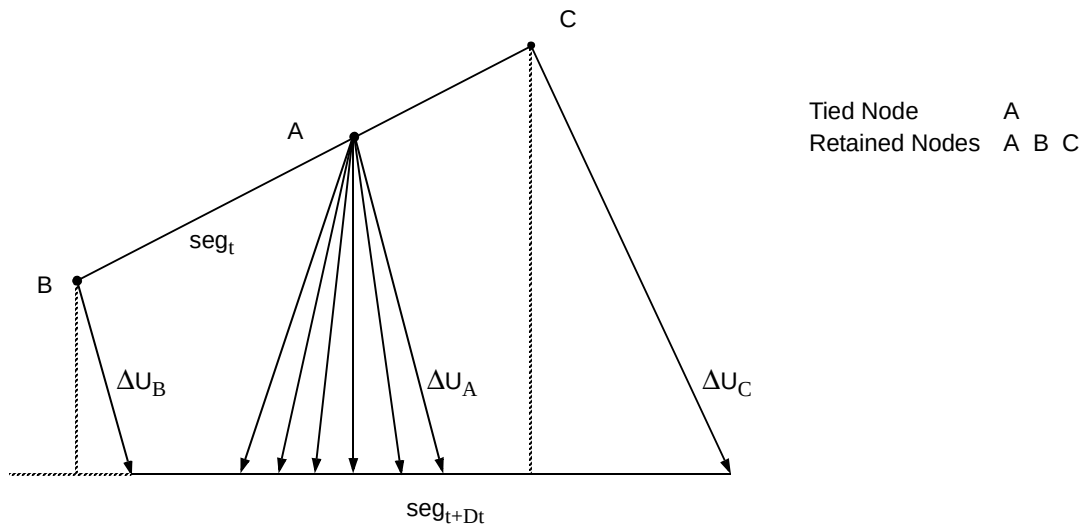


Figure 8-36 Tying in Deformable Contact (2-D) using Lower-order Elements

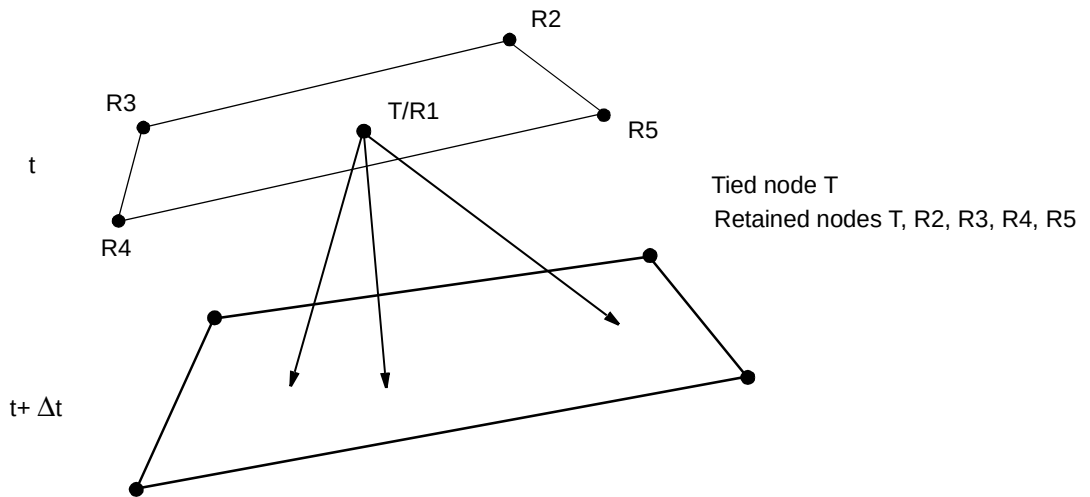


Figure 8-37 Tying in Deformable Contact (3-D) using Lower-order Elements

When the glue option is used between two deformable bodies, a simpler tying relationship is formed, such that no relative motion occurs.

Friction Modeling

The regularized form of the Coulomb friction model can be written as:

$$f_t = \frac{2\mu f_n}{\pi} \arctan\left(\frac{v_r}{RVCNST}\right)$$

is a nonlinear relation between the relative sliding velocity and the friction force. Implementation in MSC.Marc has been done using a nonlinear spring model. Noting that the behavior of a nonlinear spring, as shown in [Figure 8-38](#), is given by the equation:

$$\begin{bmatrix} K & -K \\ -K & K \end{bmatrix} \begin{bmatrix} u_1 \\ u_2 \end{bmatrix} = \begin{bmatrix} F_1 \\ F_2 \end{bmatrix}$$



Figure 8-38 Spring Model

in which K is the spring stiffness and u_1 , u_2 , F_1 , and F_2 are displacements and forces of points 1 and 2, the equivalent in terms of velocities is readily seen to read

$$\begin{bmatrix} K & -K \\ -K & K \end{bmatrix} \begin{bmatrix} v_1 \\ v_2 \end{bmatrix} = \begin{bmatrix} F_{1t} \\ F_{2t} \end{bmatrix}$$

Since K is a nonlinear function of the relative velocity, the above equation is solved incrementally, where within each increment a number of iterations may be necessary. For a typical iteration i , the equation to be solved looks like

$$\begin{bmatrix} K^i & -K^i \\ -K^i & K^i \end{bmatrix} \begin{bmatrix} \delta v_1^i \\ \delta v_2^i \end{bmatrix} = \begin{bmatrix} \Delta F_{1t}^i \\ \Delta F_{2t}^i \end{bmatrix} \quad (8-1)$$

where δv_1^i and δv_2^i are used to update v_1^i and v_2^i by

$$v_1^i = v_1^{i-1} + \delta v_1^i \quad (8-2)$$

$$v_2^i = v_2^{i-1} + \delta v_2^i$$

Notice that v_1^{i-1} and v_2^{i-1} correspond to the beginning of the iteration. For deformable-rigid contact, it is easy to see that

$$\delta v_2^i = 0, \quad (8-3)$$

since the motion of a rigid body (to which node 2 belongs) is exactly prescribed by you. In a static analysis, MSC.Marc provides no direct information about velocities, so they have to be calculated from the displacement and time increments. Denoting a time increment by Δt , we can write

$$\delta v_1^i = \frac{\delta u_1^i}{\Delta t}, \quad (8-4)$$

in which δu_1^i represents the correction of the incremental displacement Δu_1^{i-1} for iteration i like (see also Equation 8-2).

$$\Delta u_1^i = \Delta u_1^{i-1} + \delta u_1 \quad (8-5)$$

Substituting Equation 8-3 and Equation 8-4 into Equation 8-1 yields

$$\frac{1}{\Delta t} K^i \delta u_1^i = \Delta F_1^i \quad (8-6)$$

For the first iteration of an increment, an improvement of equation Equation 8-6 can be achieved by taking into account the velocity v_1^p at the end of the previous increment. Then equation Equation 8-4 can be rewritten as

$$\delta v_1^1 = \frac{\Delta u_1^1}{\Delta t} - v_1^p, \quad (8-7)$$

so that Equation 8-6 can be modified like

$$\frac{1}{\Delta t} K^1 \delta u_1^1 = \Delta F_1^1 - K[v_r^p - v_2^p] \quad (8-8)$$

For the subsequent iterations,

$$\frac{1}{\Delta t} K^i \delta u_1^i = \Delta F_1^i \quad (8-9)$$

In equation Equation 8-8, v_r^i denotes the relative velocity between the points 1 and 2 at the end of the previous increment. It must be noted that the additional term in Equation 8-8 is especially important if the velocity of the rigid body differs much from the relative velocity. This is usually the case in rolling processes, when the roll has been modeled as a rigid body. For this reason, this improved friction model is called friction for rolling.

Iterative Penetration Checking

An alternative to the increment splitting and increment time step reduction procedures when contact occurs is the iterative penetration method. Using this procedure, the iteration process is done simultaneously to satisfy both the contact constraints and global equilibrium using the Newton-Raphson procedure. This procedure is accurate

and stable, but may require more iterations. This procedure is automatically activated when the **AUTO STEP** procedure is used in static analysis or when beam-beam contact is present.

In a conventional iteration process, the finite element system calculates for each iteration:

$$K^T \delta u_i = R_{i-1}$$

where K^T is the tangent stiffness matrix and R_{i-1} are the residuals from the previous iteration.

Using the iterative penetration method K^T is now based upon the contact status at this iteration, both from a constraint and friction perspective. Furthermore, after the solution for δu_i is obtained, the contact procedure is used to determine if new penetration will occur. If, at least, one node penetrates a contact surface, a scale factor is applied to the change in displacements such that the penetrating nodes are moved back to the contact surface. This procedure can be considered a type of line search. Given that s is the fraction of δu_i such that no new penetration occurs, the displacement increment then becomes

$$\Delta U_i = \Delta U_{i-1} + s \delta u_i$$

and the total displacement is

$$U^n = \Delta U^{n-1} + \Delta U_i$$

The strains, stresses, and residuals are based upon these quantities. Separation is based upon these values. When global equilibrium is achieved, based upon the user's criteria, the solution proceeds to the next increment. Because the procedure can reduce the change in displacements, it may require more iterations to complete an increment. It is important to ensure that the maximum allowable number of iterations to complete an increment is set to a sufficiently large value.

Instabilities

In some analyses, because of instabilities such as buckling or a loss of contact, large displacements increments may occur. In such cases, it is possible to specify a maximum δu_i allowed in this case if:

$$s\delta u_i > \delta u^{allowed}$$

for any node if it is scaled down.

Finally, if the solution is still not able to converge and the cutback feature is activated, the time step is reduced in magnitude and the increment is repeated.

References

1. Oden, J. T. and Pires, E. B. "Nonlocal and Nonlinear Friction Laws and Variational Principles for Contact Problems in Elasticity," *J. of Applied Mechanics*, V. 50, 1983.
2. Ju, J. W. and Taylor, R. L. "A perturbed Lagrangian formulation for the finite element solution of nonlinear frictional contact problems," *J. De Mechanique Theorique et Appliquee*, Special issue, Supplement, 7, 1988.
3. Simo, J. C. and Laursen, T. A. "An Augmented Lagrangian treatment of contact problems involving friction," *Computers and Structures*, 42, 1002.
4. Peric, D. J. and Owen, D. R. J. "Computational Model for 3-D contact problems with friction based on the Penalty Method," *Int. J. of Meth. Engg.*, V. 35, 1992.
5. Taylor, R. L., Carpenter, N. J., and Katona, M. G. "Lagrange constraints for transient finite element surface contact," *Int. J. Num. Meth. Engg.*, 32, 1991.
6. Wertheimer, T. B. "Numerical Simulation Metal Sheet Forming Processes," VDI BERICHET, Zurich, Switzerland, 1991.

CHAPTER

9

Boundary Conditions

- Loading
- Kinematic Constraints

MSC.Marc is based on the stiffness method and deals primarily with force-displacement relations. In a linear elastic system, force and displacement are related through the constant stiffness of the system; the governing equation of such a system can be expressed as

$$Ku = F \quad (9-1)$$

where K is the stiffness matrix and u and F are nodal displacement and nodal force vectors, respectively. Equation 9-1 can be solved either for unknown displacements subjected to prescribed forces or for unknown forces (reactions) subjected to prescribed displacements. In general, the system is subjected to mixed (prescribed displacement and force) boundary conditions, and MSC.Marc computes both the unknown displacements and reactions. Obviously, at any nodal point, the nodal forces and nodal displacements cannot be simultaneously prescribed as boundary conditions for the same degree of freedom.

Note: You must prescribe at least a minimum number of boundary conditions to insure that rigid body motion does not occur.

The prescribed force boundary conditions are often referred to as loads and the prescribed displacement boundary conditions as boundary conditions.

Note: Boundary conditions can be prescribed in either the global or a local coordinate system. A nodal transformation between the global and the local coordinate systems must be carried out if the boundary condition is prescribed in a local system.

In a nonlinear stress analysis problem, MSC.Marc carries out the analysis incrementally and expresses the governing equation in an incremental form in terms of the incremental displacement vector Δu and the incremental force vector ΔF .

$$Kdu = df \quad (9-2)$$

Consequently, you must also define both the loads and the prescribed nodal displacements incrementally.

In addition to the prescribed displacement boundary conditions, constraint relations can exist among the nodal displacements. For example, the first degree of freedom of node i is equal to that of node j at all times. The expression of this constraint relation is

$$u_i = u_j \quad (9-3)$$

Generally, a homogenous linear constraint equation can be expressed as

$$u_t = a_1 u_1 + a_2 u_2 + \dots + a_n u_n \quad (9-4)$$

where u represents the degrees of freedom to be constrained, $u_1, \dots, u_n$ are other retained degrees of freedom in the structure, and $a_1, \dots, a_n$ are constants.

You can enter constraints through either the [TYING](#), [SERVO LINK](#), [RBE2](#), or [RBE3](#) options.

You can use linear/nonlinear springs and foundations to provide special support to the structure and the gap and friction element (or [CONTACT](#) option) to simulate the contact problem.

Loading

Different types of analyses require different kinds of loading. For example, the loads in stress analysis are forces; those in heat transfer analysis are heat fluxes. Force is a vector quantity defined by magnitude and direction; heat flux is a scalar quantity defined by magnitude only. Loading can be time invariant (constant value) or time dependent.

Loading Types

You can categorize a particular type of load as either a point (concentrated) load or surface/volumetric (distributed) load, depending on application conditions. The spatial distribution of the load can be uniform or nonuniform. Special loading types also exist in various analyses. For example, centrifugal loading exists in stress analysis, and convection and radiation exist in heat transfer analysis. You can add point loads directly to the nodal force vector, but equivalent nodal forces first must be calculated by MSC.Marc from distributed loads and then added to the nodal force vector. These distinguishing features are described below.

A point (or nodal) load of either a vector (force, moment) or a scalar (heat flux) quantity is a concentrated load that is applied directly to a nodal point (see [Figure 9-1](#)). In a global or local coordinate system, a force must be defined in terms of vector components (see [Figure 9-2](#)). If the force vector is defined in a local coordinate system, then a global-to-local coordinate transformation matrix must be defined for the nodal point (see [Figure 9-3](#) and [Figure 9-4](#)). For axisymmetric elements, the magnitude of the point load must correspond to the ring load integrated around the circumferences.

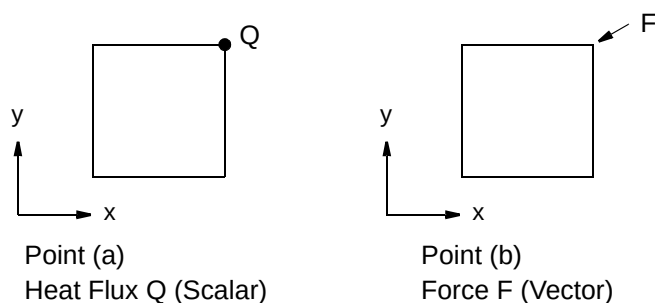


Figure 9-1 Schematic of a Point Load

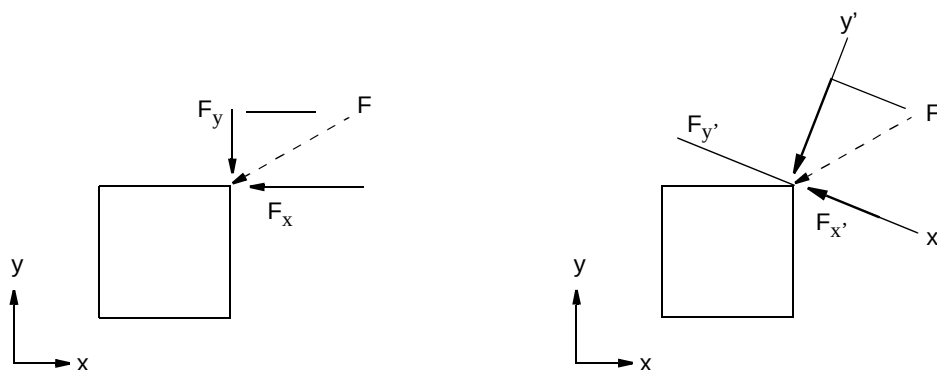


Figure 9-2 Force Components

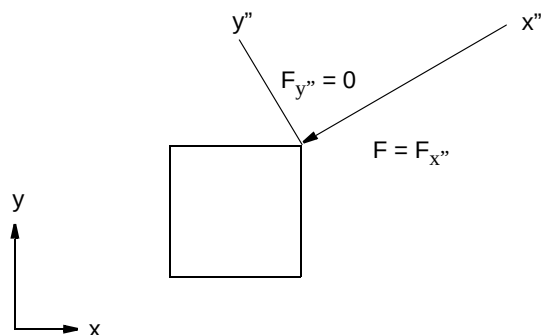
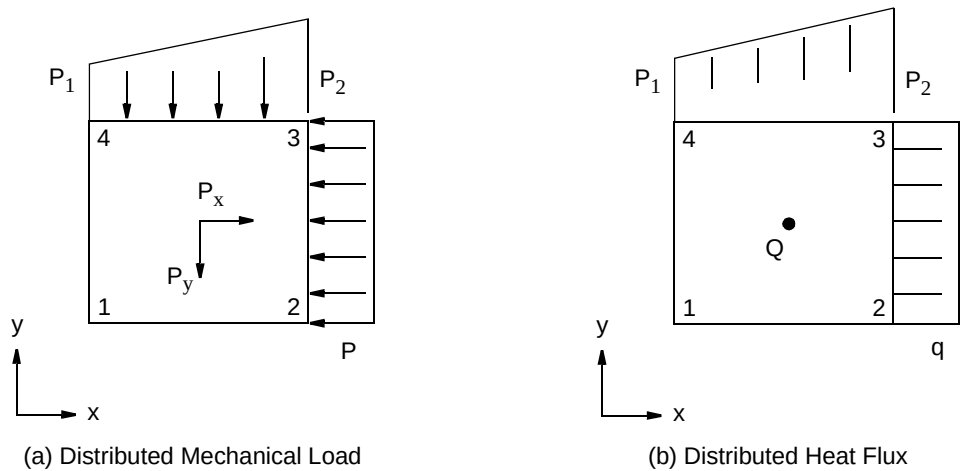


Figure 9-3 Special Selection of Local (x'' , y'') Coordinate System Force Components:
 $F_{y''} = 0$

Surface/volumetric loads, such as pressure, distributed heat flux, and body force, are distributed loads that are applied to the surface (volume) of various elements (see [Figure 9-4](#)). A surface/volumetric load is characterized by the distribution (uniform/nonuniform) and the magnitude of the load, as well as the surface to which the load is applied (surface/volume identification). The total load applied to the surface (volume) is, therefore, dependent on the area (interior) of the surface (volume).

Equivalent nodal forces first must be calculated from surface/volumetric loads and then added to the nodal force vector. MSC.Marc carries out this computation through numerical integration. (See [MSC.Marc Volume B: Element Library](#) for the numbers and locations of these integration points for different elements). The calculated equivalent nodal forces for lower-order elements are the same as those obtained by

equally lumping the uniformly distributed loads onto the nodes. However, for high-order elements, the lumping is no longer simple (see Figure 9-5). As a result, the surface/volumetric loads should not be lumped arbitrarily.



Surface 2-3: Uniform Normal Pressure p	Uniform Heat flux q
Surface 3-4: Nonuniform Normal Pressure $p_1 - p_2$	Nonuniform Heat flux $q_1 = q_2$
Whole Volume: Volumetric Loads P_x, P_y	Volumetric Heat Flux Q

Figure 9-4 Schematic of Surface/Volumetric Load

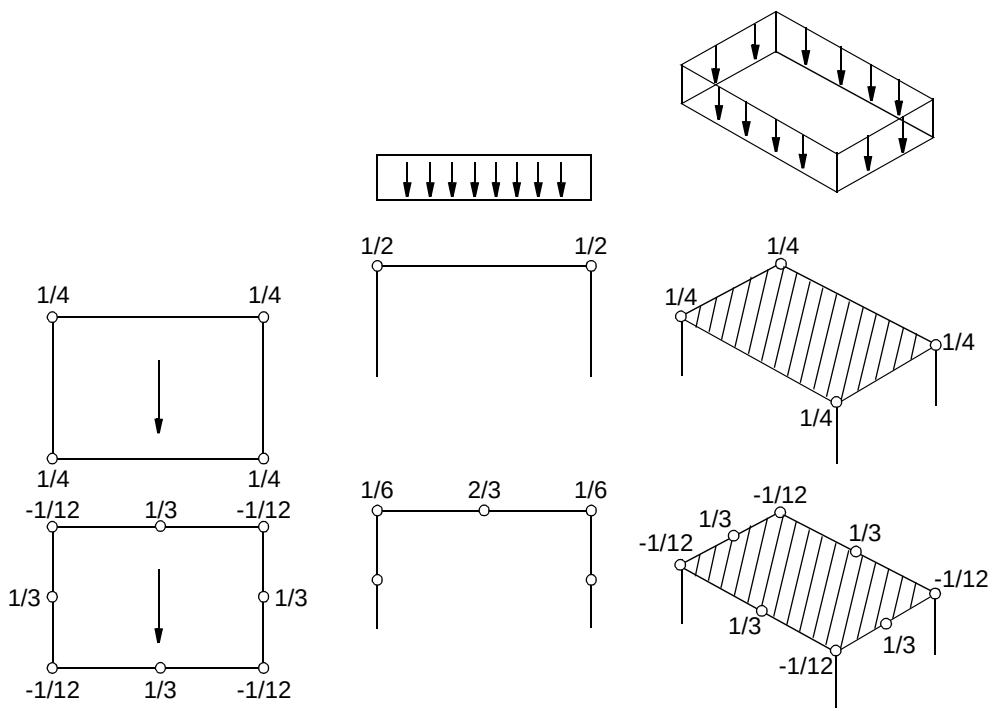


Figure 9-5 Allocation of a Uniform Body Force to Nodes for a Rectangular Element Family

Fluid Drag and Wave Loads

MSC.Marc provides a fluid drag and wave load capabilities that can be applied on beam type structures that are partially or fully submerged in fluid (see [Figure 9-6](#)). Morison's equation is used to evaluate the fluid drag loads that are associated with steady currents. Only distributed drag and buoyancy effects are considered. MSC.Marc employs Airy wave theory to evaluate wave velocities that can be invoked for dynamic analysis option.

Fluid drag and wave loads are invoked using the [DIST LOADS](#) model definition with an IBODY load type of 11. These loads also require the [FLUID DRAG](#) model definition to input the relevant information regarding the fluid elevation and its flow. [Table 9-2](#) lists input options for fluid drag and wave loads.

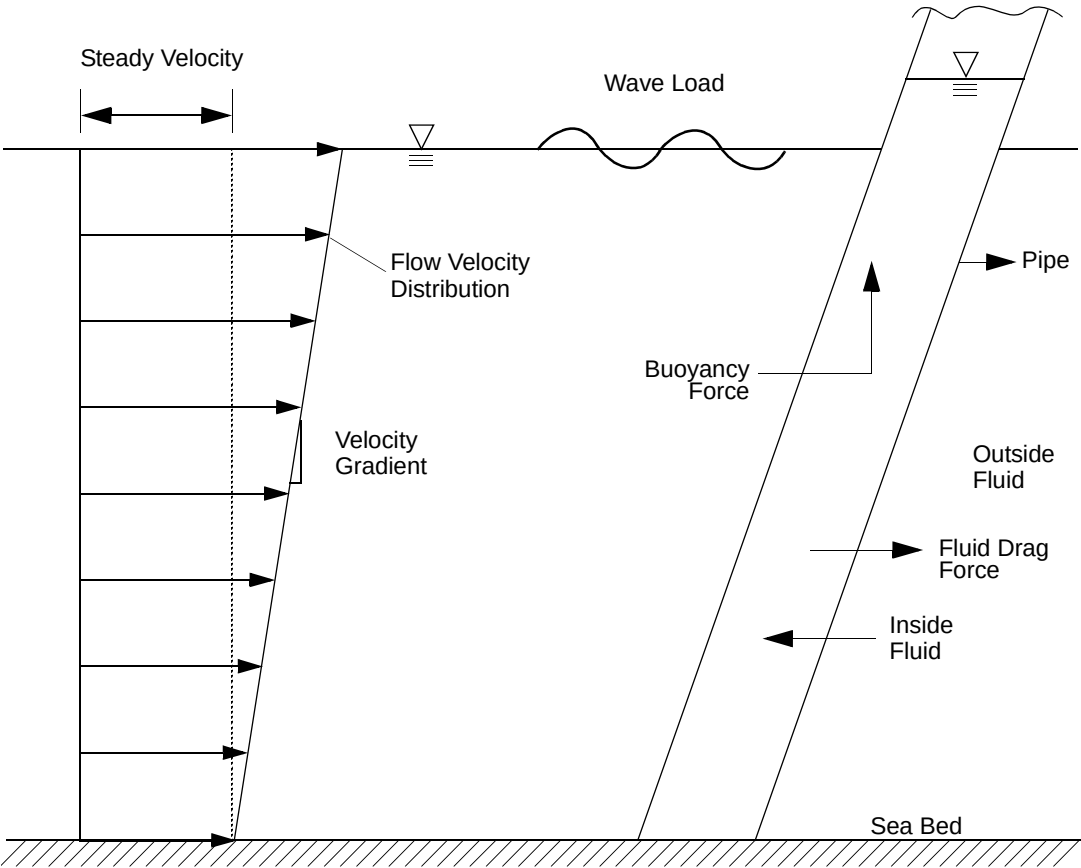


Figure 9-6 Fluid Drag and Wave Loads

Table 9-1 Input Options for Fluid Drag and Wave Loads

Load Description	Input Options		
	Model Definition	History Definition	User Subroutine
Fluid Drag Load	DIST LOADS FLUID DRAG	DIST LOADS	
Wave Load	DIST LOADS FLUID DRAG	DIST LOADS	

Mechanical Loads

MSC.Marc allows you to enter mechanical loads in various forms for stress analysis. These loads can be concentrated forces and moments, uniformly and nonuniformly distributed pressures, body forces, gravity or centrifugal loads. Table 9-2 lists input options for mechanical loads.

Table 9-2 Input Options for Mechanical Loads

Load Description	Input Options		
	Model Definition	History Definition	User Subroutine
Point Load: Concentrated Force/Moment	POINT LOAD	POINT LOAD	FORCDT
Surface Load Pressure Shearing Forces, and Distributed Moment (Uniform/ Nonuniform Distribution)	DIST LOADS	DIST LOADS	FORCEM
Volumetric Load Body Forces and Acceleration Forces	DIST LOADS	DIST LOADS	FORCEM
Centrifugal Loading	DIST LOADS ROTATION A	DIST LOADS	
Coriolis Loading	DIST LOADS ROTATION A	DIST LOADS	
Fluid Loading	DIST LOADS FLUID DRAG		

Application of centrifugal and Coriolis loadings require the **ROTATION A** model definition option, which defines the data corresponding to the axis of rotation. The actual load can be invoked by specifying an IBODY load types 100 or 103 for centrifugal and Coriolis loadings, respectively. The square of rotation speed, ω^2 , is entered in radians per time, for the magnitude of the distributed load. The mass density must also be defined in the **ISOTROPIC** or **ORTHOTROPIC** options.

Application of gravity load is achieved by using IBODY load type 102. The mass density must be defined in the **ISOTROPIC** or **ORTHOTROPIC** options. The acceleration can be given independently in the x, y, and z direction through the **DIST LOADS** option.

Use of the **FOLLOW FOR** parameter results in the equivalent nodal loads due to pressures being calculated based on the current geometry. When this option is used, any change of surface area or orientation results in a change of load. **FOLLOW FOR** automatically invokes the **LOAD COR** parameter, so stresses should be stored at all integration points. The **FOLLOW FOR** parameter is typically used when a shell structure, which can undergo large deformation and rotations, is subjected to a pressure load. You can also specify that the follower force stiffness matrix can be included.

A number of history definition options are available for input of multiple load increments. For example, the **AUTO LOAD** option generates a specified number of increments, all having the same load increment, and is useful for nonlinear analysis with proportional loads; the **PROPORTIONAL INCREMENT** option allows the previous load increment to be scaled up or down for use in the current load increment. The **AUTO INCREMENT** option allows automatic load stepping in a quasi-static analysis and is useful for both geometrically and materially nonlinear problems.

Cavity Pressure Loading

MSC.Marc allows the modeling of structures enclosing cavities by updating the cavity internal pressure as the cavity volume change. For ideal gas-filled cavities, the equation of state relating the cavity pressure and volume can be written as:

$$pV = nR_oT \quad (9-5)$$

where p is the cavity total pressure, V is the cavity volume, n is the number of molecules of the gas, R_o is the Universal Gas Constant

($R_o = 8.31447 \text{ J/(mol}^\circ\text{K)} = 1545.35 \text{ ft.lbf/(mol}^\circ\text{R)}$), and T is the absolute gas temperature. The gas mass, M , is related to the number of molecules of the gas by:

$$\mu = M/n \quad (9-6)$$

where μ is the molar mass of the gas. Substituting by n from [Equation 9-6](#) into [Equation 9-5](#) gives:

$$pV = MRT \quad (9-7)$$

where $R = R_0/\mu$ is the Specific Gas Constant. With the gas density, ρ , defined as:

$$\rho = M/V \quad (9-8)$$

the equation of state of an ideal gas can be finally written as:

$$p = \rho RT \quad (9-9)$$

The Specific Gas Constant is calculated from:

$$R = p_r/\rho_r T_r \quad (9-10)$$

where p_r is the gas reference pressure, T_r is the gas reference temperature, and ρ_r is the gas reference density. The user must ensure that the values entered for the gas reference pressure, temperature and density are consistent. The cavity total pressure is given by:

$$p = p_a + p_g \quad (9-11)$$

where p_a is the ambient pressure and p_g is the cavity gage pressure. Only the cavity gage pressure is applied to the structure forming the cavity.

The following loading scenarios are available for cavities:

Closed Cavity

If the cavity is closed, the mass of the gas is preserved

$$M = M_o \quad (9-12)$$

where M_o is the cavity mass from the previous increment. The gas is assumed to undergo a general polytropic process represented by:

$$pV^\gamma = \text{constant} \quad (9-13)$$

where γ is the polytropic exponent. The gas pressure can thus be updated using

$$\frac{p}{p_o} = \left(\frac{V_o}{V}\right)^\gamma \quad (9-14)$$

where p_o and V_o are the cavity pressure and volume from the previous increment, respectively. Using [Equation 9-7](#) and [Equation 9-14](#), the gas temperature can be updated using

$$\frac{T}{T_o} = \left(\frac{V_o}{V} \right)^{\gamma-1} \quad (9-15)$$

where T_o is the cavity temperature from the previous increment. The gas density is calculated from:

$$\rho = M/V \quad (9-16)$$

Closed cavity processes can be set to occur:

- at constant pressure, isobaric, using $\gamma = 0$.
- at constant temperature, isothermal, using $\gamma = 1$.
- with no heat transfer to the surroundings, adiabatic, using $\gamma = k$, where k is the adiabatic exponent. For ideal gases, k is a constant that depends only on the number of atoms in the gas molecule (monoatomic gases: $k = 1.67$, diatomic gases: $k = 1.4$, triatomic gases: $k = 1.33$).

Applied Pressure

In this case the new cavity pressure is updated using

$$p = p_o + \Delta p \quad (9-17)$$

The gas temperature is assumed to be constant

$$T = T_o \quad (9-18)$$

The gas density is updated using

$$\rho = p/RT \quad (9-19)$$

and the gas mass is recalculated as

$$M = \rho V \quad (9-20)$$

Applied Mass

In this case the new cavity mass is updated using

$$M = M_o + \Delta M \quad (9-21)$$

If $\Delta M \neq 0$, gas is pumped in or out of the cavity. The gas density is first updated as

$$\rho = M/V \quad (9-22)$$

the gas temperature is assumed to be constant

$$T = T_o \quad (9-23)$$

and the cavity pressure is calculated as

$$p = \rho RT \quad (9-24)$$

If $\Delta M = 0$, the cavity is assumed to be closed and Equations 9-12 to 9-16 apply with $\gamma = 1$.

User Defined Loading

The user routine, [UCAV](#), allows the user to define and control the cavity pressure for loading scenarios other than the ones specified above.

Cavity Modeling

The [CAVITY](#) parameter is used to enter the number of cavities in the model (maximum 1000), an upper bound to the number of segments in each cavity and an upper bound to the number of nodes per segment of cavity boundary.

The [CAVITY](#) model definition option is used to enter the ambient pressure, the polytropic process exponent, and the reference pressure, temperature and density for each cavity.

The applied pressure or mass is entered through the DIST LOADS [model definition](#) and [history definition](#) options. The distributed load type, entered on the DIST LOADS option of the elements forming the cavity, is modified for cavity loading according to the following relation

$$ibody\_cavity = icavity * 10000 + icavity\_type * 1000 + ibody \quad (9-25)$$

where

ibody_cavity is the cavity-modified value for the distributed load type.

icavity is the cavity id.

icavity_type is the cavity load type:

0: cavity is closed.

1: cavity is loaded with an applied pressure.

2: cavity is loaded with an applied mass.

9: cavity load is defined by user subroutine [UCAV](#).

ibody is the original value for the distributed load type (see library element description in [MSC.Marc Volume B: Element Library](#).)

In the first increment of the analysis, increment zero, the cavity temperature is calculated by averaging the temperatures of the elements forming the cavity. If there are no nodal temperatures defined, the cavity temperature is taken equal to the cavity reference temperature. For AXITO3D analysis, the cavity pressure, mass, and temperature for increment zero are read from the post file of the corresponding axisymmetric problem. If the cavity is closed in increment zero, the cavity is assumed to be initially at ambient conditions. In this case, the initial gage pressure is equal to zero and the initial mass is based on the ambient pressure and initial volume. Moreover, if the ambient pressure is equal to zero, the cavity is assumed to be initially unloaded. Tables 9-3 and 9-4 summarize the cavity functioning for the different cavity load types and incremental load values during the initial and subsequent load increments, respectively.

Table 9-3 Cavity Functioning for Increment Zero

Cavity Load Type	Incremental Load Value	Function
0	Ignored	Cavity is initially at ambient conditions
1	$\Delta p = 0.0$	Cavity is initially empty
1	$\Delta p \neq 0.0$	Cavity is initially at the applied pressure
2	$\Delta M = 0.0$	Cavity is initially empty
2	$\Delta M \neq 0.0$	Cavity initially contains the applied mass
9	Passed to UCAV	Call UCAV

Table 9-4 Cavity Functioning for Subsequent Increments

Cavity Load Type	Incremental Load Value	Function
0	Ignored	Cavity is closed, constant mass, polytropic process
1	$\Delta p = 0.0$	Cavity is open, constant pressure and temperature
1	$\Delta p \neq 0.0$	Cavity is open, pressure is applied, constant temperature
2	$\Delta M = 0.0$	Cavity is closed, constant mass, isothermal process
2	$\Delta M \neq 0.0$	Cavity is open, mass is applied, constant temperature
9	Passed to UCAV	Call UCAV

In general, standard structural elements are used to define the boundaries of cavities and no extra elements are required. However, to model the boundaries of cavities in regions where standard finite elements are not present (for example, along rigid boundaries) cavity surface elements (elements 171—174) can be used. These elements can also be glued to moving rigid surfaces. They are for volume calculation purposes only and do not contribute to the stiffness equations of the model.

For 2-D problems, the volume of the cavity is calculated as the sum of the areas of all the triangles formed by the cavity segments as bases and the coordinate system origin as the apex multiplied by the cavity thickness. Elements forming the cavity are assumed to be of equal thickness. For 3-D problems, the volume of the cavity is calculated as the sum of the volumes of all the tetrahedrons formed by the cavity patches as bases and the coordinate system origin as the apex. For axisymmetric problems, the volume of the cavity is calculated as the sum of the volumes of all the cone frustums formed by the cavity segment as the cone slant height and two parallel base circles and with an axis along the axis of symmetry of the problem. In the axisymmetric case, it is not necessary to use cavity surface elements along lines that are perpendicular to the axis of symmetry to close cavities.

In coupled thermo-mechanical analysis, heat transfer between the gas inside the cavity and the surrounding structure is not supported.

Cavities are not allowed to split or to join due to deformation or self-contact. If self-contact occurs within a cavity, the resulting cavities are still treated as a single cavity.

If the cavity is formed out of membrane and/or shell elements, the user must ensure that all elements defining the cavity are aligned; that is, the cavity surface is defined by either all top or all bottom element faces.

Thermal Loads

Element integration point temperatures are used in a thermal stress analysis to generate thermal load. The **AUTO THERM** history definition option allows automatic application of temperature increments based on a set of temperatures defined throughout the mesh as a function of time. The **CHANGE STATE** option presents the temperatures to MSC.Marc, and MSC.Marc then creates its own set of temperature steps based on a temperature change tolerance provided through this option. [Table 9-5](#) lists input options for thermal loads. You can input either the incremental temperature or the total temperatures as thermal loads.

The thermal strain increment is defined as

$$\Delta \epsilon^{th} = \int \alpha \Delta T \tag{9-26}$$

where $\Delta \epsilon^{th}$ is thermal strain increment, α is the coefficient of thermal expansion, and ΔT is the temperature increment. Equivalent nodal forces are calculated from the thermal strain increment and then added to the nodal force vector for the solution of the problem.

You can input the coefficient of thermal expansion through the [ISOTROPIC](#) or [ORTHOTROPIC](#) model definition options and the temperature increment through various options.

Table 9-5 Input Options for Thermal Loads

Load Description	Input Options			
	Parameter	Model Definition	History Definition	User Subroutine
Incremental Temperatures*	THERMAL	THERMAL LOADS	THERMAL LOADS	CREDE
Total Temperatures*	THERMAL	CHANGE STATE	CHANGE STATE	NEWSV
Initial Temperature*		INITIAL STATE		INITSV
Total Nodal Temperatures		POINT TEMP	POINT TEMP	
Initial Nodal Temperature		INITIAL TEMP		USINC
*Temperatures must be specified at each integration point (or at the centroid if the CENTROID parameter is used) of each element in the mesh.				

Initial Stress and Initial Plastic Strain

MSC.Marc allows you to enter a set of initial stresses that simulate the stress state in the structure at the beginning of an analysis. A typical example is prestress in a tensioned fabric roof. The set of initial stresses must be self-equilibrating and should not exceed the yield stress of the material. [Table 9-6](#) shows the input options for initial stress.

Table 9-6 Input Options for Initial Stress and Initial Plastic Strain

Load Description	Input Options		
	Parameter	Model Definition	User Subroutine
Initial Stress Prestress	ISTRESS	INIT STRESS	UINSTR
Initial Plastic Strain		INITIAL PLASTIC STRAIN	INITPL

MSC.Marc also provides various ways of initializing the equivalent plastic strain throughout the model. This is useful in metal forming analysis in which the previous amount of equivalent plastic strain is often required. This history dependent variable represents the amount of plastic deformation that the model was subjected to, and is used in the work (strain) hardening model. The input option is shown in [Table 9-6](#).

Heat Fluxes

In a heat transfer analysis, you can enter heat fluxes in various forms. Unlike stress analysis, heat transfer analysis requires entering the total values of flux. [Table 9-7](#) lists the input options for heat fluxes.

Table 9-7 Input Options for Heat Fluxes

Load Description	Input Options		
	Model Definition	History Definition	User Subroutine
Point Heat Flux (Sink or Source)	POINT FLUX	POINT FLUX	FORCDT
Surface Heat Flux, Convection, Radiation	DIST FLUXES FILMS	DIST FLUXES FILMS	FLUX* FILM*
Volumetric Flux Load Body Flux	DIST FLUXES	DIST FLUXES FILMS	FLUX*
* Can be used for complicated flux loadings, convection, and radiation, allowing the input of nonuniform temperature- and time-dependent boundary conditions.			

There are three special heat flux conditions representing insulation, convection, and radiation.

1. Insulation

$$q = 0 \quad (9-27)$$

No input is required for the insulated case.

2. Convection

$$q = H(T_s - T) \quad (9-28)$$

You must enter the film coefficient H and ambient temperature T through the **FILMS** option or the **FILM** user subroutine. You can also directly input the heat flux q using the **FLUX** user subroutine.

3. Radiation

$$q = \sigma \cdot \varepsilon (T_s^4 - T_\infty^4) \quad (9-29)$$

You must enter either the heat flux q using the **FLUX** user subroutine or the temperature dependent film coefficient $H(T_s, T, \sigma, \varepsilon)$ and ambient temperature T using the **FILM** user subroutine. These relationships are shown in **Equation 9-30**. The use of **FILM** is recommended to ensure a stable solution.

$$\begin{aligned} q &= \sigma \varepsilon (T_s^4 - T_\infty^4) \\ &= \sigma \cdot \varepsilon (T_s^3 + T_s^2 T_\infty + T_s T_\infty^2 + T_\infty^3) (T_s - T_\infty) \\ &= H(T_s, T_\infty) (T_s - T_\infty) \end{aligned} \quad (9-30)$$

where σ is the Stefan-Boltzmann constant, ε is emissivity, T_s and T_∞ are unknown surface and ambient temperatures, respectively.

An alternative, the **VIEW FACTOR** option can be used to read in viewfactors calculated by MSC.Marc Mentat. The **RADIATION** parameter is also required.

Mass Fluxes and Restrictors

In a hydrodynamic bearing analysis, you can enter mass fluxes, restrictors, and pump pressures as loads. **Table 9-8** lists input options for these quantities.

Table 9-8 Input Options for Mass Fluxes and Restrictors

Load Description	Input Options			
	Parameter	Model Definition	History Definition	User Subroutine
Nodal Mass Fluxes	BEARING	POINT FLUX	POINT FLUX	FORCDT
Distributed Mass Fluxes	BEARING	DIST FLUXES	DIST FLUXES	FLUX*
Restrictors	BEARING RESTRICTOR	RESTRICTOR		URESTR**
* Can be used for nonuniform mass fluxes ** Can be used for nonuniform restrictions or pump pressures.				

Electrical Currents

In coupled thermo-electrical (Joule heating) analysis and coupled electrical-thermal-mechanical analysis (Joule-mechanical), you can prescribe electrical currents as loads for the calculation of unknown nodal voltages. In magnetostatic analysis, you can also define the current. In such cases, as a steady state analysis is performed, there is no time variation of the currents. [Table 9-9](#) lists input options for electrical currents.

Table 9-9 Input Options for Electrical Currents

Load Description	Input Options		
	Model Definition	History Definition	User Subroutine
Nodal Current	POINT CURRENT	POINT CURRENT	FORCDT
Surface and Body Currents	DIST CURRENT	DIST CURRENT	FLUX

Electrostatic Charges

In an electrostatic analysis, the charge can be entered, noting that a steady state analysis is performed so there is no time variation of charge. [Table 9-10](#) summarized the input options.

Table 9-10 Input Options for Electrostatic Charge

Load Description	Input Options		
	Model Definition	History Definition	User Subroutine
Nodal Charge	POINT CHARGE		FORCDT
Distributed and Body Charges	DIST CHARGES		FLUX

Acoustic Sources

In acoustic analysis, you can enter a source pressure if a transient analysis by modal superposition is being performed. [Table 9-11](#) summarized the input options.

Table 9-11 Input Options for Acoustic Sources

Load Description	Input Options		
	Model Definition	History Definition	User Subroutine
Nodal Source	POINT SOURCE	POINT SOURCE	FORCDT
Distributed Source	DIST SOURCES	DIST SOURCES	FLUX
Nodal Source	FIXED PRESSURE	PRESS CHANGE	FORCDT

Piezoelectric Loads

In a piezoelectric analysis, both mechanical loads and electrostatic charges can be entered. These values can have time variation if a transient analysis is performed or a harmonic excitation can be applied. [Table 9-2](#) gives a summary of the mechanical input options, and [Table 9-10](#) gives a summary of the electrostatic input options.

Magnetostatic Currents

In a magnetostatic analysis, the current can be entered, noting that a steady state analysis is performed so there is no time variation of current. [Table 9-12](#) summarizes the input options.

Table 9-12 Input Options for Magnetostatic Current

Load Description	Input Options		
	Model Definition	History Definition	User Subroutine
Nodal Current	POINT CURRENT	POINT CURRENT	FORCDT
Distributed and Body Current	DIST CURRENT	DIST CURRENT	FLUX

Electromagnetic Currents and Charges

In an electromagnetic analysis, the current can be entered. These values can have time variation if a transient analysis is performed or a harmonic excitation can be applied. Table 9-13 summarizes the input options.

Table 9-13 Input Options for Currents and Charges

Load Description	Input Options		
	Model Definition	History Definition	User Subroutine
Nodal Current	POINT CURRENT	POINT CURRENT	FORCDT
Distributed and Body Current	DIST CURRENT	DIST CURRENT	FLUX
Nodal Charge	POINT CURRENT POINT CHARGE	POINT CURRENT	FORCDT
Distributed and Body Charge	DIST CHARGE	DIST CHARGE	

Kinematic Constraints

MSC.Marc allows you to input kinematic constraints through various options that include

- [Boundary Conditions](#) (prescribed nodal values)
- [Transformation of Degree of Freedom](#)
- [Shell Transformation](#)
- [Tying Constraint](#)
- [Rigid Link Constraint](#)
- [Shell-to-Solid Tying](#)
- [Insert](#)
- [Support Conditions](#)
- [Cyclic Symmetry](#)
- [MSC.Nastran RBE2 and RBE3](#)

Boundary Conditions

MSC.Marc allows you to specify the nodal value for a particular degree of freedom. If you do not give a nodal value when you specify the boundary condition, MSC.Marc sets the fixed nodal value to zero. An option allows boundary conditions to be specified at the time of two-dimensional mesh generation with MESH2D. You can apply a different set of boundary conditions for each load increment.

[Table 9-14](#) gives the nodal values for the various analyses, and [Table 9-15](#) lists the input options for boundary conditions in different analyses.

Table 9-14 Analyses with Corresponding Nodal Values

Analysis	Nodal Values
Acoustics	Pressure
Coupled Fluid Thermal	Velocity, Pressure, and Temperature
Coupled Thermo-electrical	Voltage and Temperature
Coupled Thermo-mechanical	Displacement and Temperature
Coupled Electrical-thermo-mechanical	Voltage, Temperature, and Displacement
Electrostatic	Potential
Piezoelectric	Displacement and Potential
Fluid	Velocity and Pressure
Heat transfer	Temperature

Table 9-14 Analyses with Corresponding Nodal Values (Continued)

Analysis	Nodal Values
Hydrodynamic Bearing	Pressure
Magnetostatic	Potential
Rigid Plastic Flow	Velocity
Stress	Displacements

Table 9-15 Input Options for Boundary Conditions

Load Description	Input Options		
	Model Definition	History Definition	User Subroutine
Displacement	FIXED DISP	DISP CHANGE	FORCDT
Temperature	FIXED TEMPERATURE	TEMP CHANGE	FORCDT
Voltage	VOLTAGE	VOLTAGE CHANGE	FORCDT
Pressure	FIXED PRESSURE	PRESS CHANGE	FORCDT
Potential	FIXED POTENTIAL		FORCDT
Velocity	FIXED VELOCITY	VELOCITY CHANGE	FORCDT

Transformation of Degree of Freedom

MSC.Marc allows transformation of individual nodal degrees of freedom from the global direction to a local direction through an orthogonal transformation that facilitates the application of boundary conditions and the tying together of shell and solid elements.

Transformations are assumed to be orthogonal. Once you invoke a transformation on a node, you must input all loads and kinematic conditions for the node in the transformed system. Nodal output is in the transformed system. This option is invoked using the **TRANSFORMATION** option.

The **UTRANFORM** option allows transformations to be entered via the **UTRANS** user subroutine. This allows you to transform the degrees of freedom at an individual node from global directions to a local direction through an orthogonal transformation. **UTRANFORM** allows you to change the transformation with each increment. When you invoke this option, the nodal output is in both the local and the global system.

Shell Transformation

The **SHELL TRANSFORMATION** option allows you to transform the global degree of freedom of doubly curved shells to shell degrees of freedom in order to facilitate application of forces in the shell directions, edge moments, and clamped or simply supported boundary conditions. There are four types of shell transformations.

The **SHELL TRANSFORMATION** model definition option specifies information on the shell transformation. For Types **1** and **3**, only the node number has to be specified. For Types **2** and **4**, a boundary direction (the direction cosine of t as shown in [Figure 9-7](#)) also has to be specified in the θ_1, θ_2 surface.

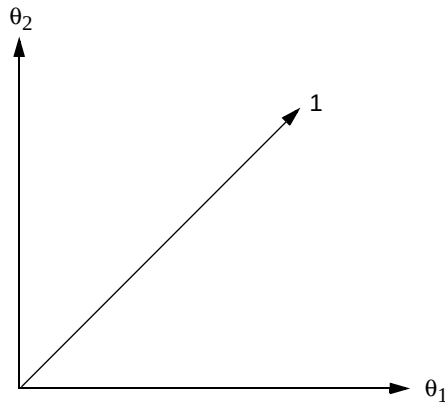


Figure 9-7 Boundary Directions in Shell Transformation

After transformation, the following definitions apply.

Type 1: Transformation for two-dimensional beams and shell nodes (Element Types **15**, **16**, and **17**). The transformation defines the degrees of freedom with respect to a local coordinate system (s, n) (see [Figure 9-15](#)). The four degrees of freedom after transformation are:

1 = u_s tangential displacement

2 = u_n normal displacement

3 = ϕ rotation

4 = ε Meridional stretch

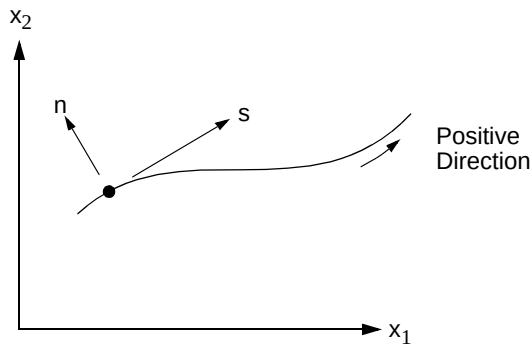


Figure 9-8 Type 1: Shell Transformation

Type 2: Transformations for doubly curved shell nodes with nine degrees of freedom (Element Type 4 corner nodes of Element Type 24). The transformation defines a local coordinate system (t, s, n). (See [Figure 9-9](#)).

The nine degrees of freedom after transformation are:

- 1 = u_t displacement in specified (boundary) direction
- 2 = u_s displacement normal to (boundary) direction but tangential to shell surface
- 3 = u_n displacement normal to shell surface
- 4 = ϕ_t rotation of shell around boundary
- 5 = ϕ_s rotation of shell around normal to boundary tangential to the shell surface
- 6 = ϕ_n rotation of boundary around normal to shell surface
- 7 = ϵ_t stretch tangential to specified (boundary) direction
- 8 = ϵ_s stretch normal to specified (boundary) direction
- 9 = γ_{ts} shear stretch in t-s direction

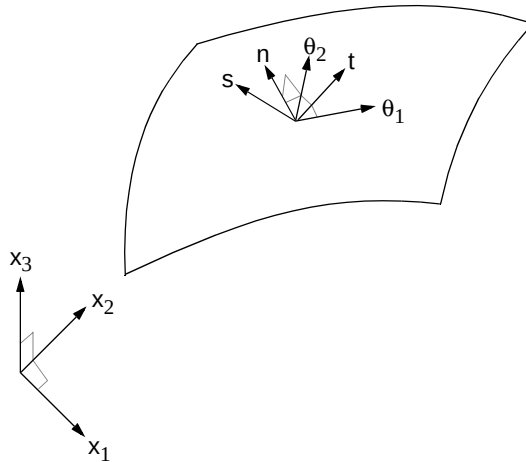


Figure 9-9 Types 2 and 4: Shell Transformations

Type 3: Transformations at midside nodes for doubly curved shell nodes with three degrees of freedom (Element Type 24). The transformation defines a local coordinate system (t, s, n), (see Figure 9-10). The degrees of freedom after transformation are:

1 = $\partial u_t / \partial s$ rotation of normal to boundary around normal to shell

2 = $\partial u_s / \partial s$ stretch normal to boundary

3 = $\partial u_n / \partial s$ rotation around boundary

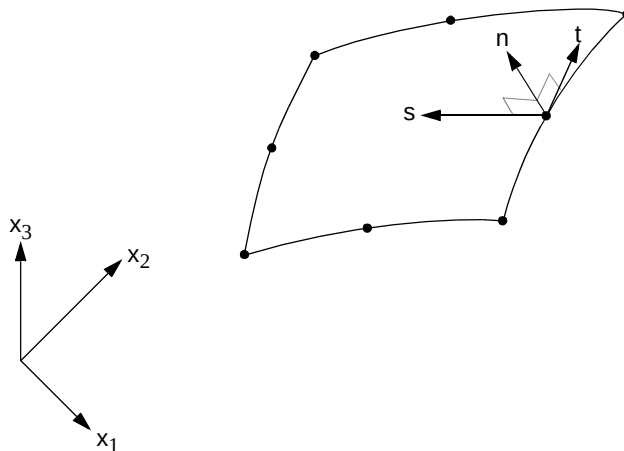


Figure 9-10 Type 3: Shell Transformation

Type 4: Transformation for doubly curved nodes with 12 degrees of freedom (Element Type 4). The transformation defines a local coordinate system (t, s, n) . The first 9 degrees of freedom after transformation are the same as 1 through 9 in Type 2 (Figure 9-9), and the remaining three are:

$$10 = \partial^2 u_t / \partial \Theta_1 \partial \Theta_2 \quad \text{variation of } g \text{ along the boundary} \quad (9-31)$$

$$11 = \partial^2 u_s / \partial \Theta_1 \partial \Theta_2 \quad \text{variation of } e \text{ along the boundary} \quad (9-32)$$

$$12 = \partial^2 u_n / \partial \Theta_1 \partial \Theta_2 \quad \text{variation of } f \text{ along the boundary} \quad (9-33)$$

When using the **SHELL TRANSFORMATION** option, the displacement increments and reaction forces are output in the local directions immediately after solution of the equations. At the end of the increment, you can print out the global displacement increments and total displacements in the global coordinate directions.

If you invoke the **FOLLOW FOR** parameter, the **SHELL TRANSFORMATION** option defines a local coordinate system in the current (updated) geometry of the structure. This additional feature is especially useful with the **UPDATE** parameter, because you can then specify edge moments and/or large edge rotations of shells and beams.

CAUTION: If you apply a shell transformation to a node, do not apply a standard transformation or shell tying type to that node.

Tying Constraint

MSC.Marc contains a generalized tying (constraint) condition option. Any constraint involving linear dependence of nodal degrees of freedom can be included in the stiffness equations.

A tying constraint involves one tied node and one or more retained nodes, and a tying (constraint) condition between the tied and retained nodes. The degrees of freedom (for example, displacements, temperatures) of the tied node are dependent on the degrees of freedom of the retained nodes through the tying condition. In some special tying conditions, the tied node can also be a retained node. The tying condition can be represented by a tying (constraint) matrix. Note that if the tying constraint involves only one retained node, the choice of which node is to be tied to retained is arbitrary.

As a simple example, impose the constraint that the first degree of freedom of node I be equal to that of node J at all times (see Figure 9-11).

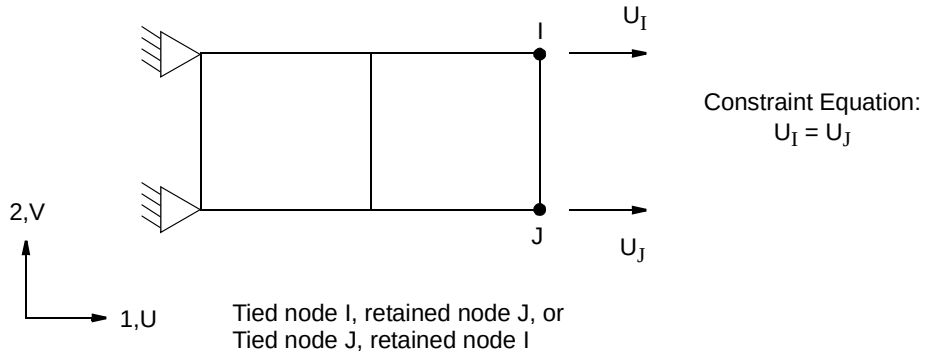


Figure 9-11 Simple Tying Constraint

As a second example, the simulation of a sliding boundary condition requires the input of both the boundary conditions and the tying constraints (see [Figure 9-12](#)).

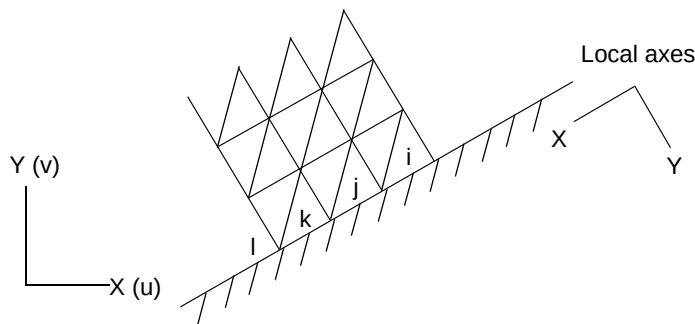


Figure 9-12 Tying Constraint Illustration (Sliding Boundary Conditions)

The example illustrated in [Figure 9-12](#) enforces rigid sliding on the boundary in the local coordinates defined above.

$$v_i = v_j = v_k = v_1 = 0 \quad (9-34)$$

$$u_i = u_j = u_k = u_1 \quad (9-35)$$

The first equation is a set of fixed boundary conditions. The second equation is a constraint equation and can be rewritten as three constraint equations:

$$u_i = u_j \quad (9-36)$$

$$u_k = u_j \quad (9-37)$$

$$u_l = u_j \quad (9-38)$$

These equations express all the u displacements in terms of u_j . In this example, node j is chosen to be the retained node; nodes i , k , and l are tied nodes. You can use the **TYING** option to enter this information.

MSC.Marc has a number of standard tying constraints that can be used for mesh refinement, shell-to-shell, shell-to-beam, beam-to-beam, and shell-to-solid intersections. [Table 9-16](#) through [Table 9-19](#) describe these options. [Table 9-20](#) through [Table 9-22](#) show the tying constraints for pipe elements, shell stiffeners, and nodal degrees of freedom. [Table 9-23](#) summarizes the rigid link constraint.

The **SERVO LINK** option uses the homogeneous linear constraint capability (tying) to input simple constraints of the form

$$u_t = a_1 u_{r1} + a_2 u_{r2} + \dots \quad (9-39)$$

where u_t is a degree of freedom to be constrained; u_{r1} , u_{r2} , ... are the other retained degrees of freedom in this structure; and a_1 , a_2 , ... are constants provided in this option.

You can use the **TYING** or **SERVO LINK** model definition option to enter standard tying constraint information, and the **TYING CHANGE** option to change tying constraints during load incrementation.

The **UFORMS** user subroutine is a powerful method to specify a user-defined constraint equation. This constraint can be nonlinear; for example, it can be dependent on time or previous deformation.

Table 9-16 Tying Constraints for Mesh Refinement

Tying Code	Number of Retained Nodes	Purpose	Remarks
31	2	Refine mesh of first order (linear displacement) elements in two dimensions	Tie interior nodes on refined side to corner nodes in coarse side (see Figure 9-13)
32	3	Refine mesh of second order (quadratic displacement in two dimensions	Tie interior nodes on refined side to edge of element of coarse side (see Figure 9-14)
33	4	Refine mesh of 8-node bricks	Tie interior node on refined side to four corner nodes of an element face on coarse side (see Figure 9-15)
34	8	Refine mesh of 20-node bricks	Tie interior nodes on refined side to eight (four corner, four midside) nodes of element face on coarse side (see Figure 9-16)

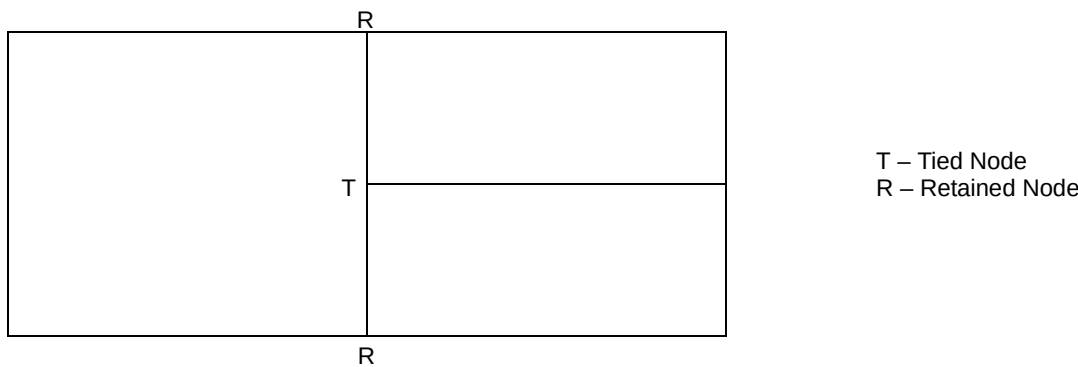


Figure 9-13 Mesh Refinement for 4-Node Quad

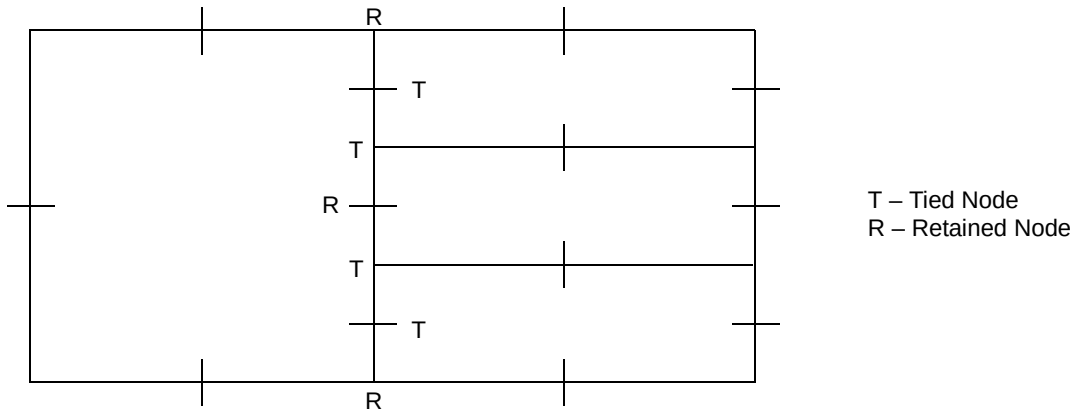


Figure 9-14 Mesh Refinement for 8-Node Quad

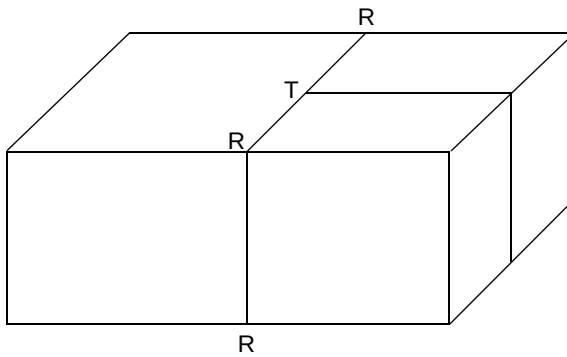


Figure 9-15 Mesh Refinement for 8-Node Brick

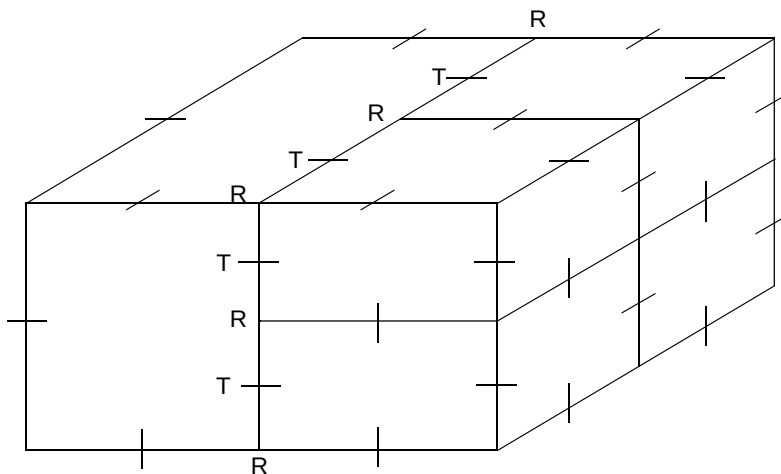


Figure 9-16 Mesh Refinement for 20-Node Brick

Table 9-17 Tying Constraints for Shell-to-Shell Intersection

Tying Code	Number of Retained Nodes	Purpose	Remarks
22	2	Join intersecting shells, Element Type 4, 8, or 24; fully moment-carrying joint	Tied node is also the second retained node*
18	2	Join intersecting shells, Element Type 4, 8, or 14; fully moment-carrying joint	Tied node is also the second retained node
28	2	Join intersecting shells, Element Type 4, 8, or 24; pinned point	Tied node is also the second retained node
24**	2	Join intersecting shells or beams, Element Types 15-17	Tied node is also the second retained node
* Thickness vector must be specified at tied nodes and retained nodes. ** See Figure 9-17.			

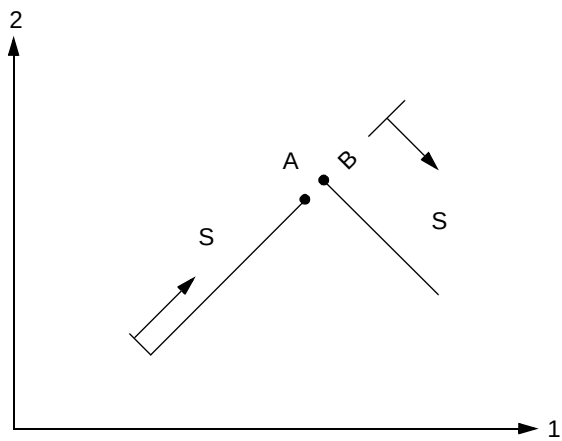


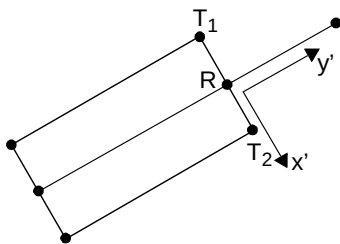
Figure 9-17 Standard Tying Type 24, Tie Shell-to-Shell or Beam-to-Beam; Moment-Carrying

Table 9-18 Tying Constraints for Beam-to-Beam Intersection

Tying Code	Number of Retained Nodes	Purpose	Remarks
13	2	Join Two Elements Type 13 under an arbitrary angle; full moment-carrying joint	Tied node also the second retained node
52	1	Pin joint for Beam Type 14, 25, 52, 76-79, 98	
53	1	Full moment-carrying joint for Beam Types 14, 25, 52, 76-79, 98	Tie interior node on refined side to four corner nodes of an element face on coarse side (see Figure 9-15)

Table 9-19 Tying Constraints for Shell-to-Solid Intersections

Tying Code	Number of Retained Nodes	Purpose	Remarks
23	1	Tie axisymmetric solid node to axisymmetric shell (Element Type 1) node	Tied and retained nodes must be transformed to local system and TRANSFORMATION option invoked (see Figure 9-18).
25	2	Join solid mesh to shell or beam (Type 15 or 16)	Similar to 23 but no transformation needed. Tied node also second retained node. (see Figure 9-19)

**Figure 9-18** Standard Tying Type 23, Tie Solid-to-Shell (Element Type 1)

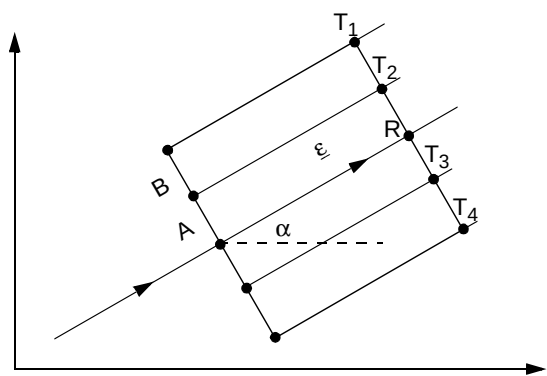


Figure 9-19 Standard Tying Type 25, Tie Solid-to-Shell (Element Type 15)

Table 9-20 Tying Constraints for Pipe Bend Element (Elements 14 and 17)

Tying Code	Number of Retained Nodes	Purpose	Remarks
15	One less than the number of shell nodes in the z-r plane of the section	Special tying types for pipe bend Element 17 to remove rigid body modes	
16	Number of shell nodes in the z-r plane of the section	Special tying types for pipe bend Element 17 to remove rigid body modes	
17	2	Special tying types for pipe bend Element 17 to couple bend section into pipe bend	

Table 9-21 Tying Constraints for Shell Stiffener (Element 13 as a Stiffener on Shell Elements 4 or 8)

Tying Code	Number of Retained Nodes	Purpose	Remarks
19	2	Use beam Element 13 as a stiffener on shell Elements 4 or 8. Tied node is beam node; first retained node is shell node, second is beam node again. Beam node should be on, or close to, the normal to the shell at the shell node	Tied node also second retained node
20	3	Create an extra node in a shell Type 8 Element tied to the interpolated shell displacements with tying Type 21 to tie a beam Element 13 or a stiffener across a shell element	Always used after tying Type 21
21	2	Same as Type 19, but tying beam to an interpolated shell node not a vertex of an element (Element Type 8 only) must be followed by Type 20 to tie interpolated shell node into shell mesh	Must be followed by tying Type 20

Table 9-22 Tying constraints for Nodal Degrees of Freedom

Tying Code	Number of Retained Nodes	Purpose	Remarks
I≤NDEG	1	Tie the Ith degree of freedom at the tied node to the Ith degree of freedom at the retained node	NDEG = number of degrees of freedom per node
100	1	Tie all degrees of freedom at the tied node to the corresponding degrees of freedom at the retained node	

Table 9-22 Tying constraints for Nodal Degrees of Freedom (Continued)

Tying Code	Number of Retained Nodes	Purpose	Remarks
>100	1	Generate several tyings of type ≤NDEG	Tying code is first degree of freedom multiplied by 100 added to last degree of freedom (209 means tie second through ninth degree of freedom at tied node to respective second through ninth degree of freedom at retained node)
<0 (negative integer)		User-defined	User-generated tying type through the UFORMS user subroutine

Rigid Link Constraint

Tying type 80 can be used to define a rigid link between nodes. This capability can be used for both small deformation and large deformation, large rotation problems. In small deformations, a linear constraint equation is used. In addition to the end points of the link, a second retained node must be given. This node is used to calculate and store the rigid body rotations. For two-dimensional problems, this represents a rotation about the Z-axis and this is stored as the first degree of freedom for the node. In 3-D, it should be noted that the rotations/moments about the x, y, and z axes for the second retained node are stored and output as its first, second, and third degrees of freedom, respectively.

A complete rigid region can be modeled by using multiple tying type 80's. In this case, the same two nodes are used for the retained nodes in all of the constraints. Transformations can be applied to the retained nodes so that any kinematic or force boundary conditions can be defined in a local coordinate system. It is not necessary that the same transformations should be applied to both retained nodes. It is also optional to transform the tied nodes to the same or a different local coordinate system.

The rigid links can be used with all elements except types 4, 8, 24, 15, 16, and 17. In addition, it should not be used in rigid plastic analysis.

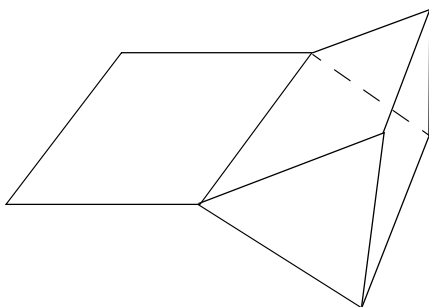
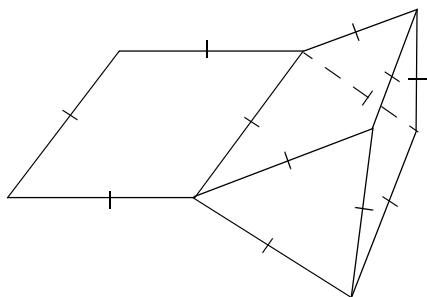
Table 9-23 Rigid Link Constraint

Tying Code	Number of Retained Nodes	Purpose	Remarks
80	2	Define a rigid link between nodes	The second retained node is an unattached node which contains the rotation

Shell-to-Solid Tying

In many problems, a region exists that is modeled with both brick elements and shell elements. A particular case of this is shown in [Figure 9-20](#) and [Figure 9-21](#).

In the first case, an 8-node brick which has been reduced to a triangular prism is connected to a 4-node shell. In the second case, a 20-node brick is connected to an 8-node shell. An automatic constraint equation is developed between the elements. Note that the thickness of the shell must be entered as the brick thickness.

**Figure 9-20** 4-Node Shell-to-Solid Automatic Constraint**Figure 9-21** 8-Node Shell-to-Solid Automatic Constraint

Insert

MSC.Marc provides a model definition option, [INSERT](#), which allows the definition of host bodies and lists of elements or nodes to be inserted in the host bodies. The degrees of freedom of the nodes in the inserted node list or element list are automatically tied using the corresponding degrees of freedom of the nodes in host body elements based on their isoparametric location in the elements.

The [INSERT](#) model definition option can be used to place reinforcing cords or rods, such as 2-D rebar membrane elements, into solid elements.

The [INSERT](#) model definition option can be used to apply point loads in some specific locations other than element nodes. It also can be used to link two different meshes.

If a node to be inserted is also a node of a host body element, no tying is applied to the node.

Transformation must not be used at any nodes of host body elements and at inserted nodes, unless the same set of local coordinate system is used for all nodes involved.

Support Conditions

MSC.Marc provides linear and nonlinear springs and foundations for the modeling of support conditions. For dynamic analysis, a dashpot can also be included. [Table 9-24](#) lists input options for linear and nonlinear springs and elastic foundations.

Table 9-24 Input Options for Springs and Elastic Foundations

Load Description	Input Options		
	Model Definition	History Definition	User Subroutine
Springs	SPRINGS		USPRNG
Elastic Foundation	FOUNDATION	FOUNDATION	USPRNG

The force in the spring is

$$F = K(u_2 - u_1) + D(v_2 - v_1) \tag{9-40}$$

where K is the spring stiffness, u_2 is the displacement of the degree of freedom at the second end of the spring, and u_1 is the displacement of the degree of freedom at the first end of the spring. In a dynamic analysis, D is the damping factor and v_2 and v_1 are the velocities of the nodes.

You can specify the elements in MSC.Marc to be supported on a frictionless, elastic foundation. The foundation supports the structure with a force per unit area (force per unit length, for beams) given by

$$p_n = Ku_n \quad (9-41)$$

where K is the equivalent spring stiffness of the foundation, and u_n is the normal displacement of the surface at a point in the same direction as p_n .

The same conventions apply to the elastic foundation specification as for pressure specification, in terms of the face of the element that is used. The force is applied whether the displacement is tensile or compressive.

Nonlinear spring stiffness for mechanical, thermal, and electrical analyses can be specified through tables using the parameter and model definition option. For more details, refer to Chapters 2 and 3 in *MSC.Marc Volume C: Program Input*.

The **USPRNG** user subroutine allows you to supply or further modify a nonlinear spring stiffness or specify a nonlinear foundation stiffness as a function of prior displacement and force.

Cyclic Symmetry

A special set of tying constraints for continuum elements can be automatically generated by the MSC.Marc program to effectively analyze structures with a geometry and a loading varying periodically about a symmetry axis. [Figure 9-22](#) shows an example where, on the left-hand side, the complete structure is given and, on the right-hand side, a sector to be modeled.

Looking at points A and B on this segment, the displacement vectors should fulfil:

$$u'_B = u_A \quad (9-42)$$

which can also be written as:

$$u_B = Ru_A \quad (9-43)$$

where the transformation matrix R depends on the symmetry axis (which, in the example above, coincides with the global Z-axis) and the sector angle α (see [Figure 9-22](#)). In MSC.Marc, the input for the option **CYCLIC SYMMETRY** consists of the direction vector of the symmetry axis, a point on the symmetry axis and the sector angle α . The following items should be noted:

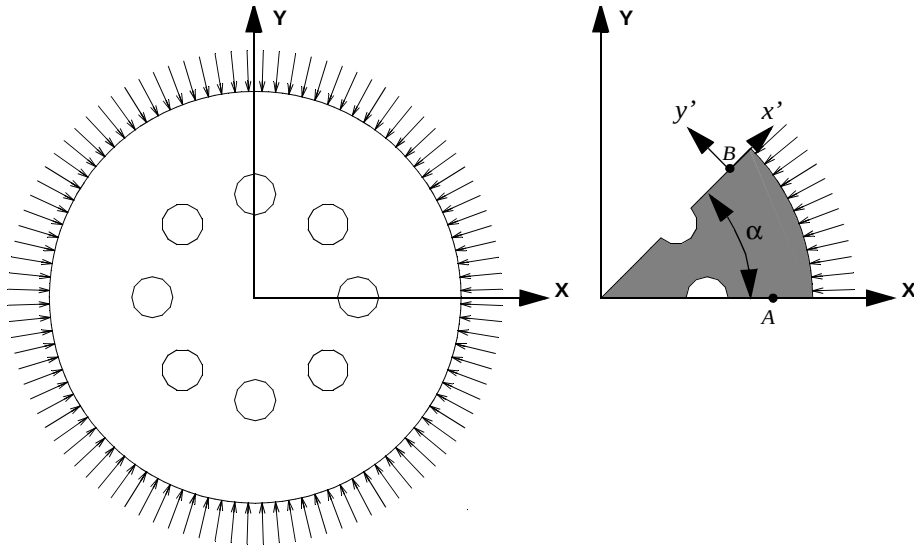


Figure 9-22 Cyclic symmetric structure: complete model (left) and modeled sector (right)

1. The meshes do not need to line up on both sides of a sector (for example, see [Figure 9-23](#)).

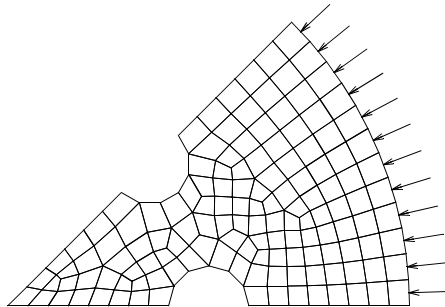


Figure 9-23 Finite element mesh for cyclic symmetric structure with different mesh densities on the sector sides

2. Any shape of the sector sides is allowed provided that upon rotating the sector $360/\alpha$ times about the symmetry axis over the sector angle α will result in the complete model.
3. The **CYCLIC SYMMETRY** option can be combined with the **CONTACT** option. In this case, both sides of the cyclic symmetry sectors need to belong to the same contact body.

4. The **CYCLIC SYMMETRY** option can be combined with global remeshing.
5. In a coupled thermo-mechanical analysis, the temperature is forced to be cyclic symmetric ($T_A = T_B$ as in [Figure 9-22](#)).
6. A nodal point on the symmetry axis is automatically constrained in the plane perpendicular to the symmetry axis.
7. The possible rigid body motion about the symmetry axis can be automatically suppressed.
8. Cyclic Symmetry is:
 - a. valid for only the continuum elements. However, the presence of beams and shells is allowed, but there is no connection of shells to shells, so the shell part can, for example, be a turbine blade and the volume part is a turbine rotor. The blade is connected to the rotor and if there are 20 blades, 1/20 of the rotor is modeled and one complete blade.
 - b. valid for nonlinear static analysis including remeshing as well as coupled analysis.
 - c. invalid for pure heat transfer.
 - d. valid for all analysis involving contact.
 - e. valid also for: eigenvalue analysis such as buckling or modal analysis, harmonic analysis, and transient dynamic analysis. However, there are restrictions in the case of modal analysis which are described in more detail in the following paragraphs.
9. In order to check which nodes get tying constraints due to cyclic symmetry, nodal post code 38 (contact status) can be used; its value will be 2.

To prevent confusion, it must be emphasized that the cyclic symmetry feature described above is different than linear cyclic symmetry commonly used in modal analysis where physical quantities such as, x_n , displacements, forces, stresses and temperature in the n -th segment are expanded in a Fourier series with terms of the cyclic components, u^k , in the fundamental region, like:

$$x_n = \frac{1}{N}u^0 + \sqrt{\frac{2}{N}} \sum_{k=1}^K [u^{k,c} \cos(n-1)k\alpha + u^{k,s} \sin(n-1)k\alpha] + \frac{(-1)^{n-1}}{\sqrt{N}} u^{\frac{N}{2}} \quad (9-44)$$

where k is the harmonic order; N is the total number of sectors; α is the fundamental inter-sector phase shift defined as $2\pi/N$; and K is defined as:

$$K = \begin{cases} \frac{N-1}{2} & \text{if } N \text{ is odd} \\ \frac{N-2}{2} & \text{if } N \text{ is even} \end{cases} \quad (9-45)$$

There are considerable savings in both computing time and data storage associated with the use of the linear cyclic symmetry concept. Assuming a finite element model with a sector size of m degrees of freedom, a real-valued cyclic symmetry approach leads, in the worst case, to one eigenvalue problem of size m and $(N-1)/2$ eigenvalue problems of size $2m$; a complex approach leads to N eigenvalue problems of size m ; while the full analysis leads to a single, but very costly, eigenvalue problem of size Nm .

Although linear cyclic symmetry can reduce the problem size greatly, it is restricted to linear analysis, and the sector must have its surface mesh on the symmetry planes to be identical on each side of the sector. The nonlinear cyclic symmetry implemented in MSC.Marc can be used for nonlinear problems, such as contact, and the nodes do not need to line up on both symmetry planes of the sector (for example, see [Figure 9-23](#)).

MSC.Nastran RBE2 and RBE3

The basic formulation of the MSC.Nastran rigid body elements RBE2 and RBE3 is similar to tying type 80, which is also known as a rigid link. However, they offer additional flexibility because the degrees of freedom may be partly unconstrained and a list of tied/retained nodes can be entered with RBE2 and RBE3.

RBE2

As an example of the use of the [RBE2](#) option, consider [Figure 9-24](#), where a cylindrical tube is shown. The tube is clamped at one end and loaded by a torque at the other end.

If it is assumed that the loaded end remains circular with a constant radius, then a single [RBE2](#) option can be easily used instead of a number of tyings of type 80. With the [RBE2](#) option, node 1 is defined as the retained or reference node and nodes 2-13 are defined as the tied nodes, with all the degrees of freedom constrained.

However, if the tube is allowed to expand in radial direction, then tying type 80 cannot be used. However, the [RBE2](#) option can still be used, but now the radial displacement is not included in the list of degrees of freedom to be constrained.

RBE2 can also be used in a large displacement analysis.

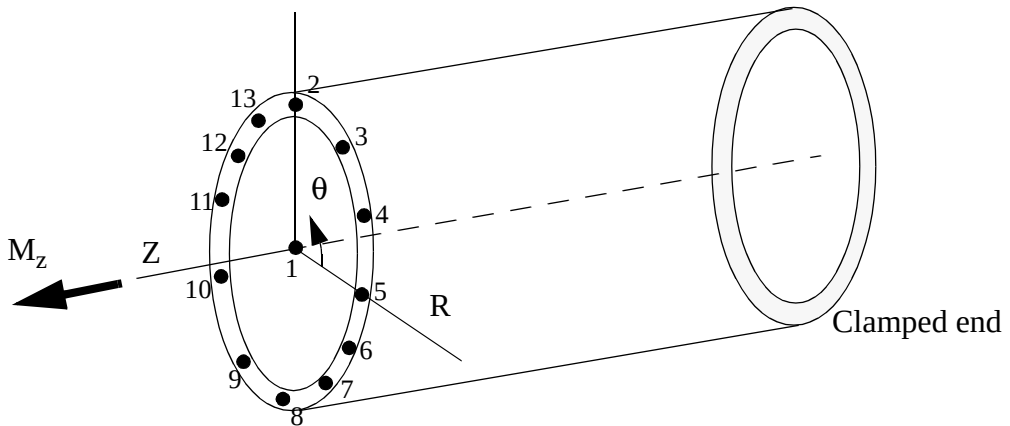


Figure 9-24 RBE2 example: Cylindrical tube loaded in torsion

RBE3

The **RBE3** option is a powerful tool to distribute applied loads in an FE model. Forces and moments applied to reference or tied node are distributed to a set of independent degrees of freedom based on the **RBE3** geometry and local weight factors.

The way in which the forces are distributed is analogous to the classical bolt pattern analysis. Consider the bolt pattern shown in **Figure 9-25** with a force F_A and moment M_A acting at reference point A. The force and moment can be transferred directly to the weighted center of gravity along with the moment produced by the force offset e .

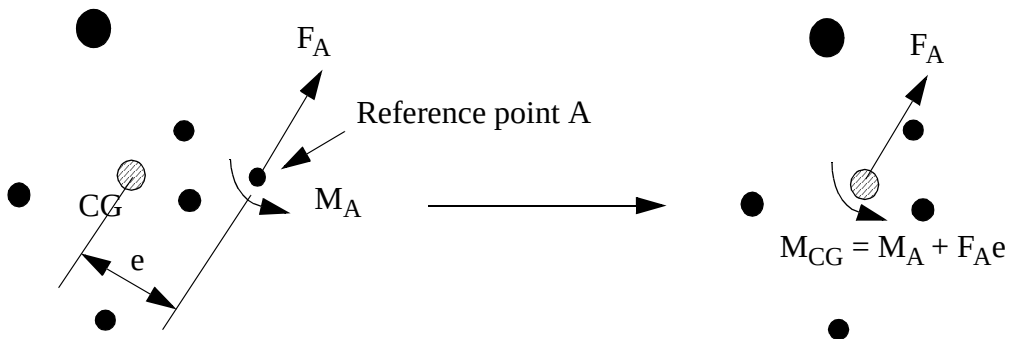
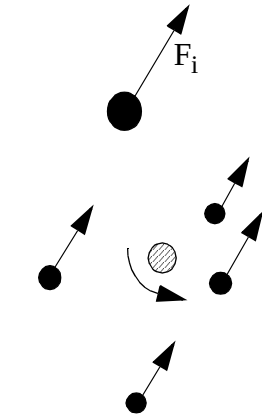


Figure 9-25 Transfer of force and moment on a reference point to the weighted center of gravity (CG)

The force is distributed to the bolts proportional to their weighting factors. The moment is distributed as forces proportional to their distance from the center of gravity times their weighting factors, as shown in [Figure 9-26](#). The total force acting on the bolts is equal to the sum of the two forces. These results apply to both in-plane and out-of-plane loadings.

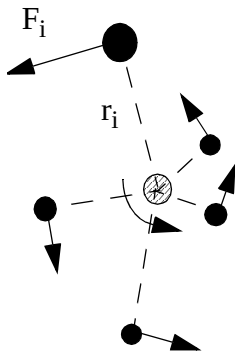
As an example, consider the cantilever plate model with a single quad shell element shown in [Figure 9-27](#). The plate is subjected to nonuniform pressure represented by a resultant force acting at a distance of 10 mm from the center of gravity. The easiest way to apply the pressure is to use an [RBE3](#) option to distribute the resultant load to each of the four corner points. Setting the weighting factors to be equal for all nodes, results in a force distribution based exclusively on the spatial location of the connected nodes. In this way, nodes 3 and 4 will get 3N each and nodes 1 and 2 will get 2N each.

RBE3 cannot be used in large displacement analysis.



Force distribution:

$$F_i = F_A \left(\frac{\omega_i}{\sum \omega_k} \right)$$



Moment distribution:

$$F_i = \left(\frac{M_{CG} \omega_i r_i}{\sum \omega_k r_k^2} \right)$$

where: F_i = force at DOF i

ω_i = weighting factor for DOF i

r_i = radius from the weighted center
of gravity to point i

Figure 9-26 RBE3 Force and Moment Distribution

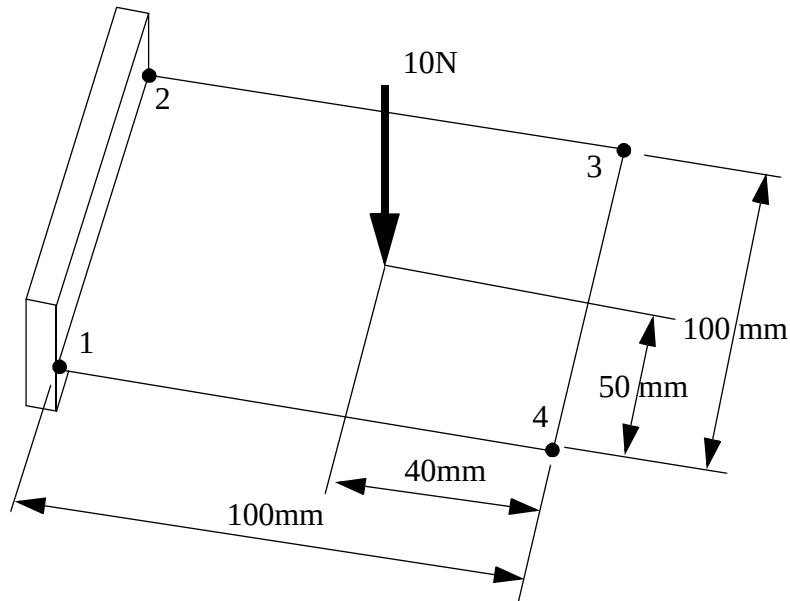


Figure 9-27 Using the RBE3 Option to Represent a Nonuniform Pressure Load

CHAPTER
10

Element Library

- Truss Elements
- Membrane Elements
- Continuum Elements
- Beam Elements
- Plate Elements
- Shell Elements
- Heat Transfer Elements
- Special Elements
- Incompressible Elements
- Constant Dilatation Elements
- Reduced Integration Elements
- Continuum Composite Elements
- Fourier Elements

- Semi-infinite Elements
- Cavity Surface Elements
- Assumed Strain Formulation
- Follow Force Stiffness Contribution
- Explicit Dynamics
- Adaptive Mesh Refinement
- References

MSC.Marc includes an extensive element library. The element library allows you to model various types of one-, two-, and three-dimensional structures, such as plane stress and plane strain structures, axisymmetric structures, full three-dimensional solid structures, and shell-type structures. [Table 10-1](#) lists the element types in the library. [Table 10-2](#) shows the structural classification of elements. (See [MSC.Marc Volume B: Element Library](#) for a detailed description of each element, a reference to the use of each element, and recommendations concerning the selection of element types for analysis).

After you select an element type or a combination of several element types for your analysis with the [ELEMENTS](#) or [SIZING](#) parameter, you must prepare the necessary input data for the element(s). In general, the data consist of element connectivity, thickness for two-dimensional beam, plate and shell elements, cross section for three-dimensional beam elements, coordinates of nodal points, and face identifications for distributed loadings.

You can choose different element types to represent various parts of the structure in an analysis. If there is an incompatibility between the nodal degrees of freedom of the elements, you have to provide appropriate tying constraints to ensure the compatibility of the displacement field in the structure. MSC.Marc assists you by providing many standard tying constraint options, but you are responsible for the consistency of your analysis.

You can use almost all of the MSC.Marc elements for both linear and nonlinear analyses, with the following exceptions.

- No plasticity or cracking is allowed in the elastic beams, the shear-panel, and the Fourier elements.

- The updated Lagrange and finite strain plasticity features are not available for all elements.
- Plasticity is available for the Herrmann elements when using the $F^e F^p$ formulation only.
- Only heat links and two-dimensional heat transfer elements can be used for hydrodynamic bearing analysis.
- Fourier analysis can be carried out only for a limited number of axisymmetric elements.

MSC.Marc defines all continuum elements in the global coordinate system. Truss, beam, plate, and shell elements are defined in a local coordinate system and the resulting output must be interpreted accordingly. You should give special attention to the use of these elements if the material properties have preferred orientations. The **ORIENTATION** option is available to define the preferred directions.

All MSC.Marc elements are numerically integrated. Element quantities, such as stresses, strains, and temperatures, are calculated at each integration point of the element if you use the **ALL POINTS** parameter. This is the default in MSC.Marc. These quantities are computed only at the centroid of the element, if the **CENTROID** parameter is used.

Table 10-1 Element Library

Element	Updated Lagrange Available	Code
2-node axisymmetric shell element	Yes	(1)
Axisymmetric triangular ring element	Yes	(2)
Two-dimensional (plane stress) 4-node, isoparametric quadrilateral	Yes	(3)
Curved quadrilateral thin-shell element	Yes	(4)
Beam column	No	(5)
Two-dimensional plane strain, constant stress triangle	Yes	(6)
8-node isoparametric three-dimensional hexahedron	Yes	(7)
3-node triangular arbitrary shell	Yes	(8)
Three-dimensional truss element	Yes	(9)
Axisymmetric quadrilateral element (isoparametric)	Yes	(10)
Plane strain quadrilateral element (isoparametric)	Yes	(11)
Gap and friction element	No	(12)
Open-section beam	No	(13)

Table 10-1 Element Library (Continued)

Element	Updated Lagrange Available	Code
Closed-section beam	Yes	(14)
Isoparametric 2-node axisymmetric shell	Yes	(15)
Isoparametric 2-node curved beam	Yes	(16)
Pipe-bend element	No	(17)
4-node isoparametric membrane	No	(18)
Generalized plane-strain quadrilateral	Yes	(19)
Axisymmetric torsional quadrilateral	Yes	(20)
Three-dimensional 20-node brick	Yes	(21)
Curved quadrilateral thick-shell element	Yes	(22)
Three-dimensional, 20-node rebar element	Yes	(23)
Curved quadrilateral shell element	Yes	(24)
Closed-section beam in three dimensions	Yes	(25)
Plane stress, 8-node distorted quadrilateral	Yes	(26)
Plane strain, 8-node distorted quadrilateral	Yes	(27)
Axisymmetric, 8-node distorted quadrilateral	Yes	(28)
Generalized plane strain distorted quadrilateral	Yes	(29)
Membrane 8-node distorted quadrilateral	No	(30)
Elastic Curved Pipe (Elbow)/Straight Beam Element	No	(31)
Plane strain, 8-node distorted quadrilateral, for incompressible behavior	Yes	(32)
Axisymmetric 8-node distorted quadrilateral, for incompressible behavior	Yes	(33)
Generalized plane strain 8-node distorted quadrilateral for incompressible behavior	No	(34)
Three-dimensional 20-node brick for incompressible behavior	Yes	(35)
Heat transfer element (three-dimensional link)	No	(36)
Heat transfer element (arbitrary planar triangle)	No	(37)
Heat transfer element (arbitrary axisymmetric triangle)	No	(38)
Heat transfer element (planar bilinear quadrilateral)	No	(39)
Heat transfer element (axisymmetric bilinear quadrilateral)	No	(40)
Heat transfer element (8-node planar biquadratic quadrilateral)	No	(41)
Heat transfer element (8-node axisymmetric biquadratic quadrilateral)	No	(42)

Table 10-1 Element Library (Continued)

Element	Updated Lagrange Available	Code
Heat transfer element (three-dimensional 8-node brick)	No	(43)
Heat transfer element (three-dimensional 20-node brick)	No	(44)
Curved Timoshenko beam element in a plane	Yes	(45)
Plane strain rebar element	Yes	(46)
Generalized plane strain rebar element	Yes	(47)
Axisymmetric rebar element	Yes	(48)
Triangular thin shell element	Yes	(49)
Triangular heat transfer shell element	No	(50)
Cable Element	No	(51)
Elastic beam	Yes	(52)
Plane stress 8-node quadrilateral with reduced integration	Yes	(53)
Plane strain 8-node distorted quadrilateral with reduced integration	Yes	(54)
Axisymmetric 8-node distorted quadrilateral with reduced integration	Yes	(55)
Generalized plane strain 10-node distorted quadrilateral with reduced integration	Yes	(56)
Three-dimensional 20-node brick with reduced integration	Yes	(57)
Plane strain 8-node distorted quadrilateral for incompressible behavior with reduced integration	Yes	(58)
Axisymmetric 8-node distorted quadrilateral for incompressible behavior with reduced integration	Yes	(59)
Generalized plane strain 10-node distorted quadrilateral incompressible behavior with reduced integration	No	(60)
Three-dimensional 20-node brick for incompressible behavior with reduced integration	Yes	(61)
Axisymmetric 8-node quadrilateral for arbitrary loading	No	(62)
Axisymmetric 8-node quadrilateral for arbitrary loading, and incompressible behavior	No	(63)
Isoparametric 3-node truss element	Yes	(64)
Heat transfer element 3-node truss (three-dimensional link)	No	(65)
8-node axisymmetric with twist, for incompressible behavior	No	(66)
8-node axisymmetric with twist	No	(67)
Elastic 4-node shear panel	No	(68)

Table 10-1 Element Library (Continued)

Element	Updated Lagrange Available	Code
Heat transfer element (8-node, planar biquadratic quadrilateral with reduced integration)	No	(69)
Heat transfer element (8-node, axisymmetric biquadratic quadrilateral with reduced integration)	No	(70)
Heat transfer element (three-dimensional 20-node brick with reduced integration)	No	(71)
Bilinear discrete Kirchhoff	Yes	(72)
Axisymmetric 8-node quadrilateral for arbitrary loading with reduced integration	No	(73)
Axisymmetric 8-node quadrilateral for arbitrary loading, for incompressible behavior with reduced integration	No	(74)
Bilinear thick shell	Yes	(75)
Thin-walled beam in three dimensions without warping	Yes	(76)
Thin-walled beam in three dimensions including warping	Yes	(77)
Thin-walled beam in three dimensions without warping	Yes	(78)
Thin-walled beam in three dimensions including warping	Yes	(79)
Plane strain 7-node quadrilateral for incompressible behavior	Yes	(80)
Generalized plane strain 7-node quadrilateral for incompressible behavior	No	(81)
Axisymmetric 5-node quadrilateral for incompressible behavior	Yes	(82)
Axisymmetric 5-node quadrilateral with twist for incompressible behavior	No	(83)
8-node three-dimensional hexahedron for incompressible behavior	Yes	(84)
Heat transfer element (bilateral shell)	No	(85)
Heat transfer element (curved quadrilateral shell)	No	(86)
Heat transfer element (curved axisymmetric shell)	No	(87)
Heat transfer element (linear axisymmetric shell)	No	(88)
Thick curved axisymmetric shell	Yes	(89)
Thick curved axisymmetric shell for arbitrary loading	No	(90)
Linear plane strain semi-infinite element	No	(91)
Linear axisymmetric semi-infinite element	No	(92)
Quadratic plane strain semi-infinite element	No	(93)
Quadratic axisymmetric semi-infinite element	No	(94)
Quadrilateral axisymmetric element with bending	No	(95)

Table 10-1 Element Library (Continued)

Element	Updated Lagrange Available	Code
8-node distorted quadrilateral axisymmetric element with bending	No	(96)
Double gap and friction element for use with Element Types 95 or 96	No	(97)
Elastic beam with transverse shear	Yes	(98)
Heat transfer link	No	(99)
Heat transfer link	No	(100)
6-node planar semi-infinite heat transfer	No	(101)
6-node axisymmetric semi-infinite heat transfer	No	(102)
9-node planar semi-infinite heat transfer	No	(103)
9-node axisymmetric semi-infinite heat transfer	No	(104)
12-node 3-D semi-infinite heat transfer	No	(105)
27-node 3-D semi-infinite heat transfer	No	(106)
12-node 3-D semi-infinite	No	(107)
27-node 3-D semi-infinite	No	(108)
8-node 3-D magnetostatic	No	(109)
12-node 3-D semi-infinite magnetostatic	No	(110)
4-node Quadrilateral Planar Electromagnetic	No	(111)
4-node Quadrilateral Axisymmetric Electromagnetic	No	(112)
8-node Quadrilateral Three-Dimensional Electromagnetic	No	(113)
4-node Quadrilateral Plane Stress, Reduced Integration with Hourglass Control	Yes	(114)
4-node Quadrilateral Plane Strain, Reduced Integration with Hourglass Control	Yes	(115)
4-node Quadrilateral Axisymmetric, Reduced Integration with Hourglass Control	Yes	(116)
8-node Three-Dimensional Brick, Reduced Integration with Hourglass Control	Yes	(117)
Incompressible 4+1-node, Quadrilateral, Plane Strain, Reduced Integration with Hourglass Control	No	(118)
Incompressible 4+1-node, Quadrilateral, Axisymmetric, Reduced Integration with Hourglass Control	No	(119)
Incompressible 8+1-node, Three-Dimensional Brick, Reduced Integration with Hourglass Control	No	(120)

Table 10-1 Element Library (Continued)

Element	Updated Lagrange Available	Code
4-node, Heat Transfer Planar, Reduced Integration with Hourglass Control	No	(121)
4-node, Heat Transfer Axisymmetric, Reduced Integration with Hourglass Control	No	(122)
8-node, Heat Transfer Three-Dimensional Brick, Reduced Integration with Hourglass Control	No	(123)
6-node, Plane Stress Triangle	Yes	(124)
6-node, Plane Strain Triangle	Yes	(125)
6-node, Axisymmetric Triangle	Yes	(126)
10-node, Tetrahedron	Yes	(127)
Incompressible, 6-node Triangle	Yes	(128)
Incompressible, 6-node Triangle	Yes	(129)
Incompressible, 10-node Tetrahedral	Yes	(130)
6-node, Heat Transfer Planar	No	(131)
6-node, Heat Transfer Plane Axisymmetric	No	(132)
10-node, Heat Transfer Tetrahedral	No	(133)
4-node, Tetrahedral	Yes	(134)
4-node, Heat Transfer Tetrahedral	No	(135)
3-node, Thin Shell	Yes	(138)
4-node, Thin Shell	Yes	(139)
4-node, Thick Shell, Reduced Integration with Hourglass Control	Yes	(140)
8-node Axisymmetric Rebar with Twist	Yes	(142)
4-node Plane Strain Rebar	Yes	(143)
4-node Axisymmetric Rebar	Yes	(144)
4-node Axisymmetric Rebar with Twist	Yes	(145)
8-node Brick Rebar	Yes	(146)
4-node Membrane Rebar	Yes	(147)
8-node Membrane Rebar	Yes	(148)
8-node Composite Brick	Yes	(149)
20-node Composite Brick	Yes	(150)
4-node Plane Strain Composite	Yes	(151)
4-node Axisymmetric Composite	Yes	(152)
8-node Plane Strain Composite	Yes	(153)

Table 10-1 Element Library (Continued)

Element	Updated Lagrange Available	Code
8-node Axisymmetric Composite	Yes	(154)
3 + 1-node Incompressible Plane Strain Triangle	Yes	(155)
3 + 1-node Incompressible Axisymmetric Triangle	Yes	(156)
4 + 1-node Incompressible Tetrahedron	Yes	(157)
4-node Isoparametric Quadrilateral 2-D (Plane Stress), with piezoelectric capability	No	(160)
4-node Plane Strain Quadrilateral Element (Isoparametric) with piezoelectric capability	No	(161)
4-node Axisymmetric Quadrilateral Element (Isoparametric) with piezoelectric capability	No	(162)
8-node Isoparametric 3-D Hexahedron with piezoelectric capability	No	(163)
4-node, Tetrahedral with piezoelectric capability	No	(164)
2-node Plane Strain Rebar Membrane	Yes	(165)
2-node Axisymmetric Rebar Membrane	Yes	(166)
2-node Axisymmetric Rebar Membrane with Twist	Yes	(167)
3-node Plane Strain Rebar Membrane	Yes	(168)
3-node Axisymmetric Rebar Membrane	Yes	(169)
3-node Axisymmetric Rebar Membrane with Twist	Yes	(170)
2-node Planar Cavity Surface	No	(171)
2-node Axisymmetric Cavity Surface	No	(172)
3-node Triangular Cavity Surface	No	(173)
4-node Quadrilateral Cavity Surface	No	(174)
8-node Heat Transfer Composite Brick	No	(175)
20-node Heat Transfer Composite Brick	No	(176)
4-node Planar Heat Transfer Composite	No	(177)
4-node Axisymmetric Heat Transfer Composite	No	(178)
8-node Planar Heat Transfer Composite	No	(179)
8-node Axisymmetric Heat Transfer Composite	No	(180)

Table 10-2 Structural Classification of Elements

Element Structural Type	Element Number	Function	Remarks
Three-dimensional truss	9	Linear	2-node straight
	12	Linear	4-node straight gap and friction
	51	Analytic	2-node cable element
	64	Quadratic	3-node curved
	97	Linear	4-node straight double gap and friction
Two-dimensional beam column	5	Linear/cubic	2-node straight
	16	Cubic	2-node curved
	45	Cubic	3-node curved Timoshenko theory
Three-dimensional beam column	13	Cubic	2-node curved open section
	14	Linear/cubic	2-node straight closed section
	25	Cubic	2-node straight closed section
	31	Analytic	2-node elastic
	52	Linear/cubic	2-node straight elastic
	76	Linear/cubic	2 + 1-node straight closed section; use with Element 72
	77	Linear/cubic	2 + 1-node straight open section; use with Element 72
	78	Linear/cubic	2-node straight closed section; use with Element 75
	79	Linear/cubic	2-node straight open section
	98	Linear	2-node straight Timoshenko theory
Axisymmetric shell	1	Linear/cubic	2-node straight
	15	Cubic	2-node curved
	89	Quadratic	3-node curved thick shell theory
	90	Quadratic	3-node curved with arbitrary loading; thick shell theory
Plane stress	3	Linear	4-node quadrilateral
	26	Quadratic	8-node quadrilateral
	53	Quadratic	8-node reduced integration quadrilateral
	114	Linear/Assumed strain	4-node quadrilateral, reduced integration, with hourglass control
	124	Quadratic	6-node triangle
	160	Linear	4-node quadrilateral with piezoelectric capability

Table 10-2 Structural Classification of Elements (Continued)

Element Structural Type	Element Number	Function	Remarks
Plane strain	6	Linear	3-node triangle
	11	Linear	4-node quadrilateral
	27	Quadratic	8-node quadrilateral
	54	Quadratic	8-node reduced integration quadrilateral
	91	Linear/special	6-node semi-infinite
	93	Quadratic/special	9-node semi-infinite
	115	Linear/Assumed strain	4-node quadrilateral, reduced integration, with hourglass control
	125	Quadratic	6-node triangle
	161	Linear	4-node quadrilateral with piezoelectric capability
Generalized plane strain	19	Linear	4 + 2-node quadrilateral
	29	Quadratic	8 + 2-node quadrilateral
	56	Quadratic	8 + 2-node reduced integration quadrilateral
Axisymmetric solid	2	Linear	3-node triangle
	10	Linear	4-node quadrilateral
	20	Linear	4-node quadrilateral with twist
	28	Quadratic	8-node quadrilateral
	55	Quadratic	8-node reduced integration quadrilateral
	62	Quadratic	8-node quadrilateral with arbitrary loading
	67	Quadratic	8-node quadrilateral with twist
	73	Quadratic	8-node reduced integration quadrilateral and arbitrary loading
	92	Linear/special	6-node semi-infinite
	94	Quadratic/special	9-node semi-infinite
	95	Linear	4-node quadrilateral with bending
	96	Quadratic	8-node quadrilateral with bending
	116	Linear/Assumed strain	4-node quadrilateral, reduced integration, with hourglass control
	126	Quadratic	6-node triangle
	162	Linear	4-node quadrilateral with piezoelectric capability
Membrane three-dimensional	18	Linear	4-node quadrilateral
	30	Quadratic	8-node quadrilateral

Table 10-2 Structural Classification of Elements (Continued)

Element Structural Type	Element Number	Function	Remarks
Doubly-curved thin shell	4	Cubic	4-node curved quadrilateral
	8	Fractional cubic	3-node curved triangle
	24	Cubic patch	4 + 4-node curved quadrilateral
	49	Linear	3 + 3-node curved triangle discrete Kirchhoff
	72	Linear	4 + 4-node twisted quadrilateral discrete Kirchhoff
	138	Linear	3-node triangle discrete Kirchhoff
	139	Linear	4-node twisted quadrilateral discrete Kirchhoff
Doubly-curved thick shell	22	Quadratic	8-node curved quadrilateral with reduced integration
	75	Linear	4-node twisted quadrilateral
	140	Linear	4-node twisted quadrilateral, reduced integration with hourglass control
Three-dimensional solid	7	Linear	8-node hexahedron
	21	Quadratic	20-node hexahedron
	57	Quadratic	20-node reduced integration hexahedron
	107	Linear/special	12-node semi-infinite
	108	Quadratic	27-node semi-infinite special
	117	Linear/Assumed strain	8-node hexahedron, reduced integration with hourglass control
	127	Quadratic	10-node tetrahedron
	134	Linear	4-node tetrahedron
	163	Linear	8-node hexahedron with piezoelectric capability
	164	Linear	4-node tetrahedron with piezoelectric capability
Incompressible plane strain	32	Quadratic	8-node quadrilateral
	58	Quadratic	8-node reduced integration quadrilateral
	80	Linear	4 + 1-node quadrilateral
	118	Linear/Assumed strain	4 + 1-node quadrilateral, reduced integration with hourglass control
	128	Quadratic	6-node triangle
	155	Linear	3 + 1-node triangle
Incompressible generalized plane strain	34	Quadratic	8 + 2-node quadrilateral
	60	Quadratic	8 + 2-node with reduced integration quadrilateral
	81	Linear	4 + 3-node quadrilateral

Table 10-2 Structural Classification of Elements (Continued)

Element Structural Type	Element Number	Function	Remarks
Incompressible axisymmetric	33	Quadratic	8-node quadrilateral
	59	Quadratic	8-node with reduced integration quadrilateral
	63	Quadratic	8-node quadrilateral with arbitrary loading
	66	Quadratic	8-node quadrilateral with twist
	74	Quadratic	8-node reduced integration quadrilateral with arbitrary loading
	82	Linear	4 + 1-node quadrilateral
	83	Linear	4 + 1-node quadrilateral with twist
	119	Linear/Assumed strain	4 + 1-node quadrilateral reduced integration, with hourglass control
	129	Quadratic	6-node triangle
	156	Linear	3 + 1-node triangle
Incompressible three-dimensional solid	35	Quadratic	20-node hexahedron
	61	Quadratic	20-node with reduced integration hexahedron
	84	Linear	8 + 1-node hexahedron node 9
	120	Linear/Assumed strain	8 + 1-node hexahedron, reduced integration with hourglass control
	130	Quadratic	10-node tetrahedron
	157	Linear	4 + 1-node tetrahedron
Pipe bend	17	Cubic	2-nodes in-section; 1-node out-of-section
	31	Special	2-node elastic

Table 10-2 Structural Classification of Elements (Continued)

Element Structural Type	Element Number	Function	Remarks
Rebar Elements	23	Quadratic	20-node hexahedron
	46	Quadratic	8-node quadrilateral plane strain
	47	Quadratic	8 + 2-node quadrilateral generalized plane strain
	48	Quadratic	8-node quadrilateral axisymmetric
	142	Quadratic	8-node axisymmetric with twist
	143	Linear	4-node plane strain
	144	Linear	4-node axisymmetric
	145	Linear	4-node axisymmetric with twist
	146	Linear	8-node hexahedron
	147	Linear	4-node membrane
	148	Quadratic	8-node membrane
	165	Linear	2-node plane strain membrane
	166	Linear	2-node axisymmetric membrane
	167	Linear	2-node axisymmetric membrane with twist
	168	Quadratic	2-node plain strain membrane
	169	Quadratic	2-node axisymmetric membrane
	170	Quadratic	2-node axisymmetric membrane with twist
Cavity surface elements	171	Linear	2-node straight
	172	Linear	2-node straight, axisymmetric
	173	Linear	3-node triangle
	174	Linear	4-node quadrilateral
Continuum Composite	149	Linear	8-node brick
	150	Quadratic	20-node brick
	151	Linear	4-node quadrilateral plane strain
	152	Linear	4-node quadrilateral axisymmetric
	153	Quadratic	8-node quadrilateral plane strain
	154	Quadratic	8-node quadrilateral axisymmetric
Three-dimensional shear panel	68	Linear	4-node quadrilateral
Heat conduction three-dimensional link	36	Linear	2-node straight
	65	Quadratic	3-node curved

Table 10-2 Structural Classification of Elements (Continued)

Element Structural Type	Element Number	Function	Remarks
Heat conduction planar	37	Linear	3-node triangle
	39	Linear	4-node quadrilateral
	41	Quadratic	8-node quadrilateral
	69	Quadratic	8-node reduced integration quadrilateral
	101	Linear/special	6-node semi-infinite
	103	Linear/special	9-node semi-infinite
	121	Linear	4-node quadrilateral, reduced integration, with hourglass control
	131	Quadratic	6-node triangular
Heat conduction axisymmetric	38	Linear	3-node triangle
	40	Linear	4-node quadrilateral
	42	Quadratic	8-node quadrilateral
	70	Quadratic	8-node reduced integration quadrilateral
	102	Linear/special	6-node semi-infinite
	104	Quadratic/special	9-node semi-infinite
	122	Linear	4-node quadrilateral, reduced integration, with hourglass control
	132	Quadratic	6-node triangular
Heat conduction solids	43	Linear	8-node hexahedron
	44	Quadratic	20-node hexahedron
	71	Quadratic	20-node reduced integration hexahedron
	105	Linear/special	12-node semi-infinite
	106	Quadratic/special	27-node semi-infinite
	123	Linear	8-node hexahedron, reduced integration, with hourglass control
	133	Quadratic	10-node tetrahedron
	135	Linear	4-node tetrahedron
Heat conduction shell	50	Linear	3-node triangle
	85	Linear	4-node quadrilateral
	86	Quadratic	8-node quadrilateral
Heat conduction axisymmetric shell	87	Quadratic	3-node curved
	88	Linear	2-node straight

Table 10-2 Structural Classification of Elements (Continued)

Element Structural Type	Element Number	Function	Remarks
Heat conduction Continuum Composite	175	Linear	8-node brick
	176	Quadratic	20-node brick
	177	Linear	4-node quadrilateral planar
	178	Linear	4-node quadrilateral axisymmetric
	179	Quadratic	8-node quadrilateral planar
	180	Quadratic	8-node quadrilateral axisymmetric
Magnetostatic three-dimensional solids	109	Linear	8-node hexahedron
	110	Linear/special	12-node semi-infinite
Electromagnetic Planar	111	Linear	4-node quadrilateral
Electromagnetic Axisymmetric	112	Linear	4-node quadrilateral
Electromagnetic Solid	113	Linear	8-node hexahedron

Distributed loads can be applied along element edges, over element surfaces, or in the volume of the element. MSC.Marc automatically evaluates consistent nodal forces through numerical integration. (See [MSC.Marc Volume B: Element Library](#) for details on this process). Concentrated forces must be applied at the nodal points.

All plate and shell elements can be used in a composite analysis. You can have a variable thickness shell, and control the thickness and material property and orientation for each layer. For the thick shell elements (types [22](#), [45](#), [75](#), or [140](#)), the interlaminar shear can also be calculated.

Five concepts differentiate the various elements types. These concepts are summarized in [Table 10-1](#) and [Table 10-2](#) and are listed below.

1. The type of geometric domain that the element is modeling. These geometric domains are:

- Truss
- Membrane
- Beam
- Plate
- Shell
- Plane stress
- Plane strain
- Generalized plane strain
- Axisymmetric
- Three-dimensional solid
- Special

2. The type of interpolation (shape) functions used in the element. These functions are:

Linear
 Quadratic
 Cubic
 Hermitian
 Special

The interpolation function is used to describe the displacement at an arbitrary point in the body.

$$u_i(x) = N_i(x)\bar{u}_i \quad (10-1)$$

where $u(x)$ is the displacement at x , N are the interpolation (shape) functions, and $\bar{u}$ are the generalized nodal displacements.

Engineering strain is

$$\varepsilon_{ij}(x) = 1/2 \left(\frac{\partial u_i(x)}{\partial x_j} + \frac{\partial u_j(x)}{\partial x_i} \right) \quad (10-2)$$

Therefore, the computational evaluation is

$$\varepsilon_{ij}(x) = 1/2 \left(\frac{\partial N_i \bar{u}_i}{\partial x_j} + \frac{\partial N_j \bar{u}_j}{\partial x_i} \right) = \beta_{ij} \bar{u}_i \quad (10-3)$$

Hence,

$$\beta_{ij} = \frac{\partial N_i}{\partial x_j} \quad (10-4)$$

3. The number of nodes in a particular element.
4. The number of degrees of freedom associated with each node, and the type of degrees of freedom.
5. The integration method used to evaluate the stiffness matrix. MSC.Marc contains elements which use full integration and reduced integration.

Truss Elements

MSC.Marc contains 2- and 3-node isoparametric truss elements that can be used in three dimensions. These elements have only displacement degrees of freedom.

Since truss elements have no shear resistance, you must ensure that there are no rigid body modes in the system.

Membrane Elements

MSC.Marc contains 4- and 8-node isoparametric membrane elements that can be used in three dimensions. These elements have only displacement degrees of freedom.

Since membrane elements have no bending resistance, you must ensure that there are no rigid body modes in the system.

Membrane elements are often used in conjunction with beam or truss elements.

Continuum Elements

MSC.Marc contains continuum elements that can be used to model plane stress, plane strain, generalized plane strain, axisymmetric and three-dimensional solids. These elements have only displacement degrees of freedom. As a result, solid elements are not efficient for modeling thin structures dominated by bending. Either beam or shell elements should be used in these cases.

The solid elements that are available in MSC.Marc have either linear or quadratic interpolation functions.

They include

- 4-, 6-, and 8-node plane stress elements
- 3-, 4-, 6-, and 8-node plane strain elements
- 6 (4 plus 2)- and 10 (8 plus 2)-node generalized plane strain elements
- 3-, 4-, 6-, and 8-node axisymmetric ring elements
- 8-, 10-, and 20-node brick elements
- 4- and 10-node tetrahedron
- 12- and 27-node semi-infinite brick elements

In general, the elements in MSC.Marc use a full-integration procedure. Some elements use reduced integration. The lower-order reduced integration elements include an hourglass stabilization procedure to eliminate the singular modes.

Continuum elements are widely used for thermal stress analysis. For each of these elements, there is a corresponding element available for heat transfer analysis in MSC.Marc. As a result, you can use the same mesh for the heat transfer and thermal stress analyses.

MSC.Marc has no singular element for fracture mechanics analysis. However, the simulation of stress singularities can be accomplished by moving the midside nodes of 8-node quadrilateral and 20-node brick elements to quarter-point locations near the crack tip. Many fracture mechanics analyses have used this quarter-point technique successfully.

The 4- and 8-node quadrilateral elements can be degenerated into triangles, and the 8- and 20-node solid brick elements can be degenerated into wedges and tetrahedra by collapsing the appropriate corner and midside nodes. The number of nodes per element is not reduced for degenerated elements. The same node number is used repeatedly for collapsed sides or faces. When degenerating incompressible elements, exercise caution to ensure that a proper number of Lagrange multipliers remain. You are advised to use the higher-order triangular or tetrahedron elements wherever possible, as opposed to using collapsed quadrilaterals and hexahedra.

Beam Elements

MSC.Marc's beam elements are 2- and 3-node, two- and three-dimensional elements that can model straight and curved-beam structures and framed structures and can serve as stiffeners in plate and shell structures. A straight beam of a circular cross section can be used for modeling the straight portion of piping systems. Translational and rotational degrees of freedom are included in beam elements. The cross-section of the beam can be standard rectangular, circular sections, or arbitrarily closed or open sections. The **BEAM SECT** parameter is used to define an arbitrary closed or open cross section.

The beam elements are numerically integrated along their axial direction using Gaussian integration. The stress strain law is integrated through the cross section using Simpson's rule. Stresses and strains are evaluated at each integration point through the thickness. This allows an accurate calculation if nonlinear material behavior is present. In elastic beam elements, only the total axial force and moments are computed at the integration points.

Plate Elements

The linear shell elements (Types [49](#), [72](#), [75](#), [138](#), [139](#), or [140](#)) or the quadratic element (Type [22](#)) can be used effectively to model plates and have the advantage that tying is unnecessary. For element type [49](#), you can indicate on the [GEOMETRY](#) option that the flat plate formulation is to be used. This reduces computational costs. To further reduce computational costs for linear elastic plate analysis, the number of points through the thickness can be reduced to one by use of the [SHELL SECT](#) parameter.

Shell Elements

MSC.Marc contains three isoparametric, doubly curved, thin shell elements: 3-, 4-, and 8-node elements (Types 4, 8, and 24, respectively) based on Koiter-Sanders shell theory. These elements are C^1 continuous and exactly represent rigid-body modes. The program defines a mesh of these elements with respect to a surface curvilinear coordinate system. You can use the **FXORD** model definition option to generate the nodal coordinates. Tying constraints must be used at shell intersections.

Thin shell analysis can be performed using either the 3-node discrete Kirchhoff theory (DKT) based element (Type 138), the 4-node bilinear DKT element (Type 139), the 6-node bilinear semi-loof element (Type 49), or the 8-node bilinear semi-loof element (Type 72). Thick shell analysis can be performed using the 4-node bilinear Mindlin element (Type 75), the 4-node reduced integration with hourglass Mindlin element (Type 140), or the 8-node quadratic Mindlin element (Type 22). The thick shell elements have been developed so that there is no locking when used for thin shell applications.

The global coordinate system defines the nodal degrees of freedom of these elements. These elements are convenient for modeling intersecting shell structures since tying constraints at the shell intersections are not needed.

MSC.Marc contains three axisymmetric shell elements: 2-node straight, 2-node curved, and 3-node curved. You can use these elements to model axisymmetric shells; combined with axisymmetric ring elements, they can be used to simulate the thin and thick portions of the structure. The program provides standard tying constraints for the transition between shell and axisymmetric ring elements.

Heat Transfer Elements

The heat transfer elements in MSC.Marc consist of the following:

- 2- and 3-node three-dimensional links
- 3-, 4-, 6-, and 8-node planar and axisymmetric elements
- 6- and 9-node planar and axisymmetric semi-infinite elements
- 8-, 10-, and 20-node solid elements
- 4- and 10-node tetrahedral
- 12- and 27-node semi-infinite brick elements
- 2-, 3-, 4-, 6-, and 8-node shell elements.

For each heat transfer element, there is at least one corresponding stress element, enabling you to use the same mesh for both the heat transfer and thermal stress analyses. Heat transfer elements are also employed to analyze coupled thermo-electrical (Joule heating) problems.

In heat transfer continuum elements or link elements, temperature is the only nodal degree of freedom.

The heat transfer shell elements can be used in two modes. In the first mode, the number of degrees of freedom is two or three for linear or quadratic temperature distribution across the complete thickness. In the second mode, the number of degrees of freedom are $M + 1$ or $2 * M + 1$ for linear or quadratic temperature distribution for each M layer. This is defined through the [HEAT](#) parameter.

In Joule heating, the voltage and temperature are the nodal degrees of freedom. Shell elements are not available for Joule heating.

Acoustic Analysis

Heat transfer elements are also used to model the compressible media in acoustic analysis. In this case, the pressure is the single nodal degree of freedom.

Electrostatic Analysis

Heat transfer elements are also used for electrostatic analysis. The scalar potential is the degree of freedom.

Fluid/Solid Interaction

Heat transfer elements are used to model the inviscid, incompressible fluid/solid interaction problems. The hydrostatic pressure is the degree of freedom.

Hydrodynamic Bearing Analysis

The three-dimensional heat transfer links and planar elements are used to model the lubricant film. As no variation occurs through the thickness of the film, two-dimensional problems are reduced to one-dimensional, and three-dimensional problems are reduced to two-dimensional. The pressure is the degree of freedom.

Magnetostatic Analysis

For two-dimensional problems, a scalar potential can be used; hence, the heat transfer planar and axisymmetric elements are employed. The single degree of freedom is the potential. For three-dimensional analyses, magnetostatic elements are available. In such cases, there are three degrees of freedom to represent the vector potential.

Electromagnetic Analysis

In electromagnetic problems, a vector potential, augmented with a scalar potential, is used. There are lower-order elements available for these analyses.

Soil Analysis

There are three types of soil/pore pressure analysis. If a pore pressure analysis only is performed, “heat transfer” elements ([41](#), [42](#), or [44](#)) are used. If an uncoupled soil analysis is performed, the standard elements ([21](#), [27](#), or [28](#)) are used. If a coupled analysis is performed, the Herrmann elements ([32](#), [33](#), or [35](#)) are used. In this case, the last degree of freedom is the pore pressure.

Fluid Analysis

When performing fluid analysis, any planar, axisymmetric or solid continuum element can be used. These elements either used a mixed formulation or a penalty formulation based upon the value of the **FLUID** parameter. For the mixed formulation, each node of lower- or higher-order continuum element has velocities as well as pressure degrees of freedom. The penalty method yields elements with only the velocities as the nodal variables.

Piezoelectric Analysis

In a piezoelectric analysis, a strong coupling exist between stress and electric field. There are lower-order elements available for this analysis. The first two or three degrees of freedom (in resp. 2-D or 3-D) are available for displacement components, and the last (3rd or 4th) degree of freedom is available for the electric potential.

Special Elements

MSC.Marc contains unique features in the program.

Gap-and-Friction Elements

The gap-and-friction elements (12, 97) are based on the imposition of gap closure constraint and frictional-stick or frictional-slip through Lagrange multipliers. These elements provide frictional and gapping connection between any two nodes of a structure; they can be used in several variations, depending on the application. In the default formulation, the elements simulate a gap in a fixed direction, such that a body does not penetrate a given flat surface. Using the optional formulation, you can constrain the true distance between two end-points of the gap to be greater than some arbitrary distance. This ability is useful for an analysis in which a body does not penetrate a given two-dimensional circular or three-dimensional spherical surface. Finally, you can update the gap direction and closure distance during analysis for the modeling of sliding along a curved surface.

Pipe-bend Element

The pipe-bend (3-node elbow) element 17 is designed for linear and nonlinear analysis of piping systems. It is a modified axisymmetric shell element for modeling the bends in a piping system. The element has a beam mode superposed on the axisymmetric shell modes so that ovalization of the cross-section is admitted. The twisting of a pipe-bend section is ignored because the beam has no flexibility in torsion. Built-in tying constraints are used extensively for coupling the pipe-bend sections to each other and to straight beam elements.

Curved-pipe Element

The curved-pipe (2-node) element 31 is designed for linear analysis only. The stiffness matrix is based upon the analytic elastic solution of a curved pipe.

Shear Panel Element

The shear panel (4-node) element 68 is an elastic element of arbitrary shape. It is an idealized model of an elastic sheet. This element only provides shearing resistance. It must be used with beam stiffeners to ensure any normal or bending resistance. The shear-panel element is restricted to linear material and small displacement analysis.

Cable Element

The cable element (51) is an element which exactly represents the catenary behavior of a cable. It is an elastic element only.

Rebar Elements

The rebar elements (23, 46, 47, 48, 142, 143, 144, 145, 146, 147, 148, 165, 166, 167, 168, 169, and 170) are hollow elements in which you can place single-direction strain members (reinforcing cords or rods). The rebar elements are used in conjunction with other solid elements (filler) to represent a reinforced material such as reinforced concrete. The reinforcing members and the filler are accurately represented by embedding rebar elements into solid elements. The rebar elements can be used for small as well as large strain behavior of the reinforcing cords. Also, any kind of material behavior can be simulated by the rebar elements in MSC.Marc.

Incompressible Elements

Incompressible and nearly incompressible materials can be modeled by using a special group of elements in the program. These elements, based on modified Herrmann variational principles, are capable of handling large deformation effects as well as creep and thermal strains. The incompressibility constraint is imposed by using Lagrange multipliers. Generally, the low (linear) order elements have a single additional node which contains the Lagrange multiplier, while the high-order (quadratic) elements have Lagrange multipliers at each corner node.

Elements 155 (plane strain triangle), 156 (axisymmetric triangle), and 157 (3-D tetrahedron) are low-order elements and are exceptions within the incompressible element group. They have an additional node located at the center of the elements and have Lagrange multipliers at each corner node. The shape function of the center node is a bubble function. See Figure 10-1 for an element 155 case. The degrees of freedom of the center node is condensed out on the element level before the assembly of the global matrix.

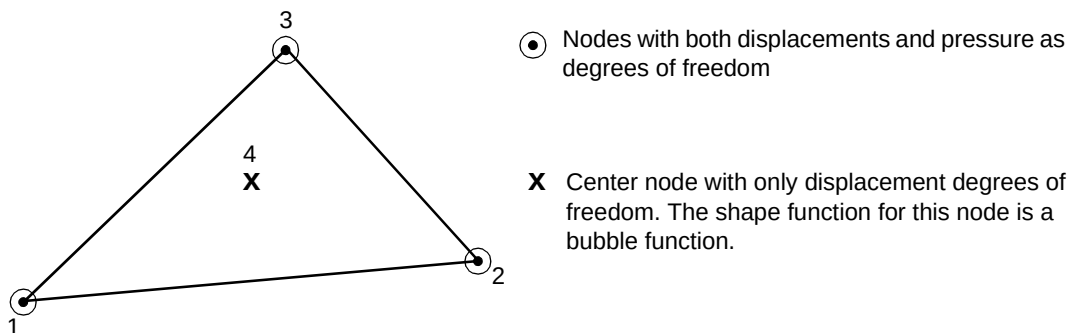


Figure 10-1 Element 155

Large Strain Elasticity

The incompressible elements based on Herrmann formulations can be used for large strain analysis of rubber-like materials, using either total Lagrangian formulation or updated Lagrangian formulation.

Large Strain Plasticity

The incompressible elements can be used for large strain analysis of elasto-plastic materials using updated Lagrangian formulation, based on multiplicative decomposition of deformation gradient.

Rigid-Plastic Flow

In rigid-plastic flow analysis, with effectively no elasticity, incompressible elements must be used. For such analyses, you have a choice of using either the elements with Lagrange multipliers discussed above, or standard displacement continuum elements. In the latter case, a penalty function is used to satisfy the incompressibility constraint.

Constant Dilatation Elements

You can choose an integration scheme option, which makes the dilatational strain constant throughout the element. This can be accomplished using the [CONSTANT DILATATION](#) parameter or by setting the second field of the [GEOMETRY](#) option to one. Constant dilatational elements are recommended for use in approximately incompressible, inelastic analysis, such as large strain plasticity, because conventional elements can produce volumetric locking due to overconstraints for nearly incompressible behavior. This option is only available for elements of lower order (Types [7](#), [10](#), [11](#), [19](#), and [20](#)). For the lower-order reduced integration elements ([114](#) to [117](#)) with hourglass control, as only one integration point is used, these elements do not lock, and effectively, a constant dilatation formulation is used. (The constant dilatation approach has also been referred to mean dilatation and B-bar approach in literature.)

Reduced Integration Elements

MSC.Marc uses a reduced integration scheme to evaluate the stiffness matrix or the thermal conductivity matrix for a number of isoparametric elements. The mass matrix and the specific heat matrix of the element are always fully integrated.

For lower-order, 4-node quadrilateral elements, the number of numerical integration points is reduced from 4 to 1; in 8-node solid elements, the number of numerical integration points is reduced from 8 to 1. For 8-node quadrilateral elements, the number of numerical integration points is reduced from 9 to 4; for 20-node solid elements, the number of numerical integration points is reduced from 27 to 8.

The energy due to the higher-order deformation mode(s) associated with high-order elements is not included in the analysis. Reduced integration elements and fully integrated elements can be used together in an analysis. Reduced integration elements are more economical than fully integrated elements and they can improve analysis accuracy. However, with near singularities and in regions of high-strain gradients, the use of reduced integration elements can lead to oscillations in the displacements and produce inaccurate results. Using reduced integration elements results in zero-energy modes, or breathing nodes. In the lower-order elements, an additional stabilization stiffness is added which eliminates these so-called “hourglass” modes.

Continuum Composite Elements

There is a group of isoparametric elements in MSC.Marc which can be used to model composite materials. Different material properties can be used for different layers within these elements. The number of continuum layers within an element, the thickness of each layer, and the material identification number for each layer are input via the [COMPOSITE](#) option. A maximum number of 2040 layers can be used in one element.

These structural elements are available as both lower- and higher-order and can be used for plane strain (element types [151](#) and [153](#)), axisymmetric (element types [152](#) and [154](#)) or 3-D solid analysis (element types [149](#) and [150](#)). Corresponding heat transfer elements are available for lower- and higher-order and can be used for planar (element types [177](#) and [179](#)), axisymmetric (element types [178](#) and [180](#)) or 3-D solid analysis (element types [175](#) and [176](#)).

Fourier Elements

A special class of elements exists which allows the analysis of axisymmetric structures with nonaxisymmetric loads. The geometry and material properties of these structures do not change in the circumferential direction, and the displacement can be represented by a Fourier series. This representation allows a three-dimensional problem to be decoupled into a series of two-dimensional problems. Both solid and shell Fourier elements exist in MSC.Marc. These elements can only be used for linear elastic analyses.

Semi-infinite Elements

This group of elements can be used to model unbounded domains. In the semi-infinite direction, the interpolation functions are exponential, such that the function (displacement) is zero at the far domain. The rate of decay of the function is dependent upon the location of the midside nodes. The interpolation function in the non semi-infinite directions are either linear or quadratic.

These elements can be used for static plane strain, axisymmetric, or 3-D solid analysis. They can also be used for heat transfer, electrostatic, and magnetostatic analyses.

Cavity Surface Elements

This group of elements can be used to define the boundaries of cavities where standard finite elements are not used; for example, along rigid boundaries. These elements are for volume calculation purposes only and do not contribute to the stiffness equations of the model. They can be used for plane strain, plane stress, axisymmetric, and 3-D analyses. No material properties are needed for these elements. They do not undergo any deformation except if they are attached or glued to other elements or surfaces.

Assumed Strain Formulation

Conventional isoparametric four-node plane stress and plane strain, and eight-node brick elements behave poorly in bending. The reason is that these elements do not capture a linear variation in shear strain which is present in bending when a single element is used in the bending direction. For the six elements (3, 7, 11, 160, 161, and 163), the interpolation functions have been modified such that shear strain variation can be better represented. For the lower-order reduced integration elements (114 to 123), an assumed strain formulation written with respect to the natural coordinates is used. For elastic isotropic bending problems, this allows the exact displacements to be obtained with only a single element through the thickness. This is invoked by using the **ASSUMED STRAIN** parameter or by setting the third field of the **GEOMETRY** option to one.

Follow Force Stiffness Contribution

When activating the **FOLLOW FOR** parameter, the distributed loads are calculated based upon the current deformed configuration. It is possible to activate an additional contribution which goes into the stiffness matrix. This improves the convergence. This capability is available for element types [3](#), [7](#), [10](#), [11](#), [18](#), [72](#), [75](#), [80](#), [81](#), [82](#), [83](#), [84](#), [114](#), [115](#), [116](#), [117](#), [118](#), [119](#), [120](#), [139](#), [140](#), [149](#), [151](#), [152](#), [160](#), [161](#), [162](#), or [163](#). Inclusion of the follower force stiffness can result in nonsymmetric stiffness for nonenclosed volumes, thereby, resulting in increased computational times. You can flag the nonsymmetric solver in the **SOLVER** option.

Explicit Dynamics

The explicit dynamics formulation IDYN=5 model is restricted to the following elements:

1, 2, 3, 5, 6, 7, 9, 10, 11, 18, 19, 20, 52, 64, 75, 89, 98, 114, 115, 116, 117, 118, 119, 120, 130, 139, and 140.

When using this formulation, the mass matrix is defined semi-analytically; for example, no numerical integration is performed. In addition, a quick method is used to calculate the Courant stability limit associated with each element. For these reasons, this capability is limited to the elements mentioned above.

Adaptive Mesh Refinement

MSC.Marc has a capability to perform local adaptive mesh refinement to improve the accuracy of the solution. This capability is invoked by using the ADAPTIVE [parameter](#) and [model definition](#) option. The adaptive meshing is available for the following 2-D and 3-D elements:

2, 3, 6, 7, 10, 11, 18, 19, 20, 37, 38, 39, 40, 43, 75, 80, 81, 82, 83, 84, 111, 112, 113, 114, 115, 116, 117, 118, 119, 120, 121, 122, 123, 138, 139, 140, 155, 156, 157, 160, 161, 162, 163, and 164.

References

1. Herrmann, L.R. "Elasticity Equations for Incompressible and Nearly Incompressible Materials by a Variational Theorem." *AIAA Journal*, 3 (10): 1896-2000, 1965.
2. Nagtegaal, J.C., Parks, D.M., and Rice, J.R. "On Numerically Accurate Finite Element Solutions in the Fully Plastic Range." *Computer Methods in Applied Mechanics and Engineering*, 4, 153-178, 1974.
3. Zienkiewicz, O.C. *The Finite Element Method*. 3rd ed., London: McGraw-Hill, 1977.

CHAPTER
11

Solution Procedures for Nonlinear Systems

- Considerations for Nonlinear Analysis
- Full Newton-Raphson Algorithm
- Modified Newton-Raphson Algorithm
- Strain Correction Method
- The Secant Method
- Direct Substitution
- Arc-length Methods
- Remarks
- Convergence Controls
- Singularity Ratio
- Solution of Linear Equations

■ [Flow Diagram](#)

■ [References](#)

This chapter discusses the solution schemes in MSC.Marc for nonlinear problems. Issues of convergence controls, singularity ratio, and available solvers for linearized system of equations are also discussed. In a nonlinear problem, a set of equations must be solved incrementally. The governing equation of the linearized system can be expressed, in an incremental form, as

$$K\delta u = r \quad (11-1)$$

where δu and r are the correction to the incremental displacements and residual force vectors, respectively.

There are several solution procedures available in MSC.Marc for the solution of nonlinear equations:

- [Full Newton-Raphson Algorithm](#)
- [Modified Newton-Raphson Algorithm](#)
- [Strain Correction Method](#)
- [The Secant Method](#)
- [Direct Substitution](#)
- [Arc-length Methods](#)

Considerations for Nonlinear Analysis

Nonlinear analysis is usually more complex and expensive than linear analysis. Also, a nonlinear problem can never be formulated as a set of linear equations. In general, the solutions of nonlinear problems always require incremental solution schemes and sometimes require iterations (or recycles) within each load/time increment to ensure that equilibrium is satisfied at the end of each step. Superposition cannot be applied in nonlinear problems.

The four iterative procedures available in MSC.Marc are: Newton-Raphson, Modified Newton-Raphson, Newton-Raphson with strain correction modification, and a secant procedure. If the R-P flow contribution model is chosen, a direction substitution is used. See the appendix for a discussion of these iterative procedures.

A nonlinear problem does not always have a unique solution. Sometimes a nonlinear problem does not have any solution, although the problem can seem to be defined correctly.

Nonlinear analysis requires good judgment and uses considerable computing time. Several runs are often required. The first run should extract the maximum information with the minimum amount of computing time. Some design considerations for a preliminary analysis are:

- Minimize degrees of freedom whenever possible.
- Halve the number of load increments by doubling the size of each load increment.
- Impose a coarse tolerance on convergence to reduce the number of iterations. A coarse run determines the area of most rapid change where additional load increments might be required. Plan the increment size in the final run by the following rule of thumb: there should be as many load increments as required to fit the nonlinear results by the same number of straight lines.

MSC.Marc solves nonlinear static problems according to one of the following two methods: tangent modulus or initial strain. Examples of the tangent modulus method are elastic-plastic analysis, nonlinear springs, nonlinear foundations, large displacement analysis and gaps. This method requires at least the following three controls:

- A tolerance on convergence
- A limit to the maximum allowable number of recycles
- Specification of a minimum number of recycles

An example of the initial strain method is creep or viscoelastic analysis. Creep analysis requires the following tolerance controls:

- Maximum relative creep strain increment control
- Maximum relative stress change control
- A limit to the maximum allowable number of recycles

To input control tolerances, use the model definition option **CONTROL**. These values can be reset upon restart or through the **CONTROL** history definition option. See **Convergence Controls** in this chapter for further discussion on tolerance controls.

Behavior of Nonlinear Materials

Nonlinear behavior can be time- (rate-) independent, or time- (rate-) dependent. For example, plasticity is time-independent and creep is time-dependent. Both viscoelastic and viscoplastic materials are also time-dependent. Nonlinear constitutive relations must be modeled correctly to analyze nonlinear material problems. A comprehensive discussion of constitutive relations is given in **Chapter 7**.

Scaling the Elastic Solution

The parameter **SCALE** causes scaling of the linear-elastic solution to reach the yield stress in the highest stressed element. Scaling takes place for small displacement elastic-plastic analysis, where element properties do not depend on temperature. The **SCALE** parameter causes all aspects of the initial solution to be scaled, including displacements, strains, stresses, temperature changes, and loads. Subsequent incrementation is then based on the scaled solution.

Load Incrementation

Several history definition options are available in MSC.Marc to input mechanical and thermal load increments (see **Table 11-1**). The choice is between a fixed and an automatic load stepping scheme.

For a fixed scheme, the load step size remains constant during a load case. The fixed schemes are **AUTO LOAD** for static mechanical, **CREEP INCREMENT** for creep, **DYNAMIC CHANGE** for dynamic mechanical, and **TRANSIENT NON AUTO** for thermal/thermo-mechanically coupled.

For an adaptive scheme, the load step size changes from one increment to the other and also within an increment depending on convergence criteria and/or user-defined physical criteria. The adaptive schemes are **AUTO STEP** and **AUTO INCREMENT** for static mechanical, **AUTO STEP** and **AUTO CREEP** for creep, **AUTO STEP**, **AUTO**

THERM, and **AUTO THERM CREEP** for thermally driven mechanical problems, **AUTO STEP** and **TRANSIENT** for dynamic mechanical, and **AUTO STEP** and **TRANSIENT** for thermal/thermo-mechanically coupled.

The adaptive stepping scheme of choice is **AUTO STEP**. **AUTO STEP** has been designed as a unified load stepping scheme and many of the capabilities of the other stepping schemes can now be handled by **AUTO STEP**. **AUTO STEP** can be used for mechanical (static, creep, dynamic), thermal, and thermo-mechanically coupled analysis problems.

- In the **AUTO STEP** scheme, a recycle based convergence criterion is used to automatically determine the time step based upon a comparison of the actual number of recycles needed in an increment to a user-specified desired number of recycles. In addition to this recycle based criterion, user-defined or program-determined physical criteria based upon strain, stress, displacement, or temperature increments can be used to control the time step. Reductions in the time step through cut-backs are used to satisfy both convergence criteria and physical criteria.

More details on the **AUTO STEP** option are provided in the next section. While the default adaptive stepping procedure is **AUTO STEP**, other available adaptive time-stepping options that have been previously designed for particular loading situations can also be used. Such analysis types are listed below. It should be noted that **AUTO STEP** can now be used for many of these situations, with just the default convergence based criterion or/and by the addition of suitable physical criteria.

- Post buckling or snap-through analyses require the so-called arclength method which is available through the **AUTO INCREMENT** option. This option can only be used in static mechanical analyses and the applied load is automatically increased or decreased in order to maintain a certain arclength.

Note: **AUTO INCREMENT** can also be used for general situations without instabilities, but, in general, the **AUTO STEP** option is preferred for these situations.

- For creep analysis, the available adaptive options are **AUTO CREEP** and **AUTO STEP**. The **AUTO CREEP** option determines the time step in an explicit creep analysis based on the creep strain change and the stress change (see Volume C: Program Input, Chapter 3). These checks are not available by default in **AUTO STEP**. These may however be incorporated by using an absolute or relative creep strain increment criterion and an absolute or relative stress increment criterion as additional user-defined physical criteria in combination with the default convergence criterion. Addition of these user-defined criteria for **AUTO STEP** is quite simple. If an appropriate input flag is set, two physical criteria are automatically added by the program at run-time for explicit creep problems:

creep strain increment/elastic strain = 0.5, and stress increment/stress at beginning of increment = 0.5. It should also be noted that **AUTO STEP** is usually more reliable in cases involving creep and contact.

- For the case of automatic load stepping for a thermally loaded elastic-creep/elastic-plastic-creep stress analysis, the available adaptive schemes are **AUTO THERM CREEP** and **AUTO STEP**. Allowable increments for normalized creep strain, normalized stress and state variables can be optionally prescribed for **AUTO STEP** either through user-defined criteria or program determined automatic physical criteria.
- For thermally driven mechanical problems, the available options are **AUTO THERM** and **AUTO STEP**. The thermal loads derived from a thermal analysis are applied using the **CHANGE STATE** option in a mechanical analysis. In the **AUTO THERM** scheme, the load step of the mechanical analysis is automatically adjusted based upon user-specified (allowed) changes in temperature from the thermal analysis per increment. For example, if there is a change of 50° in the thermal analysis in one increment but only a change of 10° per increment is allowed in the mechanical analysis, **AUTO THERM** splits up the thermal increment into five mechanical increments. Allowable state variable increments can be optionally prescribed for **AUTO STEP** either through a user-defined criterion or program determined automatic physical criterion. If these criteria are violated in an increment, **AUTO STEP** cuts the time-step back and repeats the increment with a smaller time step.

Table 11-1 History Definition Options for Load Incrementation

Load Type	Fixed	Adaptive
Mechanical	AUTO LOAD** DYNAMIC CHANGE** CREEP INCREMENT**	AUTO STEP AUTO INCREMENT AUTO TIME* AUTO THERM AUTO CREEP AUTO THERM CREEP*
Thermal	TRANSIENT NON AUTO**	AUTO STEP TRANSIENT*

* The option is obsolete; use **AUTO STEP** instead.

** These fixed stepping schemes are useful for complex loading histories that vary with time.

Selecting Load Increment Size

Selecting a proper load step increment is an important aspect of a nonlinear solution scheme. Large steps often lead to many recycles per increment and, if the step is too large, it can lead to inaccuracies and nonconvergence. On the other hand, using too small steps is inefficient.

Fixed Load Incrementation

When a fixed load stepping scheme is used, it is important to select an appropriate load step size that captures the loading history and allows for convergence within a reasonable number of recycles. For complex load histories, it is necessary to prescribe the loading through time tables while setting up the run.

For fixed stepping, there is an option to have the load step automatically cut back in case of failure to obtain convergence. When an increment diverges, the intermediate deformations after each recycle can show large fluctuations and the final cause of program exit can be any of the following: maximum number of recycles reached (exit 3002), elements going inside out (exit 1005 or 1009) or, in a contact analysis, nodes sliding off a rigid contact body (exit 2400), and nodes not being projected properly onto 3-D NURBS (exit 2401). These deformations are normally not visible as post results (there is a feature to allow for the intermediate results to be available on the post file, see the [POST](#) option). If the cutback feature is activated and one of these numbers occur, the state of the analysis at the end of the previous increment is restored from a copy kept in memory, and the increment is subdivided into a number of subincrements. The step size is halved until convergence is obtained or the user-specified number of cutbacks has been performed. Once a subincrement is converged, the analysis continues to complete the rest of the original increment. No results are written to the post file during subincrementation. When the original increment is finished, the calculation continues to the next increment with the original increment count maintained. If the global remeshing option is activated in conjunction with the cutback feature, then, for exit 1005 or 1009, the chosen contact body is remeshed and the analysis is repeated with the original time step before the first cutback.

Automatic Load Incrementation

In many nonlinear analyses, it is useful to have MSC.Marc figure out the appropriate load step size automatically. The [AUTO INCREMENT](#) option is a so-called arc-length method and is designed for applications like post buckling and snap-through analysis. This method is described in detail in [Arc-length Methods](#) in this chapter.

The time step to be used for an increment is adaptively assessed in automatic load incrementation methods. The assessment of the corresponding load step for the increment however varies, depending on whether the loading is prescribed in conjunction with the **TABLE** model definition option or not. The **TABLE** option for boundary conditions is optionally supported in the current version.

- When the **TABLE** option is not used for the loading (this is the default), then mechanical loads are applied in a proportional manner and thermal loads are applied instantaneously. This means that any automatic load incrementation method is limited to mechanical input histories that only have linear variations in load or displacement and thermal input histories that have immediate ramps in flux or temperature. Furthermore, all of the automatic load incrementation methods only require the prescription of the total load or displacement at the end of the load case. For example, one may not use a rigid body with a linearly changing velocity, since the resulting displacement of the rigid body would give parabolically changing displacements. In this case, one would need to use a constant velocity for the automatic load incrementation methods to work properly.

For any automatic load incrementation, care must be taken to appropriately define the loading history in each loadcase. The load case should be defined between appropriate break points in the load history curve. For example, in **Figure 11-1**, correct results would be obtained upon defining three distinct loadcases between times $0 - t_1$, $t_1 - t_2$, and $t_2 - t_3$ during the model preparation. However, if only one load case is defined for the entire load history between $0 - t_3$, the total applied load for the loadcase is zero.

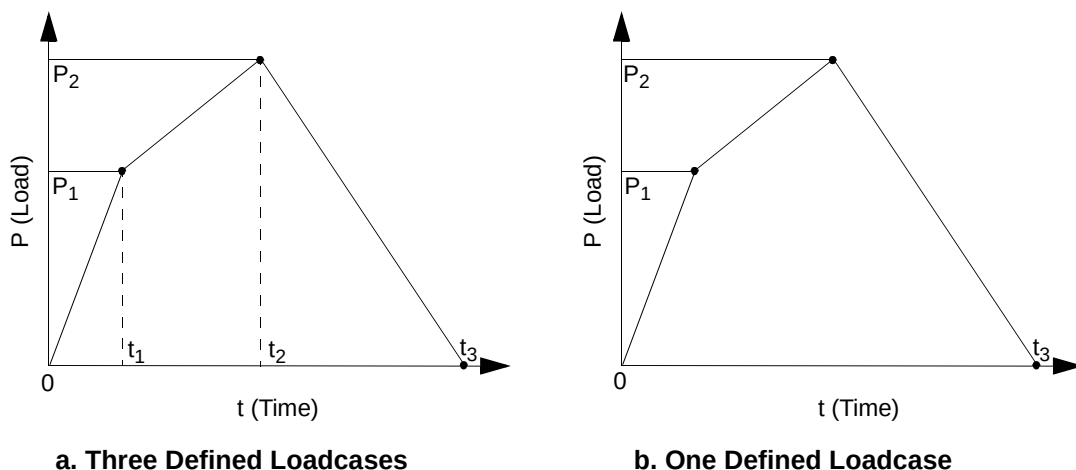


Figure 11-1 Defining Loadcases for Automatic Load Incrementation

- When the **TABLE** option is used to describe the time-wise variation of the loading (optional in current version), the magnitude of the load is determined from the and applied. This means that for the case described in **Figure 11-1**, one loadcase can now be used to prescribe the load. Furthermore, depending on whether time t is between $0 - t_1$, $t_1 - t_2$, or $t_2 - t_3$, the load is ramped up or ramped down. It should be noted that when the input deck is written out using the format, then all of the load associated with that boundary condition is applied instantaneously. Furthermore, use of time tables to prescribe loads is not allowed for **AUTO INCREMENT** since the latter scheme controls the arc length of the response and the algorithm should be capable of either increasing or decreasing the load automatically.

AUTO STEP

The scheme appropriate for most applications is **AUTO STEP**. The primary control of the load step is based upon the number of recycles needed to obtain convergence. There are a number of optional user-specified physical criteria that can be used to additionally control the load step. The user inputs needed to define the **AUTO STEP** scheme are described in *MSC.Marc Volume C: Program Input*.

Recycling Criterion

The default recycle based criterion works as follows: The user specifies a desired number of recycles (set to 3 by default). For problems with severe nonlinearities or for problems with very small convergence tolerances, it may be necessary to increase this number. This number is used as a target value for the load stepping scheme. If the

number of recycles required in the current increment is less than the desired number, the load step for the next increment is increased. The factor with which the time step is increased, S_u , defaults to 1.2 and can be specified by the user.

Time Step Cutback Scheme

The load step is never increased during an increment. If the number of recycles needed to obtain convergence exceeds the desired number, the load step size is scaled back, the recycling cutback number N_r is incremented by 1 and the increment is performed again with the new load step. The scaleback factor for the N_r^{th} cutback is taken as

s^{N_r} , where the factor s is calculated from the expression

$s = \langle T_s/T_m \rangle^{2/(N_{rm}(N_{rm} + 1))}$; where N_{rm} is the maximum number of recycling related cutbacks for the increment and is calculated from

$N_{rm} = \log_{10}(10^{-5} * T_s/T_m)$, T_s is the time increment before any recycling related cutbacks occur for the increment and T_m is the minimum possible time step for the increment. T_m is equal to the value set by the user (10^{-5} by default) if there is no

quasi-static inertial damping and is equal to 10^{-3} times the value set by the user (10^{-8} by default) if there is quasi-static inertial damping. The scaleback factor for any

cutback is the smaller of (s^{N_r} , $1/S_u$). This scheme guarantees that no matter what the starting time step for an increment, the minimum time step is reached in a reasonable number of cutbacks if the increment consistently fails to converge. For special cutbacks such as maximum number of recycles reached (exit 3002), elements going inside out (exit 1005 or 1009) or, in a contact analysis, nodes sliding off a rigid contact body (exit 2400) nodes not being projected properly onto a 3-D NURB (exit 2401),

the scaleback factor is the smaller of (s^{N_r} , 0.5). If the minimum time step is reached and the analysis still fails to converge, it is terminated with exit 3015. If the 'proceed when not converged' option is used, then the analysis proceeds to the next increment if and when the maximum number of recycles are reached.

Exceptions

There are some exceptions to the basic scheme outlined above. If an increment is consistently converging with the current load step and the number of recycles exceeds the desired number, the number of recycles is allowed to go beyond the desired number until convergence or up to the user specified maximum number. The time step is then decreased for the next increment by $1/S_u$. An increment is determined to be converging if the convergence ratio was decreasing in three previous recycles.

Special rules also apply in a contact analysis. For quasi-static problems, the **AUTO STEP** option is designed to only use the automated penetration check option (see **CONTACT** option, 7th field of 2nd data block; option 3 is always used). Even if you flag the increment splitting penetration check option wherein penetration is checked only at the end of an otherwise converged increment, MSC.Marc internally converts it to the automated penetration check wherein penetration is checked during every iteration. More details on the two penetration schemes are provided in Chapter 8. During the recycles, the contact status can keep changing (new nodes come in contact, nodes slide to new segments, separate etc.). Whenever the contact status changes during an increment, a new set of contact constraints are incorporated into the equilibrium equations and more recycles are necessary in order to find equilibrium. These extra recycles, due to contact changes, are not counted when the recycle number is checked against the desired number for determining if the load step needs to be decreased within the increment. Thus, only true Newton-Raphson iterations are taken into account. For the load step of the next increment, the accumulated number of recycles during the previous increment is used. This ensures that the time step is not increased when there are many changes in contact during the previous increment.

Thermal Analysis

For the most part, the recycling criterion works in a similar fashion for thermal analysis (heat transfer analysis, or thermal part of a coupled thermo-mechanical analysis). The recycling criterion is used to satisfy the tolerance value provided for the temperature error in estimate on the **CONTROL** option; that is, the analysis cuts back and starts the increment over if the temperature error in estimate is not consistently converging within the desired number of recycles. If the temperature estimate consistently converges in three previous recycles, the analysis continues recycling. Once the temperature estimate tolerance is satisfied, the actual incremental temperature change ΔT_a is calculated and checked against the corresponding tolerance ΔT_m provided on the **CONTROL** option. It should be noted that if a user criterion on temperature is also provided, the latter parameters over-ride the one provided on the **CONTROL** option. More details on how user criteria are handled are described in the next section. If the incremental temperature change is not satisfied, the time step is scaled-back using a factor = $0.8((\Delta T_m)/(\Delta T_a))$. In this case, the thermal pass is simply continued with the smaller time step without a formal cut-back because if there was a formal cutback, the analysis would have redo the temperature error in estimate convergence from scratch. However, if the maximum number of allowed recycles is reached before thermal convergence is achieved, a formal cutback with a scale-back factor of $\min(s^{Nr}, 0.5)$ is made and the increment is repeated.

User-defined Physical Criteria

In addition to allowing MSC.Marc to use the number of recycles for automatically controlling the step size for **AUTO STEP**, user-specified physical criteria can optionally be used for controlling the step size. The user-specified physical criteria work as follows. The user can specify the maximum allowed incremental change within certain ranges for specific quantities during an increment. The quantities available are displacements, rotations, stresses, strains, strain energy, temperature (in thermal or thermomechanically coupled analyses), and state variables (in mechanical or thermomechanically coupled analysis). These criteria can be utilized in one of two ways.

By default, they are used as limits, which means that the load step is immediately decreased if a criterion is violated during any iteration of the current increment, but they do not influence the decision to change the load step for the next increment; that is, only the actual number of recycles versus desired number of recycles controls the load step for the next increment.

The criteria can also be used as targets; in which case, they are used as the main means for controlling the time step for the current and next increments. If the calculated values of the criteria are higher than the user-specified values in any iteration, the time step is scaled down and the current increment is repeated. If the calculated values of the criteria for the current increment are consistently smaller than the user-specified values prior to convergence, the time step for the next increment is scaled up. The scale factor used for reduction or increase is the ratio between the actual value and the target value and this factor is limited by user-specified minimum and maximum factors (defaults to 0.1 and 10 respectively). If this type of load step control is used together with the recycle based control, the time step can be reduced in the current increment due to whichever criterion that is violated. The decision to increase the step size for the next increment is solely based upon the physical criteria.

Specification of user-defined physical criteria can be further simplified by setting a special flag in the **AUTO STEP** option that allows for physical criteria to be automatically added by the solver at run-time. These automatic criteria serve as upper-bound controls to prevent run-away Newton-Raphson iterations that ultimately cause the program to abort. Currently, four mechanical criteria are automatically added depending on the kind of analysis that is being run: a total strain criterion is added for any large displacement analysis and the maximum allowable equivalent total strain increment at any point in the model is set to 50%; a plastic strain criterion is added for any large displacement, finite strain analysis and the maximum allowable equivalent plastic strain increment at any point in the model is set to 10%; a relative creep strain criterion and a relative stress change criterion are added for any explicit creep analysis wherein the maximum allowable creep strain change/elastic strain and

the maximum allowable equivalent stress change/equivalent stress are each set to 0.5; a state variable criterion is added for any large displacement analysis wherein the maximum allowable temperature increment is such that the equivalent stress increment associated with the change in thermal properties of the materials does not exceed 50% of the total equivalent stress. These criteria are only added in the analysis if there are no competing explicitly defined user-criteria found. It should also be noted that these automatic criteria are only used as limits; that is, they are used to control the time step within an increment but not for the next increment. Finally, for [AXITO3D](#) and automatic rezoning problems, no automatic physical criteria are added.

Failure to satisfy user-defined physical criteria can occur due to two reasons — the maximum number of cutbacks allowed by the user can be exceeded, or the minimum time step can be reached. In this case, the analysis terminates with exit 3002 and exit 3015, respectively. These premature terminations can be avoided by using the option to continue the analysis even if physical criteria are not satisfied. If this flag is set on the [AUTO STEP](#) option, and either the maximum number of user-allowed cutbacks or the minimum time step is reached, a mechanical analysis moves on to the next increment if it is otherwise converged (see [Convergence Controls](#) in this chapter) or continues to recycle and scales back based solely on the recycling criterion. Setting this flag for a thermal analysis simply allows it to move on to the next increment.

Other AUTO STEP Options

In many analyses it is convenient to obtain post file results at specified time intervals. This is naturally obtained with a fixed load stepping scheme but not with an automatic scheme. Traditionally, the post output frequency is given as every n^{th} increment. With the auto step procedure, you can request post output to be obtained at equally spaced time intervals. In this case, the time step is temporarily modified to exactly reach the time for output. The time step is then restored in the following increment.

When tables are used in the new input format to specify loads with complex time variations, in most cases, it is important that the exact peaks and valleys of the loading history are not missed due to the adaptive time stepping. By default, [AUTO STEP](#) adjusts the time step temporarily so that the peaks and valleys of the loading history are reached exactly. The time step is then restored in the following increment.

In addition to controlling the time step through the recycle based criterion and the physical criteria, direct control of the time step is possible through the use of the user subroutine [UTIMESTEP](#). In this case, the new time step that is determined by the auto step algorithm enters the program as input and the modified time step by the user is returned as output. More details are provided in Volume D: User Subroutines and Special Routines, Chapter 2.

The **AUTO STEP** option also has an optional artificial damping feature available for mechanical statics analyses. If the time step goes below the user-specified minimum time step, t_{min} , MSC.Marc normally stops with exit number 3015; but if the artificial damping feature is activated, the analysis continues with a smaller time step (in this case, the smallest time step in the program is set to 0.001 times the user-specified minimum time step). For time steps smaller than t_{min} , the solution is stabilized by adding a factored lumped mass matrix to the stiffness matrix and modifying the force vector consistently. The factor associated with the lumped mass matrix is calculated using the following procedure. At the end of the first stable increment of the loadcase (that is, convergence in that increment being reached before t_{min} is reached), the damping factor c is calculated from $c = \alpha \Delta t_1 (\Delta SE_1) / (\Delta u_1^T M \Delta u_1)$ where α is 10^{-5} by default; Δt_1 is the time step for the first increment at convergence; ΔSE_1 is the incremental strain energy; Δu_1 is the nodal displacement vector; M is the lumped mass matrix calculated with unit mass density. For subsequent increments in the loadcase where stabilization is required, this factor c is used to scale the lumped mass matrix calculated with unit density times the nodal velocity vector. If stabilization is needed prior to the first stable increment (for example, in the very first increment of the loadcase), the damping factor c is calculated from iterations that are stable (that is, iterations in which the time step is larger than t_{min}). The damping factor is further scaled by a nonlinear factor that increases from 0 at t_{min} to a value of about 10^6 at $0.001 t_{min}$. Any form of artificial stabilization is turned off once the time step increases above t_{min} . If the feature is used, it might be useful to write post file results at fixed time intervals; otherwise, many increments might appear on the post file even for a small time period. The default scheme provided for calculating c should generally provide for good energy balance in all increments excepting those undergoing excessive instabilities. If energy balance is not being maintained due to application of excessive damping, it could be controlled by reducing α but you run the risk that the amount of damping may not be sufficient to let unstable increments converge.

The **AUTO STEP** algorithm is further modified for transient dynamics problems:

- When penetration is detected in dynamic contact problems, instead of using the default iterative penetration procedure, a time cutback is made and the increment is repeated with a smaller time step that avoids the penetration. This scheme allows for momentum conservation without spurious penetration induced oscillations in the response. If no other cutbacks are made, the time step is restored in the following increment.
- Additional checks are made in transient dynamic problems to control the time step since larger time steps that may have been assessed based on the recycle based criterion or physical criteria can give rise to unacceptable time integration errors. An additional check is made at the end of each increment to see if the time step needs to be reduced for the following increment. The scheme that is followed is a modified version of the scheme outlined by Bergan and Mollestad [Ref. 1]. The time step for increment $n + 1$ should satisfy the inequality

$$(\Delta t)_{n+1} \leq 2\pi\alpha \frac{\sqrt{v_n^T M v_n}}{\sqrt{v_n^T K_T v_n}}$$

where α is 0.075 by default; v_n is the velocity vector at the end of the increment n ; M is the mass matrix; K_T is the tangent stiffness matrix. The time step is only reduced if the value predicted by the above equation is less than 67% of the current time step. This check is only made for the Newmark-beta and the Single Step Houbolt methods. The check is bypassed if t_{n+1} is already at t_{min} , if the strain energy is negligible (for example, rigid body motion), or if the user has flagged the ‘use physical criteria as targets’ flag. In the last case, it is assumed that the user only wants the physical criteria to determine the time step for the next increment.

The defaults of the **AUTO STEP** option are carefully chosen to be adequate in a wide variety of applications. There are cases, however, when the settings may need to be modified. Assume that the default settings are used, which means that the recycle based control is active with an initial load of one per cent of the total. If the structure is weakly nonlinear, convergence is obtained in just a few recycles and the time steps for successive increments get progressively larger. This can lead to problems if the initially weakly nonlinear structure suddenly exhibits stronger nonlinearities; for instance, occurrence of plasticity or parts coming into contact. Possible remedies to this problem include:

- (i) decrease the time step scale factor from 1.2 to a smaller number so the step size does not grow so rapidly;

- (ii) use a physical criterion like maximum increment of displacements to limit the load step;
- (iii) use the maximum time step to limit large steps;
- (iv) decrease the desired and maximum number of recycles to make the scheme more prone to decrease the load step if more recycles are needed.

Another situation is if the structure is highly nonlinear and convergence is slow. In this case, it may be necessary to increase the desired number and maximum number of recycles. In general, there is a close connection between the convergence tolerances used and the desired number and maximum number of recycles. In many cases, it may be beneficial to use one or more physical criteria; for example, the increment of plastic strain as targets for controlling the load step. This can easily be achieved by allowing the program to add automatic physical criteria where appropriate. This is especially a good idea if the 'proceed if not converged' option is used or if the 'non-positive definite flag' is set since the added physical criteria then serve as controls to limit the time step and produce a realistic numerical solution in each increment rather than letting the solution proceed unchecked with unrealistic results.

Residual Load Correction

The residual load is applied as a correcting force to ensure that equilibrium is maintained and, hence, that an accurate solution is obtained for nonlinear problems.

The residual load correction enforces global equilibrium at the start of each new increment. This prevents the accumulation of out-of-equilibrium forces from increment to increment and makes the solution less sensitive to the step size.

Figure 11-2 shows how stiffness is based on the state at the start of a step. The variables are defined below for increments $i = 1, 2, 3$:

- F_i applied forces for $i = 1, 2, 3$
- u_i calculated displacements for $i = 1, 2, 3$
- R_i residual loads for $i = 1, 2, 3$

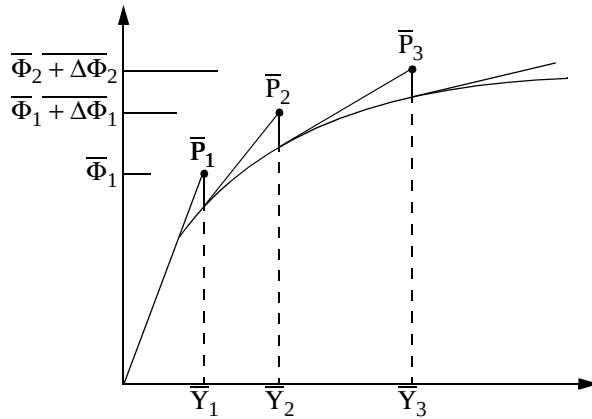


Figure 11-2 Stiffness Based on State at Start of Step

The residual load correction is the difference between the internal forces and the externally applied loads. The residual load correction is expressed as

$$R = P - \int \beta^T \sigma dV \quad (11-2)$$

where β is the differential operator which transforms displacements to strains, σ is the current generalized stresses, P is the total applied load vector, and R is the residual load correction.

In order to evaluate the residual load correction accurately, evaluate the integral by summing the contributions from all integration points. The residual load correction feature requires that stresses be stored at all the integration points. Data storage at all integration points is the default in MSC.Marc, but can be overridden in linear analysis by use of the **CENTROID** parameter.

Restarting the Analysis

The model definition option **RESTART** creates a restart file for the current analysis which can be used in subsequent analyses. It can also be used to read in a previously generated file to continue the analysis. The **RESTART** option is very important for any multi-increment analysis because it allows you to continue the analysis at a later time. The default situation writes the restart information to unit 8 and reads a previously generated file from unit 9. For post processing, option **RESTART** can be used to plot or combine load cases (see **CASE COMBIN**). Upon restart, you can use the model definition **REAUTO** option to redefine parameters associated with an automatic load sequence.

To save storage space, it is not necessary to store each increment of analysis. The frequency can be set using the **RESTART** option, and subsequently modified using the **RESTART INCREMENT** option. It is also possible to store only the last converged solution by using the **RESTART LAST** option. This should not be used with elastic analysis because the stiffness matrix is not stored.

Full Newton-Raphson Algorithm

The basis of the Newton-Raphson method in structural analysis is the requirement that equilibrium must be satisfied. Consider the following set of equations:

$$K(u)\delta u = F - R(u) \quad (11-3)$$

where u is the nodal-displacement vector, F is the external nodal-load vector, R is the internal nodal-load vector (following from the internal stresses), and K is the tangent-stiffness matrix. The internal nodal-load vector is obtained from the internal stresses as

$$R = \sum_{\text{elem}} \int_V \beta^T \sigma dv \quad (11-4)$$

In this set of equations, both R and K are functions of u . In many cases, F is also a function of u (for example, if F follows from pressure loads, the nodal load vector is a function of the orientation of the structure). The equations suggest that use of the full Newton-Raphson method is appropriate.

Suppose that the last obtained approximate solution is termed δu^i , where i indicates the iteration number. Equation 11-3 can then be written as

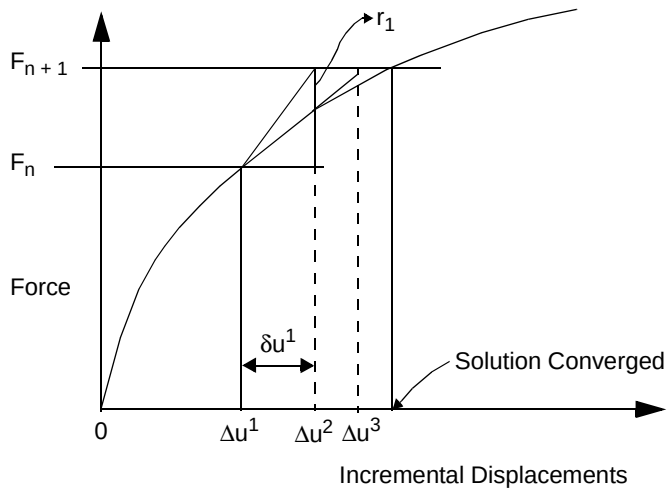
$$K(u_{n+1}^{i-1})\delta u = F - R(u_{n+1}^{i-1}) \quad (11-5)$$

This equation is solved for δu^i and the next appropriate solution is obtained by

$$\Delta u^i = \Delta u^{i-1} + \delta u^i \text{ and } u_{n+1}^i = u_{n+1}^{i-1} + \delta u^i \quad (11-6)$$

Solution of this equation completes one iteration, and the process can be repeated. The subscript n denotes the increment number representing the state $t = n$. Unless stated otherwise, the subscript $n + 1$ is dropped with all quantities referring to the current state.

The full Newton-Raphson method is the default in MSC.Marc (see Figure 11-3). The full Newton-Raphson method provides good results for most nonlinear problems, but is expensive for large, three-dimensional problems, when the direct solver is used. The computational problem is less significant when the iterative solvers are used.

**Figure 11-3** Full Newton-Raphson

Modified Newton-Raphson Algorithm

The modified Newton-Raphson method is similar to the full Newton-Raphson method, but does not reassemble the stiffness matrix in each iteration.

$$K(u^0)\delta u^i = F - R(u^{i-1}) \quad (11-7)$$

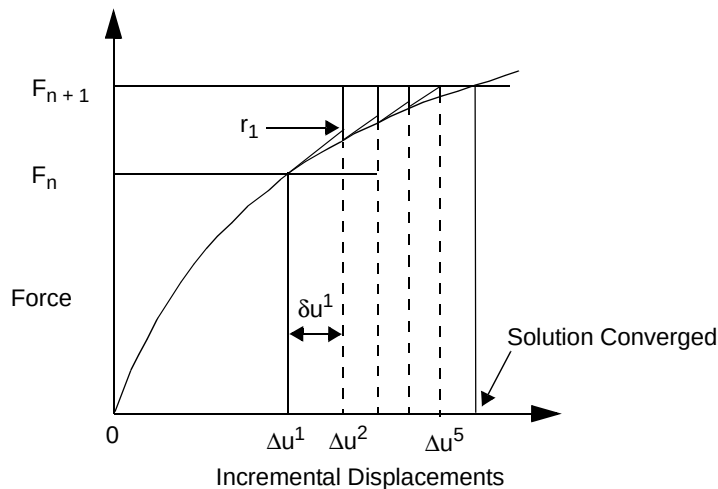


Figure 11-4 Modified Newton-Raphson

The process is computationally inexpensive because the tangent stiffness matrix is formed and decomposed once. From then on, each iteration requires only forming the right-hand side and a backward substitution in the solution process. However, the convergence is only linear, and the potential for a very large number of iterations, or even nonconvergence, is quite high.

If contact or sudden material nonlinearities occur, reassembly cannot be avoided. The modified Newton-Raphson method is effective for large-scale, only mildly nonlinear problems. When the iterative solver is employed, simple back substitution is not possible, making this process ineffective. In such cases, the full Newton-Raphson method should be used instead.

If the load is applied incrementally, MSC.Marc recalculates the stiffness matrix at the start of each increment or at selected increments, as specified.

Strain Correction Method

The strain correction method is a variant of the full Newton method. This method uses a linearized strain calculation, with the nonlinear portion of the strain increment applied as an initial strain increment in subsequent iterations and recycles. This method is appropriate for shell and beam problems in which rotations are large, but membrane stresses are small.

In such cases, rotation increments are usually much larger than the strain increments, and, hence, the nonlinear terms can dominate the linear terms. After each displacement update, the new strains $E_{\alpha\beta}^{i+1}$ are calculated from u^i and $\delta u (= \delta u^i)$ which yield

$$E_{\alpha\beta}^{i+1} = E_{\alpha\beta}^i + \frac{1}{2}(\delta u_{\alpha,\beta} + \delta u_{\beta,\alpha}) + u_{\kappa,\alpha}^i \delta u_{\kappa,\beta} + \delta u_{\kappa,\alpha} u_{\kappa\beta}^i + \delta u_{\kappa,\alpha} \delta u_{\kappa,\alpha} \quad (11-8)$$

This expression is linear except for the last term. Since the iteration procedures start with a fully linearized calculation of the displacement increments, the nonlinear contributions yield strain increments inconsistent with the calculated displacement increments in the first iteration. These errors give rise to either incorrect plasticity calculations (when using small strain plasticity method), or, in the case of elastic material behavior, yields erroneous stresses. These stresses, in their turn, have a dominant effect on the stiffness matrix for subsequent iterations or increments, which then causes the relatively poor performance.

The remedy to this problem is simple and effective. The linear and nonlinear part of the strain increments are calculated separately and only the linear part of

$$(E_{\alpha\beta}^l) = E_{\alpha\beta}^i + \frac{1}{2}(\delta u_{\alpha,\beta} + \delta u_{\beta,\alpha}) + u_{\kappa,\alpha}^i \delta u_{\kappa,\beta} + \delta u_{\kappa,\alpha} u_{\kappa\beta}^i \quad (11-9)$$

is used for calculation of the stresses. The nonlinear part

$$(E_{\alpha\beta}^{nl})^{i+1} = \frac{1}{2} \delta u_{\kappa,\alpha} \delta u_{\kappa,\beta} \quad (11-10)$$

is used as an “initial strain” in the next iteration or increment, which contributes to the residual load vector defined by

$$R^C = \int_{V_0} \delta_{\kappa,\beta} X_{\kappa,\alpha} L^{\alpha\beta\gamma\delta} \Delta E_{\gamma\delta}^{nl} dV \quad (11-11)$$

This “strain correction” term is defined by

$$K(u_{n+1}^i)\delta u = F - R(u_{n+1}^i) - R^C \quad (11-12)$$

Since the displacement and strain increments are now calculated in a consistent way, the plasticity and/or equilibrium errors are greatly reduced. The performance of the strain correction method is not as good if the displacement increments are (almost) completely prescribed, which is not usually the case. Finally, note that the strain correction method can be considered as a Newton method in which a different stiffness matrix is used.

The Secant Method

The secant method used by MSC.Marc is based on the Davidon-rank one, quasi-Newton update. The secant method is similar to the modified Newton-Raphson method in that the stiffness matrix is calculated only once per increment. The residual is modified to improve the rate of convergence. When the iterative solver is employed, simple back substitution is not possible, making this process ineffective. Use the full Newton-Raphson method instead.

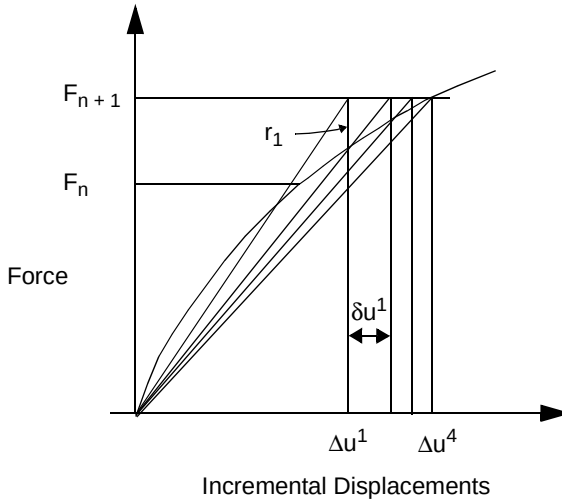


Figure 11-5 Secant Newton

The quasi-Newton requirement is that a stiffness matrix for iteration i could be found based on the right-hand sides of iterations, i and $i - 1$, as follows

$$K^i \delta u^i = [F - R(u_{n+1}^i)] - [F - R(u_{n+1}^{i-1})] = r^i - r^{i-1} \quad (11-13)$$

This problem does not uniquely determine K_i . The Davidon-rank one update uses an additive form on the inverse of the tangent stiffness matrix as follows:

$$(K^i)^{-1} = (K^0)^{-1} + \frac{[\delta u^{i-1} - (K^0)^{-1}(r^i - r^{i-1})][\delta u^{i-1} - (K^0)^{-1}(r^i - r^{i-1})]^T}{[\delta u^{i-1} - (K^0)^{-1}(r^i - r^{i-1})]^T (r^i - r^{i-1})} \quad (11-14)$$

At every iteration, a modified Newton-Raphson solution is obtained first:

$$(\delta u^*)^i = (K^1)^{-1} r^i \quad (11-15)$$

Substitution in [Equation 11-13](#) allows the direct calculation of δu^i by means of several vector operations. First, a coefficient C is obtained

$$C = \frac{[\delta u^{i-1} + (\delta u^*)^i - (\delta u^*)^{i-1}]^T r^i}{[\delta u^{i-1} + (\delta u^*)^i - (\delta u^*)^{i-1}]^T (r - r^{i-1})} \quad (11-16)$$

and finally, a correction to incremental displacements is calculated as:

$$\delta u^i = (1 - C)(\delta u^*)^i - C\delta u^{i-1} + C(\delta u^*)^{i-1} \quad (11-17)$$

Direct Substitution

In the Eulerian formulation (R-P FLOW parameter), the governing equation of the system can be expressed as

$$Kv = F \quad (11-18)$$

where v is a velocity vector, and F is a force vector.

This equation is very nonlinear because K is a nonlinear function of v . By default, a direct substitution method is used to solve the problem. If v^i is the velocity at iteration i , the result of iteration $i + 1$ is

$$K(v^i) v^{i+1} = F \quad (11-19)$$

If this method does not converge in 10 iterations, it is possible to switch into a full Newton-Raphson method.

Load Correction

You usually stop iterating whenever the solution reaches a certain precision; this can be measured, for instance, by how close

$$r = F - R \quad (11-20)$$

is from zero, or a variety of other measurements (convergence on energy or generalized displacements).

In the solution of linearized set of equations, there is always a residual error, r , in a solution. The best way not to let it creep up during incrementation is to carry this error into the next increment. This procedure is known as Residual Load Correction.

The solution of the system of equations for which the solution needs to be obtained is:

$$K(u_{n+1}^i) \delta u^i = r_{n+1}^i + r_n \quad (11-21)$$

Arc-length Methods

The solution methods described above involve an iterative process to achieve equilibrium for a fixed increment of load. Besides, none of them have the ability to deal with problems involving snap-through and snap-back behavior. An equilibrium path as shown in [Figure 11-6](#) displays the features possibly involved.

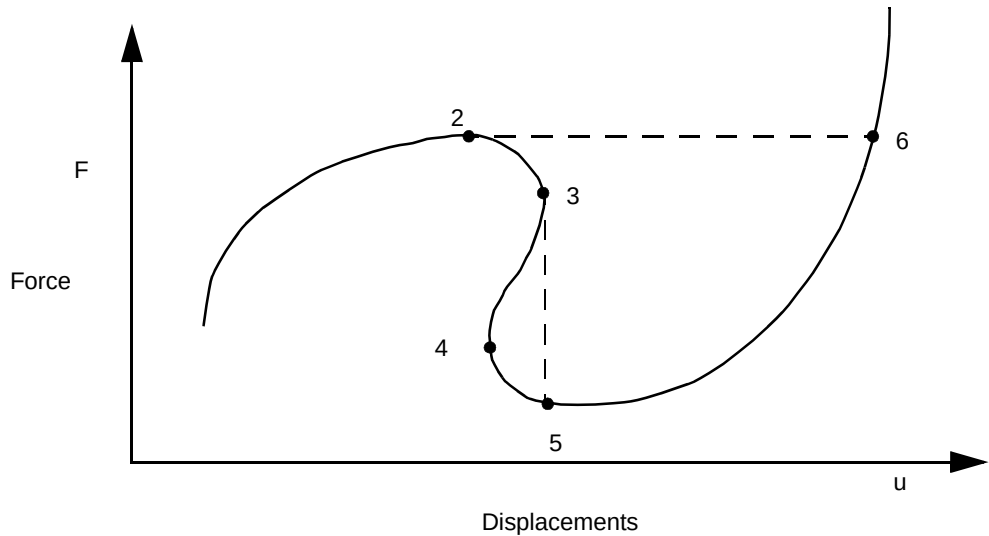


Figure 11-6 Snap-through Behavior

The issue at hand is the existence of multiple displacement vectors, u , for a given applied force vector, F . This method provides the means to ensure that the correct displacement vector is found by MSC.Marc. If you have a load controlled problem, the solution tends to jump from point 2 to 6 whenever the load increment after 2 is applied. If you have a displacement controlled problem, the solution tends to jump from 3 to 5 whenever the displacement increment after 3 is applied. Note that these problems appear essentially in quasi-static analyses. In dynamic analyses, the inertia forces help determine equilibrium in a snap-through problem.

Thus, in a quasi-static analysis sometimes it is impossible to find a converged solution for a particular load (or displacement increment):

$$\lambda_{n+1}F - \lambda_n F = \Delta\lambda F$$

This is illustrated in Figure 11-6 where both the phenomenon of snap-through (going from point 2 to 3) and snap-back (going from point 3 to 4) require a solution procedure which can handle these problems without going back along the same equilibrium curve.

As shown in Figure 11-7, assume that the solution is known at point A for load level $\lambda_n F$. For arriving at point B on the equilibrium curve, you either reduce the step size or adapt the load level in the iteration process. To achieve this end, the equilibrium equations are augmented with a constraint equation expressed typically as the norm of incremental displacements. Hence, this allows the load level to change from iteration to iteration until equilibrium is found.

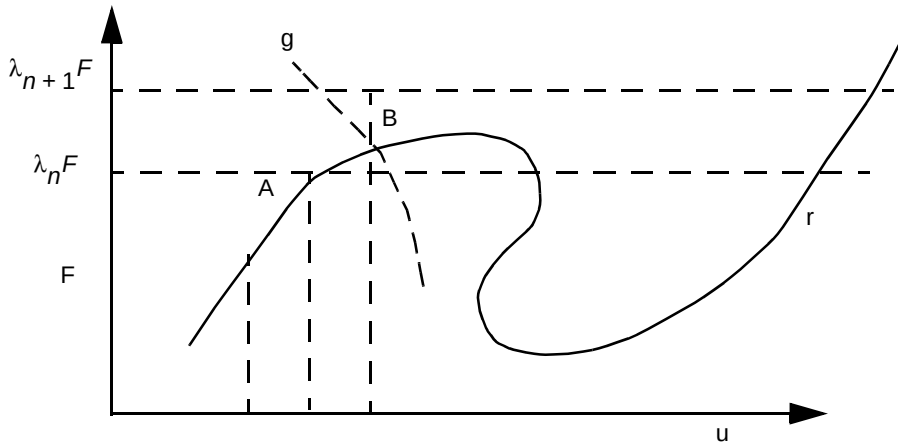


Figure 11-7 Intersection of Equilibrium Curve with Constraining Surface

The augmented equation, $c(u, \lambda)$, describes the intersection of the equilibrium curve with an auxiliary surface g for a particular size of the path parameter η :

$$r(u, \lambda) = \lambda F - R(u) = 0 \quad (11-22)$$

$$c(u, \lambda) = g(u, \lambda) - \Delta\eta = 0$$

Variations of the parameter η moves the surface whose intersection with the equilibrium curve r generates a sequence of points along the curve. The distance between two intersection points, denoted with η_0 and η , denoted by l is the so-called arc-length.

Linearization of Equation 11-22 around point A in Figure 11-7 yields:

$$\begin{bmatrix} K & P \\ n^T & n_0 \end{bmatrix} \begin{Bmatrix} \delta u \\ \delta \lambda \end{Bmatrix} = \begin{Bmatrix} -r \\ -r_0 \end{Bmatrix} \quad (11-23)$$

where:

$$K = \frac{\partial r}{\partial u} : P = \frac{\partial r}{\partial \lambda} \quad (11-24)$$

$$n^T = \frac{\partial c}{\partial u} : n_0 = \frac{\partial c}{\partial \lambda} \quad (11-25)$$

$$r = \lambda F - R \quad (11-26)$$

$$r_0 = g(u, \lambda) - \Delta \eta \quad (11-27)$$

It can be noted that a standard Newton-Raphson solution procedure is obtained if the constraint condition is not imposed. The use of the constraint equation causes a loss of the banded system of equations which would have been obtained if only the K matrix was used. Instead of solving the $N + 1$ set of equations iteratively, the block elimination process is applied.

Consider the residual at iteration i to which the fraction of load level

λ^{i-1} corresponds

$$r^i(\lambda^{i-1}) = \lambda^{i-1}F - R^i(u^{i-1}) \quad (11-28)$$

The residual for some variation of load level, $\delta \lambda^i$, becomes

$$r^i(\lambda^{i-1} + \delta \lambda^i) = \delta \lambda^i F + r^i(\lambda^{i-1}) \quad (11-29)$$

which can be written as:

$$\delta u^i(\lambda^{i-1} + \delta \lambda^i) = \delta u^i(\lambda^{i-1}) + \delta \lambda^i \delta u_*^i \quad (11-30)$$

$$\text{where } \delta u^i(\lambda^{i-1}) = (K^i)^{-1} r \quad (11-31)$$

$$\text{and } \delta u_*^i = (K^i)^{-1} F \quad (11-32)$$

Notice that δu_*^i does not depend on the load level. The equation above essentially establishes the influence of a change in the load level $\delta \lambda^i$ during one iteration on the change in displacement increment for that iteration. After one iteration is solved, this equation is used to determine the change in the load level such that the constraint is followed. There are several arc-length methods corresponding to different constraints.

Among them, the most well-known arc-length method is one proposed by Crisfield, in which the iterative solution in displacement space follows a spherical path centered around the beginning of the increment. This requirement is translated in the formula:

$$c = l^2 = \Delta u^i \Delta u^i \quad (11-33)$$

where l is the arc length. The above equation with the help of [Equation 11-30](#) and [Equation 11-13](#) is applied as:

$$\begin{aligned} & [(\delta u_*^i)^T \delta u_*^i](\delta \lambda^i)^2 + [2(\Delta u^{i-1} + \delta u^i(\lambda^{i-1}))^T \delta u_*^i](\delta \lambda^i) + \\ & [(\Delta u^{i-1} + \delta u^i(\lambda^{i-1}))^T (\Delta u^{i-1} + \delta u^i(\lambda^{i-1})) - l^2] = 0 \end{aligned} \quad (11-34)$$

The equation above is interpreted with $i = 1$ and $\delta u^1 = 0$ in the prediction phase while retaining the full form of [Equation 11-34](#) in the correction phase. Two solutions for $\delta \lambda$ are available. We choose the one that maintains a positive angle of the displacement increment from one iteration to the next.

The two roots of this scalar equation are $(\delta \lambda^i)_1$ and $(\delta \lambda^i)_2$. To avoid going back on the original load-deflection curve, the angle between the incremental displacement vectors, Δu^{i-1} and Δu^i (before and after the current iteration, respectively) should be positive. Two alternative values of Δu^i (namely, $(\Delta u^i)_1$ and $(\Delta u^i)_2$ corresponding to $(\delta \lambda^i)_1$ and $(\delta \lambda^i)_2$ are obtained and the cosine of two corresponding angles (ϕ_1 and ϕ_2) are given by

$$\cos \phi_1 = \frac{[(\Delta u_{n+1}^i)_1]^T \Delta u_{n+1}^{i-1}}{l} \quad (11-35)$$

$$\text{and } \cos \phi_2 = \frac{[(\Delta u_{n+1}^i)_2]^T \Delta u_{n+1}^{i-1}}{l} \quad (11-36)$$

The constraints in Equation 11-33 and Equation 11-34 are imposed at every iteration. Disadvantage of the quadratic equation suggested by Crisfield is the introduction of an equation with two roots and thus the need for an extra equation to solve the system for the calculated roots if two real roots exists. This situation arises when the

contribution Δu^1 (or δu^1) is very large in comparison to the arc-length. This can be avoided in most cases by setting sufficiently small values of the error tolerance on the residual force. In case the above situation still persists despite the reduction of error tolerance, MSC.Marc has two options to proceed:

- a. To attempt to continue the analysis with the load increment used in the initial step of auto increment process.
- b. Use the increment resulting from the linear constraint for the load.

This is circumvented in Ramm's procedure due to the linearization.

Another approach to impose the constraint is due to Ramm, who also makes use of a quadratic equation to impose the constraint giving rise to the Riks-Ramm method. The difference is that while Crisfield imposes the constraint as a quadratic equation, Ramm linearized the constraint.

Geometrically, the difference between the two methods is that the Crisfield method enforces the correction on the curve of the augmented equation introducing no residual for the augmented equation. Ramm takes the intersection between the linearizations of the curves which gives a residual of the augmented equation for the next step. Both methods converge to the same solution, the intersection of the two curves, unless approximations are made.

The Riks-Ramm constraint is linear, in that:

$$c = l^2 = \Delta u_n \Delta u_{n+1}$$

which results in a linear equation for $\delta\lambda$:

$$\Delta u_n^T (\delta u^i + \delta\lambda \delta u_*^i) = l^2$$

Thus, the load parameter predictor is calculated as:

$$\delta\lambda_{n+1}^1 = \frac{\|\Delta u_n\| l - (\Delta u_n)^T [(K^i)^{-1} r^i]}{\Delta u_n^T (\delta u_*^1)} \quad (11-38)$$

while during the corrector phase it is:

$$\delta\lambda_{n+1}^i = - \frac{(\Delta u_{n+1}^i)^T [(K^i)^{-1} r^i]}{(\Delta u_{n+1}^i)^T (\delta u_*^i)} \quad (11-39)$$

It is noted that in the definition of the constraint, the normalized displacement of the previous step is used for the normal to the auxiliary surface $\frac{\partial c}{\partial u} = n$. Thus, problems can arise if the step size is too big. In situations with sharp curvatures in the solution path, the normal to the prediction may not find intersections with the equilibrium curve. Note that the norm of the displacement increment during the iterations is not constant in Riks-Ramm method.

In contact problems, sudden changes of the stiffness can be present (due to two bodies which are initially not in contact suddenly make contact). Hence, a potential problem exists in the Riks-Ramm method if the inner-product of the displacement due to the load vector δu_*^i and the displacement increment Δu_n is small. This could result in a very large value of the load increment for which convergence in the subsequent iterations is difficult to achieve. Therefore, a modified predictor can be used resulting in a modified Riks-Ramm procedure as:

$$\Delta \lambda^1 = \frac{l_{n-1} \|\delta u_*^1\| - [\alpha \delta u_*^1]^T \delta u^1}{[\alpha \delta \Delta u_*^1] \Delta u_*^1} \quad (11-40)$$

where

$$\alpha = \frac{\Delta u_n^T \delta u_*^i}{\|\Delta u_n^T \delta u_*^i\|} \quad (11-41)$$

This method effectively scales the load increment to be applied in the prediction and is found to be effective for contact problems.

Refinements and Controls

The success of the methods depend on the suitable choice of the arc-length:

$$C = l^2$$

The initial value of the arc-length is calculated from the initial fraction β of the load specified by you in the following fashion:

$$K \delta u = \beta F - R \quad (11-42)$$

$$l_{ini}^2 = \|\Delta u\| \quad (11-43)$$

In subsequent steps the arc-length can be reduced or increased at the start of a new load step depending on the number of iterations I_0 in the previous step. This number of iterations is compared with the desired number of iterations I_d which is typically set to 3 or 5. The new arc-length is then given by:

$$l_{\text{new}}^2 = \frac{I_d}{I_0} l_{\text{prev}}^2 \quad (11-44)$$

Two control parameters exist to limit the maximum enlargement or the minimum reduction in the arc-length.

$$\min < \frac{l^2}{l_{\text{ini}}^2} < \max \quad (11-45)$$

In addition, the maximum value can be set to the load multiplier during a particular iteration. In general, control on the limiting values with respect to the arc-length multiplier is preferred in comparison with the maximum fraction of the load to be applied in the iteration since a solution is sought for a particular value of the arc-length.

Also, attention must be paid to the following:

1. In order to tract snap-through problems, the method of allowing solution if the stiffness matrix becomes nonpositive needs to be set.
2. The maximum number of iterations must be set larger than the desired number of iterations.

Remarks

Which solution method to use depends very much on the problem. In some cases, one method can be advantageous over another; in other cases, the converse might be true.

Whether a solution is obtainable or not with a given method, usually depends on the character of the system of equations being solved, especially on the kind of nonlinearities that are involved.

As an example in problems which are linear until buckling occurs, due to a sudden development of nonlinearity, it is necessary for you to guide the arc-length algorithm by making sure that the arc length remains sufficiently small prior to the occurrence of buckling.

Even if a solution is obtainable, there is always the issue of efficiency. The pros and cons of each solution procedure, in terms of matrix operations and storage requirements have been discussed in the previous sections. A very important variable regarding overall efficiency is the size of the problem. The time required to assemble a stiffness matrix, as well as the time required to recover stresses after a solution, vary roughly linearly with the number of degrees of freedom of the problem. On the other hand, the time required to go through the solver varies roughly quadratically with the bandwidth, as well as linearly with the number of degrees of freedom.

In small problems, where the time spent in the solver is negligible, you can easily wipe out any solver gains, or even of assembly gains, with solution procedures such as a line search which requires a double stress recovery. Also, for problems with strong material or contact nonlinearities, gains obtained in assembly in modified Newton-Raphson can be nullified by increased number of iterations or nonconvergence.

There are some developments widely covered in the literature and available in MSC.Marc which have not been covered in the chapter. Namely, iterative methods and element-by-element iterative solution schemes, which completely avoid the assembly of a global stiffness matrix and the subsequent direct solution of a system of equations. The development of new solution procedures is still an active field of research in the academic community.

Convergence Controls

The default procedure for convergence criterion in MSC.Marc is based on the magnitude of the maximum residual load compared to the maximum reaction force. This method is appropriate since the residuals measure the out-of-equilibrium force, which should be minimized. This technique is also appropriate for Newton methods, where zero-load iterations reduce the residual load. The method has the additional benefit that convergence can be satisfied without iteration.

The basic procedures are outlined below.

1. RESIDUAL CHECKING

$$\frac{\|F_{residual}\|_{\infty}}{\|F_{reaction}\|_{\infty}} < TOL_1 \quad (11-46)$$

$$\frac{\|F_{residual}\|_{\infty}}{\|F_{reaction}\|_{\infty}} < TOL_1 \text{ and } \frac{\|M_{residual}\|_{\infty}}{\|M_{reaction}\|_{\infty}} < TOL_2 \quad (11-47)$$

$$\|F_{residual}\|_{\infty} < TOL_1 \quad (11-48)$$

$$\|F_{residual}\|_{\infty} < TOL_1 \text{ and } \|M_{residual}\|_{\infty} < TOL_2 \quad (11-49)$$

Where F is the force vector, and M is the moment vector. TOL_1 and TOL_2 are control tolerances. $\|F\|_{\infty}$ indicates the component of F with the highest absolute value. Residual checking has two drawbacks. First, if the **CENTROID** parameter is used, the residuals and reactions are not calculated accurately. Second, in some special problems, such as free thermal expansion, there are no reaction forces. The program uses displacement checking in either of these cases.

2. DISPLACEMENT CHECKING

$$\frac{\|\delta u\|_{\infty}}{\|\Delta u\|_{\infty}} < TOL_1 \quad (11-50)$$

$$\frac{\|\delta u\|_{\infty}}{\|\Delta u\|_{\infty}} < TOL_1 \text{ and } \frac{\|\delta \phi\|_{\infty}}{\|\Delta \phi\|_{\infty}} < TOL_2 \quad (11-51)$$

$$\|\delta u\|_{\infty} < TOL_1 \quad (11-52)$$

$$\|\delta u\|_{\infty} < TOL_1 \text{ and } \|\delta \phi\|_{\infty} < TOL_2 \quad (11-53)$$

where Δu is the displacement increment vector, δu is the correction to incremental displacement vector, $\Delta \phi$ is the correction to incremental rotation vector, and $\delta \phi$ is the rotation iteration vector. With this method, convergence is satisfied if the maximum displacement of the last iteration is small compared to the actual displacement change of the increment. A disadvantage of this approach is that it results in at least one iteration, regardless of the accuracy of the solution.

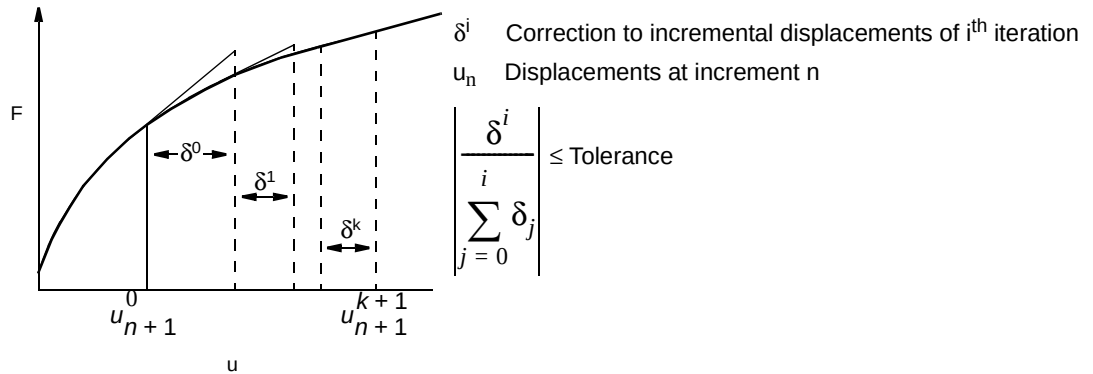


Figure 11-9 Displacement Control

3. STRAIN ENERGY CHECKING

This is similar to displacement testing where a comparison is made between the strain energy of the latest iteration and the strain energy of the increment. With this method, the entire model is checked.

$$\frac{\delta E}{\Delta E} < TOL_1 \quad (11-54)$$

where ΔE is the strain energy of the increment and δE is the correction to incremental strain energy of the iteration. These energies are the total energies, integrated over the whole volume. A disadvantage of this approach is that it results in at least one iteration, regardless of the accuracy of the solution. The advantage of this method is that it evaluates the global accuracy as opposed to the local accuracy associated with a single node.

Different problems require different schemes to detect the convergence efficiently and accurately. To do this, the following combinations of residual checking and displacement checking are also available.

4. RESIDUAL OR DISPLACEMENT CHECKING

This procedure does convergence checking on both residuals (Procedure 1) and displacements (Procedure 2). Convergence is obtained if one converges.

5. RESIDUAL AND DISPLACEMENT CHECKING

This procedure does a convergence check on both residuals and displacements (Procedure 4). Convergence is achieved if both criteria converge simultaneously.

For problems where maximum reactions or displacements are extremely small (even close to the round-off errors of computers), the convergence check based on relative values could be meaningless if the convergence criteria chosen is based on these small values. It is necessary to check the convergence with absolute values; otherwise, the analysis is prematurely terminated due to a nonconvergent solution. Such situations are not predicable and usually happen at certain stages of an analysis. For example, problems with stress free motion (rigid body motion or free thermal expansion) and small displacements (springback or constraint thermal expansion) may need to check absolute value at some stage of the analysis, as shown in [Table 11-1](#). However, it is also difficult to determine when to check the absolute value and how small the absolute criterion value should be. In order to improve the robustness of an FE analysis, MSC.Marc allows you to use the AUTO SWITCH option to switch the convergence check scheme automatically if the above mentioned situation occurs during the analysis. Using the AUTO SWITCH option allows MSC.Marc to automatically change the convergence check scheme to Procedure 4 if small reactions or displacements are detected. This function can be deactivated by specifying an absolute value check as before.

Table 11-1 Effectiveness of various Relative Tolerance Convergence Testing Criterion

Analysis Type	Convergence Variable		
	Displacement/ Rotation	Residual Force/ Torque	Strain Energy
Stress-free motion	Yes	No	No
Springback	No	Yes	No
Free Thermal Expansion	Yes	No	No
Constraint Thermal Expansion	No	Yes	Yes
Yes – relative tolerance testing works. No – relative tolerance testing doesn't work.			

Singularity Ratio

The singularity ratio, R , is a measure of the conditioning of the system of linear equations. R is related to the conditioning number, C , which is defined as the ratio between the highest and lowest eigenvalues in the system. The singularity ratio is an upper bound for the inverse of the matrix conditioning number.

$$1/R \leq C \quad (11-55)$$

C and R establish the growth of errors in the solution process. If the errors on the right-hand side of the equation are less than E prior to the solution, the errors in the solution will be less than δ , with

$$\delta \leq CE \quad (11-56)$$

The singularity ratio is a measure that is computed during the Crout elimination process of MSC.Marc using the direct solver. In this process, a recursive algorithm redefines the diagonal terms

$$K_{kk}^{(k)} = K_{kk}^{(k-1)} - \sum_{m=i}^{k-1} K_{mk} K_{mk} \quad 1 \leq i \leq k-1 \quad (11-57)$$

where i is a function of the matrix profile. K_{kk} is a diagonal of the k^{th} degree of freedom. The singularity ratio is defined as

$$R = \min K_{kk}^{(k)} / K_{kk}^{(k-1)} \quad (11-58)$$

If all $K_{kk}^{(k)}$ and $K_{kk}^{(k-1)}$ are positive, the singularity ratio indicates loss of accuracy during the Crout elimination process. This loss of accuracy occurs for all positive definite matrices. The number of digits lost during the elimination process is approximately equal to

$$n_{lost} = -\log_{10} R \quad (11-59)$$

The singularity ratio also indicates the presence of rigid body modes in the structure. In that case, the elimination process produces zeros on the diagonal $K_{kk}^{(k)} \cong 0$. Exact zeros never appear because of numerical error; therefore, the singularity ratio is of the order

$$R = O(10^{-n_{\text{digit}}}) \quad (11-60)$$

where n_{digit} is the accuracy of floating-point numbers used in the calculation. For most versions of MSC.Marc, $n_{\text{digit}} > 12$. If rigid body modes are present, $K_{kk}^{(k)}$ is very small or negative. If either a zero or a negative diagonal is encountered, execution of MSC.Marc is terminated because the matrix is diagnosed as being singular.

You can force the solution of a nonpositive definite or singular matrix. In this case, MSC.Marc does not stop when it encounters a negative or small term $K_{kk}^{(k)}$ on the diagonal. If you use Lagrangian multiplier elements, the matrix becomes nonpositive definite and MSC.Marc automatically disables the test on the sign of $K_{kk}^{(k)}$. However, it still tests on singular behavior.

Note: The correctness of a solution obtained for a linearized set of equations in a nonpositive definite system is not guaranteed.

Solution of Linear Equations

The finite element formulation leads to a set of linear equations. The solution is obtained through numerically inverting the system. Because of the wide range of problems encountered with MSC.Marc, there are several solution procedures available.

Most analyses result in a system which is real, symmetric, and positive definite. While this is true for linear structural problems, assuming adequate boundary conditions, it is not true for all analyses. MSC.Marc has two main modes of solvers – direct and iterative. Each of these modes has two families of solvers, based upon the storage procedure. While all of these solvers can be used if there is adequate memory, only a subset uses spill logic for an out-of-core solution. Finally, there are classifications based upon nonsymmetric and complex systems. This is summarized below:

	Direct Profile	Iterative Sparse	Direct Sparse	Vendor Provided Sparse*	Multifrontal Sparse
Solver Option	0	2	4	6	8
Real Symmetric	Yes	Yes	Yes	Yes	Yes
Real Nonsymmetric	Yes	No	No	No	Yes
Complex Symmetric	Yes	No	No	SUN only	Yes
Complex nonsymmetric	No	No	No	No	Yes
Out-of-core	Yes	No	Yes	SGI only	Yes
Possible problem with poorly conditioned systems	No	Yes	No	No	No
*Available for SGI, HP, and SUN platforms only.					

The choice of the solution procedure is made through the **SOLVER** option.

Direct Methods

Traditionally, the solution of a system of linear equations was accomplished using direct solution procedures, such as Cholesky decomposition and the Crout reduction method. These methods are usually reliable, in that they give accurate results for virtually all problems at a predictable cost. For positive definite systems, there are no computational difficulties. For poorly conditioned systems, however, the results can

degenerate but the cost remains the same. The problem with these direct methods is that a large amount of memory (or disk space) is required, and the computational costs become very large.

Iterative Methods

MSC.Marc offers iterative solvers as a viable alternative for the solution of large systems. These iterative methods are based on *preconditioned conjugate gradient methods*. The single biggest advantage of these iterative methods is that they allow the solution of very large systems at a reduced computational cost. This is true regardless of the hardware configuration. The disadvantage of these methods is that the solution time is dependent not only upon the size of the problem, but also the numerical conditioning of the system. A poorly conditioned system leads to slow convergence – hence increased computation costs.

When discussing iterative solvers, two related concepts are introduced: *fractal dimension*, and *conditioning number*. Both are mathematical concepts, although the fractal dimension is a simpler physical concept. The *fractal dimension*, the range of which is between 1 and 3, is a measure of the “chunkiness” of the system. For instance, a beam has a fractal dimension of 1, while a cube has a fractal dimension of 3.

The *conditioning number* is related to the ratio of the lowest to the highest eigenvalues of the system. This number is also related to the *singularity ratio*, which has been traditionally reported in the MSC.Marc output when using a direct solution procedure. In problems involving beams or shells, the conditioning number is typically small, because of the large differences between the membrane and bending stiffnesses.

Preconditioners

The choice of preconditioner can substantially improve the conditioning of the system, which in turn reduces the number of iterations required. While all positive definite systems with N degrees of freedom converges in N iterations, a well conditioned system typically converges in less than the square root of N iterations.

The available preconditioners available in the sparse iterative solver are diagonal, scaled, and incomplete Cholesky.

The sparse iterative solver requires an error criteria to determine when convergence occurs. The default is to use an error criteria based upon the ratio between the residuals in the solution and the reaction force. After obtaining the solution of the linear equations u^c evaluate:

$$Ku^C = F^C \quad (11-61)$$

The residual from the solution procedure is:

$$\text{Res} = F^A - F^C = F^A - Ku^C \quad (11-62)$$

If the system is linear (K does not change) and exact numerics are preformed, then $\text{Res} = 0$.

Because this is an iterative method the residual is nonzero, but reduces in size with further iterations. Convergence is obtained when

$$\text{Res}/\text{Reac} < \text{TOL} \quad (11-63)$$

The tolerance is specified through the [SOLVER](#) option.

Storage Methods

In general, a system of linear equations with N unknowns is represented by a matrix of size N by N , or N^2 variables. Fortunately, in finite element or finite difference analyses, the system is “banded” and not all of the entries need to be stored. This substantially reduces the memory (storage) requirements as well as the computational costs.

In the finite element method, additional zeroes often exist in the system, which results in a partially full bandwidth. Hence, the *profile* (or *skyline*) method of storage is advantageous. This profile storage method is used in MSC.Marc to store the stiffness matrix. When many zeroes exist within the bandwidth, the *sparse storage methods* can be quite advantageous. Such techniques do not store the zeroes, but require additional memory to store the locations of the nonzero values. You can determine the “sparsity” of the system (before decomposition) by examining the statements:

- “Number of nodal entries excluding fill in” x
- “Number of nodal entries including fill in” y

If the ratio (x/y) is large, then the sparse matrix storage procedure is advantageous.

Nonsymmetric Systems

The following analyses types result in nonsymmetric systems of equations:

- Inclusion of convective terms in heat transfer analysis
- Coriolis effects in transient dynamic analysis
- Fluid mechanics
- Steady state rolling

- Soil analysis
- Follower force stiffness
- Frictional contact

The first four always result in a nonsymmetric system. The last three can be solved either fully using the nonsymmetric solver, or (approximately) using a symmetric solver. The nonsymmetric problem uses twice as much memory for storing the stiffness matrix.

Complex Systems

MSC.Marc utilizes a complex operator matrix for dynamic harmonic analyses when the damping matrix is present or for dynamic, acoustic, or piezoelectric harmonic analyses. The matrix is always symmetric.

Iterative Solvers

In MSC.Marc, an iterative sparse solver is available using a *sparse matrix technique*. This method is advantageous for different classes of problems.

There exist certain types of analyses for which the sparse iterative solver is not appropriate. These types include:

- elastic analysis
- explicit creep analysis
- complex harmonic analysis
- substructures
- central difference techniques
- eigenvalue analysis
- use of gap elements

Elastic or explicit creep analysis involves repeated solutions using different load vectors. When a direct solver is used, this is performed very efficiently using back substitution. However, when an iterative solver is used, the stiffness matrix is never inverted, and the solution associated with a new load vector requires a complete re-solution.

The sparse iterative solver can exhibit poor convergence when shell elements or Herrmann incompressible elements are present.

Basic Theory

A linear finite element system is expressed as:

$$Ku = F \quad (11-64)$$

And a nonlinear system is expressed as:

$$K^T \Delta u = F - R = r \quad (11-65)$$

where K is the elastic stiffness matrix, K^T is the tangent stiffness matrix in a nonlinear system, Δu is the displacement vector, F is the applied load vector, and r is the residual.

The linearized system is converted to a minimization problem expressed as:

$$\psi(u) = 1/2 u^T K u - u^T F \quad (11-66)$$

For linear structural problems, this process can be considered as the minimization of the potential energy. The minimum is achieved when

$$u = K^{-1} F \quad (11-67)$$

The function ψ decreases most rapidly in the direction of the negative gradient.

$$\nabla \psi(u) = F - K u = r \quad (11-68)$$

The objective of the iterative techniques is to minimize function, ψ , without inverting the stiffness matrix. In the simplest methods,

$$u_{k+1} = u_k + \alpha_k r_k \quad (11-69)$$

where

$$\alpha_k = r_k^T r_k / r_k^T K r_k \quad (11-70)$$

The problem is that the gradient directions are too close, which results in poor convergence.

An improved method led to the *conjugate gradient method*, in which

$$u_{k+1} = u_k + \alpha_k P_k \quad (11-71)$$

$$\alpha_k = P_k^T r_{k-1} / P_k^T K P_k \quad (11-72)$$

The trick is to choose P_k to be K conjugate to $P_1, P_2, \dots, P_{k-1}$. Hence, the name “conjugate gradient methods. Note the elegance of these methods is that the solution may be obtained through a series of matrix multiplications and the stiffness matrix never needs to be inverted.

Certain problems which are ill-conditioned can lead to poor convergence. The introduction of a preconditioner has been shown to improve convergence. The next key step is to choose an appropriate preconditioner which is both effective as well as computationally efficient. The easiest is to use the diagonal of the stiffness matrix. The incomplete Cholesky method has been shown to be very effective in reducing the number of required iterations.

Flow Diagram

Figure 11-10 is a diagram showing the flow sequence of MSC.Marc. This diagram shows the input phase, equivalent nodal load vector calculation, matrix assembly, matrix solution, stress recovery, and output phase. It also indicates load incrementation and iteration within a load increment.

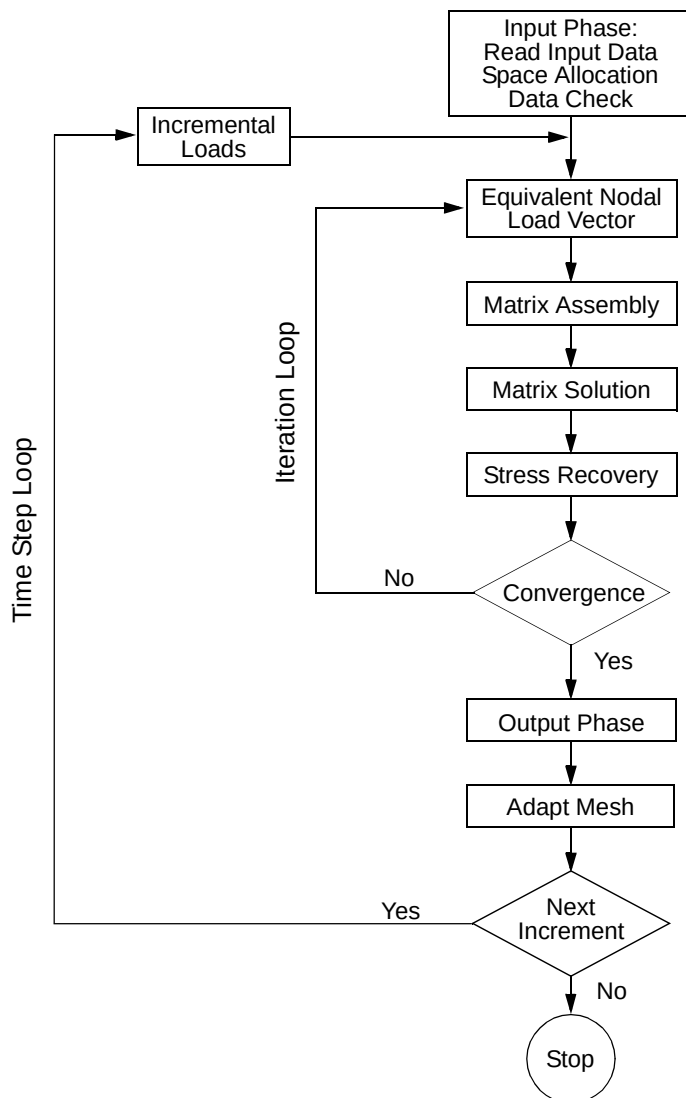


Figure 11-10 MSC.Marc Flow Diagram

References

1. Bergan, P. G. and E. Mollestad, "An Automatic Time-Stepping Algorithm for Dynamic Problems", *Computer Methods in Applied Mechanics and Engineering*, 49, (1985), pp. 299 - 318
2. Bathe, K. J. *Finite Element Procedures*, Prentice Hall, Englewood Cliffs, NJ, 1996.
3. Riks, E. "An incremental approach to the solution of solution and buckling problems", *Int. J. of Solids and Structures*, V. 15, 1979.
4. Riks, E. "Some Computational Aspects of the Stability Analysis of Nonlinear Structures", *Comp. Methods in Appl. Mech. and Eng.*, 47, 1984.
5. Crisfield, M. A. "A fast incremental iterative procedure that handles snapthrough", *Comput. & Structures*, V. 13, 1981.
6. Ramm, E. "Strategies for tracing the nonlinear response near limit points," in K. J. Bathe et al (eds), *Europe-US Workshop on Nonlinear Finite Element Analysis in Structural Mechanics*, Ruhr University Bochum, Germany, Springer-Verlag, Berlin, pp/ 63-89. Berlin, 1985.

CHAPTER 12

Output Results

- Workspace Information
- Increment Information
- Selective Printout
- Restart
- Element Information
- Nodal Information
- Post File
- Program Messages
- MSC.Marc ADAMS Results Interface
- MSC.Marc SDRC I-DEAS Results Interface
- Status File

This chapter summarizes the information that MSC.Marc provides in the output. In addition to reviewing your input, MSC.Marc provides information about the procedures the program uses and the workspace allocation. All calculated results are automatically written to the output file unless the user specifically requests otherwise.

Workspace Information

MSC.Marc reports several aspects of workspace information, specifically, the allocation of memory workspace and the size of the work files.

For the general memory, MSC.Marc first gives the workspace needed for the input and stiffness assembly. This number tells how much memory is needed to store the user-supplied data, the program-calculated data, and two-element stiffness matrices. Each set of data comprises three parts: overhead, element information, and nodal information. Element information consists of properties, geometries, strains, and stresses, and nodal information consists of coordinates, displacements, and applied forces.

Next, MSC.Marc specifies the internal core allocation parameters. These values are useful in user subroutines and are often provided to the user subroutines. The values provided here are:

- number of degrees of freedom per node (ndeg)
- maximum number of coordinates per node
- maximum number of invariants per integration point (neqst)
- maximum number of nodes per element (nnodmx)
- maximum number of stress components per integration point (nstrmx)
- number of strain components per integration point (ngens)

Note the following:

- The number of degrees of freedom per node indicates the number of boundary conditions necessary to eliminate rigid body modes.
- The number of stresses per integration point is the number of stresses per layer multiplied by the number of layers.
- The number of invariants per integration point is the number of layers for either shell or beam elements.

The [ELSTO](#) parameter stores element information on an auxiliary storage device, rather than in main memory. You can invoke this option, or MSC.Marc invokes it automatically. In this option, MSC.Marc:

- sets a flag for element storage (IELSTO)
- specifies how many words are used per element (NELSTO)
- specifies the number of elements per buffer (MXELS)
- specifies the total amount of space needed to store this information

The number of elements per buffer should be at least two, and you can change the buffer size in the [ELSTO](#) parameter option to increase the number of elements which will be in the buffer.

MSC.Marc reports the memory for different parts as it is allocated.

For contact bodies:

allocated workspace of 4567 words of memory for body 1

For incremental backup:

space needed for incremental backup: 555313

For boundary conditions:

allocated 600 words of memory due to kinematic boundary conditions

We also get printouts like

total workspace needed with in-core matrix storage = 893281

This printout refers to the memory in general memory (see [MSC.Marc Workspace Requirements](#) in Chapter 2 Program Initiation) and is not the real total memory used.

The solution of the equations of equilibrium result in the formation of the operator matrix and the decomposition/inversion of the matrix. In an mechanical analysis, this is the global stiffness matrix. For some solvers, there is an intermediate stage where the matrix is restructured from a sparse format to a more optimal format for numerical calculations. The allocation of this memory is reported in the memory summary (see below) either under `solver: first part` or under the `allocated separately: category`. This is summarized as follows:

Solver	Operator Matrix	Restructured Matrix	Decomposition or Inverted Matrix
0 -Profile direct	first part	first part	first part
2 - Sparse Iterative	first part	N/A	N/A
4 - Sparse Direct	first part	first part	first part
6 - Hardware Provided	first part	allocated separately	allocated separately
8 - Multi-front Direct Sparse	first part	N/A	allocated separately

For solvers 0, 2, and 4, all the memory for the matrix solution is allocated within the general memory. For solver 6 and 8, the workspace for the restructuring stage (solver 6 only) and decomposition part is allocated separately outside of the general memory.

Solvers 0 and 4 have an out-of-core option for both storing the operator matrix and for inverting the matrix. Solver 6 (SGI) and solver 8 have an out-of-core option for inverting the matrix only. For these solvers, there must be enough memory to assemble the operator matrix in-core.

When using solver 0, if the out-of-core solver is invoked, one observes:

```
total workspace needed with in-core matrix storage =      89686633
out-of-core matrix storage will be used
core allocation based on      50498 nodal entries per assembly buffer.
...file 14-- maximum record length=      40960
               approximate no. of words on file=      3686400
...file 11-- maximum record length=  44622892
               approximate no. of words on file=      99316916
...file 12-- maximum record length=  44622892
               approximate no. of words on file=      89245784
...file 13-- maximum record length=       712
               approximate no. of words on file=       712
```

If solver 4 were used, one would observe:

```
total workspace needed with sparse in-core matrix storage =      34288801
out-of-core matrix storage will be used
core allocation based on      50498 nodal entries per assembly buffer.
...file 14-- maximum record length=      40960
               approximate no. of words on file=      3686400
core allocation based on     160473 nodal entries per assembly buffer.
...file  2-- maximum record length=      40960
               approximate no. of words on file=     23429120
```

Note that until 14 contains the true sparse matrix, and, as the same problem is used, the I/O is the same for both solvers 0 and 4. For solver 0, this matrix gets expanded on unit 11 using a row blocking system, and decomposed to unit 12. For solver 4, it gets expanded and decomposed using unit 2. Solver 4 uses substantially less I/O than solver 0.

For solver 8, we observe

```
solver workspace needed for out-of-core matrix storage =      6389794
solver workspace needed for in-core matrix storage      =      31339042
matrix solution will be out-of-core
approximate disk space for out-of-core matrix storage =      62678084
```

At the end of the run, MSC.Marc prints out a summary of the memory used. This typically looks something like

memory usage:	Mbyte	words	% of total
within general memory (sizing):			
element storage:	0	648280	37.7
nodal vectors:	0	80928	4.7
optimization related:	0	4	0.0
element stiffness matrices:	0	23426	1.4
miscellaneous	0	66975	3.9
solver: first part	0	74172	4.3
allocated separately:			
solver:	0	252902	14.7
incremental backup:	0	556179	32.3
contact:	0	13462	0.8
tyings:	0	1000	0.1
transformations:	0	3488	0.2
kinematic boundary conditions:	0	600	0.0

total:	4	1721416	
general memory (sizing) allocated	20	6250000	
general memory (sizing) used	0	893785	
totally allocated workspace	24	7077631	
totally used workspace	4	1721416	

In a parallel analysis, MSC.Marc also prints out the memory usage summed over the domains. The memory printout, during the execution, can also be obtained by means of the control file. Suppose the job defined in the input file `jobname.dat` is running. The memory summary printout is obtained by creating a file called `jobname.cnt` with a single line containing the word “memory”. The information is output at the beginning of the next increment.

If the **SIZING** parameter is to be used for initial memory allocation, the value to look for is the one for the line

```
general memory (sizing) used0 893785
```

In this example, one would choose a value of 900000 for **SIZING** to insure that no dynamic reallocation of memory is performed.

Increment Information

A variety of information is given for each increment, and often for each iteration, of the analysis. This information is useful in determining the accuracy and the stability of the analysis.

Summary of Loads

The printout entitled “Load Increments Associated with Each Degree of Freedom” allows you to check the load input quickly. It represents the sum, over all nodes in the mesh, of the point loads and the equivalent forces obtained after distributed loads (pressures, body forces) are applied. For example, you can easily check a pressure because the total force in a global coordinate direction would be the projected area normal to that direction of the surface upon which the pressure is applied, multiplied by the pressure magnitude.

Timing Information

The amount of CPU necessary to reach the given location in the analysis is indicated by the following output:

- start of increment
- start of assembly
- start of matrix solution
- end of matrix solution
- end of increment

As an example, subtract the time associated with the start of the matrix solution from the time associated with the end of the matrix solution to determine how much time was spent in the equation solver.

Singularity Ratio

The singularity ratio is a measure of the conditioning of the matrix (see the [Appendix A](#) for further details). This ratio is printed each time there is a solution of the matrix equations. You can measure the influence of the nonlinearities in the structure by examining the change in singularity ratios between increments.

Convergence

Several messages are printed that concern the convergence of the solution. These messages indicate the displacement, velocity, or residual error and are very important because they provide information concerning the accuracy of the solution procedure.

These messages also indicate the ratio of the error and its relative quantity. This ratio must be less than that given in the [CONTROL](#) option for convergence to occur. See [Appendix A](#) for details on convergence testing.

When the iterative solver is used, additional messages are printed regarding the convergence of the solution procedures.

Selective Printout

MSC.Marc gives you several options and user subroutines for the control of the MSC.Marc output.

Options

PRINT CHOICE allows you to select how much of the element and nodal information is to be printed. The possible selections are

- group of elements
- group of nodes
- which layers (form beam and shell elements)
- which integration points
- increment frequency between printouts

The data entered through this option remain in control until you insert a subsequent **PRINT CHOICE** set. You can include such a set with either the model definition or with the history definition set. To obtain the default printout after a previous **PRINT CHOICE**, invoke **PRINT CHOICE** using blank entries.

Note: The **PRINT CHOICE** option has no effect on the restart or post file.

You can use the **PRINT ELEMENT** capability as a replacement for, or in conjunction with, the **PRINT CHOICE** option. The enhancements this option offers over the **PRINT CHOICE** option are the following:

1. You have a choice of integration and layer points for each element to be printed, which is especially useful when several different element types are used in one analysis.
2. You have a choice of the type of quantity to be printed, for instance, stresses could be printed for all elements, but strains for only a few.
3. The types of output quantities that can be selected are stresses, strains, creep strain, thermal strain, cracking strain, Cauchy stress, state variables, strain energy, or all nonzero quantities.
4. The output can be placed in a file other than the standard output file.
5. The **PRINT NODE** option can be used to obtain element quantities such as stresses and strains at the nodal points of each element. These are obtained by extrapolating the integration point values to the nodes.

The **PRINT NODE** option is an alternative to the **PRINT CHOICE** option for controlling the output of nodal quantities. The additional capabilities of this option compared to the **PRINT CHOICE** are as follows:

1. You can choose which of the following nodal quantities are to be printed:
 - incremental displacements
 - total displacements velocities
 - accelerations
 - reaction forces
 - generalized stresses
2. Different quantities can be printed for different nodal points.
3. The output can be placed in a file other than the standard output file.

Note: All quantities are saved on the restart file, and you can obtain them at a later time by using the **RESTART** option. Quantities that were not printed out are available for later use.

The **SUMMARY** option allows you to get a quick summary of the results obtained in the analysis. This option prints the maximum and minimum quantities in tabular form. The is designed for direct placement into reports. You control the increment frequency of summary information and the file unit to which the information is written.

The **SUMMARY** option reports on the physical components and the Tresca, von Mises, and mean values of stress, plastic strain, and creep strain. It also reports on such nodal quantities as displacements, velocities, accelerations, and reaction forces.

The sort options (**ELEM SORT** and **NODE SORT**) allow you to sort calculated quantities in either ascending or descending order. These quantities may either be sorted by their real magnitude or their absolute magnitude. The sort options also allow you to control the type of quantity to be sorted, for example, equivalent stress and the number of items to be sorted. The sort options print the sorted values in tabular form. The is designed for direct placement into reports. You may control the increment frequency of sorted values and the file unit to which these values are written.

The **PRINT VMASS** option allows you to selectively choose which elements and associated volumes, masses, and strain energy are to be printed. In order to have correct mass computations, mass density for each element must be given. The volumes, masses, and energies can be written on either standard output file unit 6, or user specified unit. The center of mass is also calculated and output.

User Subroutines

User subroutines can be used to obtain additional output, which can be accessed at each time/load increment. The available subroutines are the following:

- **IMPD** – obtains nodal quantities, such as displacements, coordinates, reaction forces, velocities, and accelerations
- **ELEVAR** – obtains element quantities such as strains, stresses, state variables, and cracking information
- **ELEVEC** – outputs element quantities during harmonic subincrements
- **INTCRD** – obtains the integration point coordinates used for forming the stiffness matrix

Restart

One capability of the **RESTART** option allows you to recover the output at increments where printout was suppressed in previous runs. This option can be used to print time/load increments for a number of consecutive increments. Under this option, MSC.Marc does not do any analysis.

In conjunction with the **RESTART** option, you can use either the **PRINT CHOICE** or **PRINT ELEMENT**, or **PRINT NODE** option to select a region of the model for which you want to obtain results.

Element Information

The main output from an increment includes element information followed by nodal information. The system provides the element data at each integration point. If you use the **CENTROID** parameter, the system provides the element data at the centroidal point. You can control the amount of printed output by using either the **PRINT CHOICE** or **PRINT ELEMENT** option. All quantities are total values at the current state (at the end of the current increment), and the physical components are printed for each tensor quantity (stress, strain, and generalized stress and strain). The orientation of these physical components is generally in the global coordinate system; however, their orientation depends on the element type. (*MSC.Marc Volume B: Element Library* explains the output for each particular element type.) Also, the physical components may be printed with respect to a user defined preferred system.

In addition to the physical components, certain invariants are given, as follows:

1. Tresca intensity - the maximum difference of the principal values, and the measure of intensity usually required for ASME code analysis
von Mises intensity - defined for stress as

$$\bar{\sigma} = \sqrt{\frac{3}{2} S_{ij} S_{ij}} \quad S_{ij} = \sigma_{ij} - \frac{1}{3} \delta_{ij} \sigma_{kk} \quad (12-1)$$

where σ is a stress tensor and S is the deviatoric stress tensor. For beam and truss elements, $\sigma_{22} = \sigma_{33} = 0$, and additional shears are zero; for plane stress elements, including plates and shells, $\sigma_{33} = 0$.

2. von Mises intensity - calculated for strain type quantities as

$$\bar{\epsilon} = \sqrt{\frac{2}{3} \epsilon_{ij} \epsilon_{ij}} \quad (12-2)$$

MSC.Marc uses these measures in the plasticity and creep constitutive theories. For example, incompressible metal creep and plasticity are based on the equivalent von Mises stress. For beam, truss, and plane stress elements, an incompressibility assumption is made regarding the noncalculated strain components.

- a. For beam and truss elements, this results in

$$\epsilon_{22} = \epsilon_{33} = -\frac{1}{2} \epsilon_{11} \quad (12-3)$$

b. For plane stress elements

$$\epsilon_{33} = -(\epsilon_{11} + \epsilon_{22}) \quad (12-4)$$

c. For beam and plane stress elements taken as a whole

$$\epsilon_{kk} = 0 \quad (12-5)$$

3. Mean normal intensity - calculated as

$$\begin{aligned} &(\sigma_{11} + \sigma_{22} + \sigma_{33})/3 \\ &\quad \text{or} \\ &(\epsilon_{11} + \epsilon_{22} + \epsilon_{33})/3 \end{aligned} \quad (12-6)$$

Equation 12-6 represents the negative hydrostatic pressure for stress quantities. For strain quantities, the equation gives the dilatational magnitude. This measurement is important in hydrostatically dependent theories (Mohr-Coulomb or extended von Mises materials), and for materials susceptible to void growth.

4. The principal values are calculated from the physical components. MSC.Marc solves the eigenvalue problem for the principal values using the Jacobi transformation method. Note that this is an iterative procedure and may give slightly different results from those obtained by solving the cubic equation exactly.
5. State variables are given at any point where they are nonzero.

Solid (Continuum) Elements

For solid (continuum) elements, stress, strain, and state variables are the only element quantities given. MSC.Marc prints out stresses and total strain for each integration point. In addition, it prints out thermal, plastic, creep, and cracking strains, if they are applicable. Note that the total strains include the thermal contribution.

Shell Elements

MSC.Marc prints generalized stresses and generalized total strains for each integration point. For thick shell (Type 22, 72, 75) elements interlaminar shear stresses are printed at the interface of two laminated layers.

The generalized stresses printed out for shell elements are

$$\frac{1}{t} \int_{-t/2}^{+t/2} \sigma_{ij} dy \quad (12-7)$$

(Section Force per Unit Thickness)

$$\frac{1}{t} \int_{-t/2}^{+t/2} y \sigma_{ij} dy \quad (12-8)$$

(Section moment per Unit Thickness)

The generalized strains printed are

$$E_{\alpha\beta}; \quad \alpha, \beta = 1, 2 \quad (12-9)$$

(Stretch)

$$\kappa_{\alpha\beta}; \quad \alpha, \beta = 1, 2 \quad (12-10)$$

(Curvature)

Physical stress values are output only for the extreme layers unless you invoke either [PRINT CHOICE](#) or [PRINT ELEMENT](#). In addition, thermal, plastic, creep, and cracking strains are printed for values at the layers, if applicable. Although the total strains are not output for the layers, you can calculate them using the following equations.

$$\epsilon_{11} = E_{11} + h\kappa_{11} \quad (12-11)$$

$$\epsilon_{22} = E_{22} + h\kappa_{22} \quad (12-12)$$

$$\gamma_{12} = E_{12} + h\kappa_{12} \quad (12-13)$$

where h is the directed distance from the midsurface to the layer; E_{ij} are the stretches; and κ_{ij} are the curvatures as printed. You can obtain these quantities by using the [ELEVAR](#) subroutine.

Beam Elements

The printout for beam elements is similar to shell elements, except that the section values are force, bending and torsion moment, and bimoment for open section beams. For beam element type 45, interlaminar shear stresses are printed at the interface of laminated layers if the **TSHEAR** parameter is used.

These values are given relative to the section axes (X, Y, Z) which are defined in the **COORDINATES** or the **GEOMETRY** option.

Before a beam member can be designed, it is necessary to understand the section forces distribution along the axial direction of the beam. For example, if variations of shear force and moment along axial direction are plotted, the graphs are termed shear diagram and moment diagram, respectively. To postprocess these diagrams, post code 261 needs to be specified along with the post codes (shown in Table 12-1) which define the corresponding section forces of the beam element.

Table 12-1 Post Codes for Beam Section Forces

Element Type	2-D/ 3-D	Post Code						
		264 (Axial Force)	265 (Moment xx or Moment ip)	266 (Moment yy or Moment op)	267 (Shear xz or Shear ip)	268 (Shear yz or Shear op)	269 (Torque)	270 (Bimoment)
5	2-D	X	X					
16	2-D	X	X					
45	2-D	X	X		X			
13	3-D	X	X	X			X	X
14	3-D	X	X	X			X	
25	3-D	X	X	X			X	
31	3-D	X	X	X	X	X	X	
52	3-D	X	X	X			X	
76	3-D	X	X	X			X	
77	3-D	X	X	X			X	
78	3-D	X	X	X			X	
79	3-D	X	X	X			X	
98	3-D	X	X	X	X	X	X	

Heat Transfer Elements

MSC.Marc prints the element temperatures and temperature gradients. In the coupled thermo-electric (Joule heating) analysis, the program also prints the element voltage, current, and heat fluxes.

Gap Elements

MSC.Marc prints the gap contact force, friction forces, and the amount of slip.

Linear and Nonlinear Springs

MSC.Marc prints the spring forces.

Hydrodynamic Bearing

MSC.Marc prints the pressure in the lubricant.

Nodal Information

MSC.Marc also prints out the following quantities at each nodal point.

Stress Analysis

- Incremental displacements - the amount of deformation that occurred in the last increment
- Total displacements - the summation of the incremental displacements
- Total equivalent nodal forces (distributed plus point loads) - the total force applied to the model through distributed loads (pressures) and point loads
- Reaction forces at fixed boundary conditions
- Residual loads at nodal points that are not fixed by boundary conditions

Note: Each value listed above is given at each nodal point, unless you invoke the [PRINT CHOICE](#) or [PRINT NODE](#) option. If you invoke the [TRANSFORMATION](#) option, MSC.Marc prints the nodal information relative to the user-defined system, rather than the global coordinate system.

Reaction Forces

MSC.Marc computes reaction forces based on the integration of element stresses. This is the only way to compute total reaction forces in a nonlinear analysis. Since such integration is only exact if the stresses are known at each integration point, the reaction forces are not printed if you use the [CENTROID](#) parameter.

In a nonlinear analysis, you should check that the reaction forces are in equilibrium with the external forces. If they are not in equilibrium, the analysis will be inaccurate, usually due to excessively large incremental steps. In most cases, the equilibrium is automatically ensured due to the convergence testing in MSC.Marc.

Residual Loads

The residual loads are a measure of the accuracy of the equilibrium in the system during analysis. This measure is very important in a nonlinear analysis and should be several orders of magnitude smaller than the reaction forces.

Dynamic Analysis

In a dynamic analysis, MSC.Marc prints out the total displacement, velocity, and acceleration at each time increment.

Heat Transfer Analysis

In a heat transfer analysis, MSC.Marc prints out nodal temperatures and nodal heat fluxes. If a thermo-electrical analysis is performed, MSC.Marc prints nodal voltages as well.

Rigid-Plastic Analysis

In steady-state rigid-plastic analysis, MSC.Marc prints nodal velocities.

Hydrodynamic Bearing Analysis

In a hydrodynamic bearing analysis, MSC.Marc gives the mass flux at the nodal points.

Electrostatic Analysis

In an electrostatic analysis, MSC.Marc prints the scalar potential and the charge.

Magnetostatic Analysis

In a magnetostatic analysis, MSC.Marc prints the vector potential and the current.

Electromagnetic Analysis

In an electromagnetic analysis, MSC.Marc prints out the vector and scalar potential, the current, and the charge.

Piezoelectric Analysis

In a piezoelectric analysis, MSC.Marc prints out displacements and the electric potential, and reaction forces and reaction charges.

Acoustic Analysis

In an acoustic analysis, MSC.Marc prints the pressure and the source.

Contact Analysis

The standard output includes, at the end of each increment, a summary of information regarding each body. This information reports the increment's rigid body velocity, the position of the center of rotation, and the total loads on the body. These last values are obtained by adding the contact forces of all nodes in contact with the rigid body in cause. Deformable bodies being in equilibrium have no load reported.

Additional information can be obtained by means of the **PRINT,5** parameter. In such cases, all the contact activity is reported. Namely, every time a new node touches a surface, or separates from a surface, a corresponding message is issued.

Contact with rigid surfaces entails an automatic transformation. Displacement increments and reactions in the transformed coordinates - tangent and normal to the contact interface - are also reported for every node that is in contact in the usual MSC.Marc manner.

Post File

The post file contains results of the analysis performed. This information can be viewed using MSC.Marc Mentat or MSC.Patran to observe the displaced mesh, contours of element quantities, principal quantities, time history behavior, response gradients, design variables, etc. The post file contains the finite element mesh and any changes to the finite element mesh due to either rezoning or adaptive mesh refinement. Basic nodal quantities (such as displacements, velocities, acceleration, applied loads, and reaction forces) are automatically placed on the post file. It is also possible to include user-defined nodal vectors by use of the [UPOSTV](#) user subroutine. Element quantities, such as stress, strain, plastic strain) must be explicitly requested by using the [POST](#) model definition option. It is possible to have user-defined element quantities placed on the post file for subsequent display by using the [PLOTV](#) user subroutine.

Forming Limit Parameter (FLP)

For continuum or shell elements, the Principal Engineering Strains can be calculated based on the true strain values. Correspondingly, the Forming Limit Parameter can be obtained for shell/membranes according to the data of the Forming Limit Diagram (FLD).

The principal engineering strains are calculated based on the true strain according to following relation:

$$e_i = \exp(\varepsilon_i) - 1 \quad (12-14)$$

where e_i are the principal engineering strain components and ε_i are the principal true strain components.

The principal engineering strain values can be selected for postprocessing. In addition, if shell/membrane elements are used, the Forming Limit Parameters (FLP) can also be selected. The *FLP* is defined as the ratio of the major principal engineering strain to the maximum allowable major principal engineering strain given by the Forming Limit Diagram. Based on this definition, it is calculated by

$$FLP = e_1 / FLD(e_2) \quad (12-15)$$

where e_1 means the major principal engineering strain, e_2 is the minor principal engineering strain. $FLD(e_2)$ is the forming limit corresponding to e_2 . *FLD* is defined by input data in model definition through material properties. By plotting the major principal engineering strain of each integration point of an element as point (e_2, e_1) on the chart, as shown in [Figure 12-1](#), it is easy to find that for points below the *FLD* line is safe. Strain rates above the *FLD* line imply failure. To consider the *FLP* value of these fields, one can find that in the field which is above *FLD* line, the value of *FLP* is larger than 1. On the *FLD* line, *FLP* equals 1, and *FLP* is less than 1 if a point located in the field below the *FLD* line. Therefore, by plotting the *FLP* value in MSC.Marc Mentat, it can be seen if the sheet metal forming process is successful.

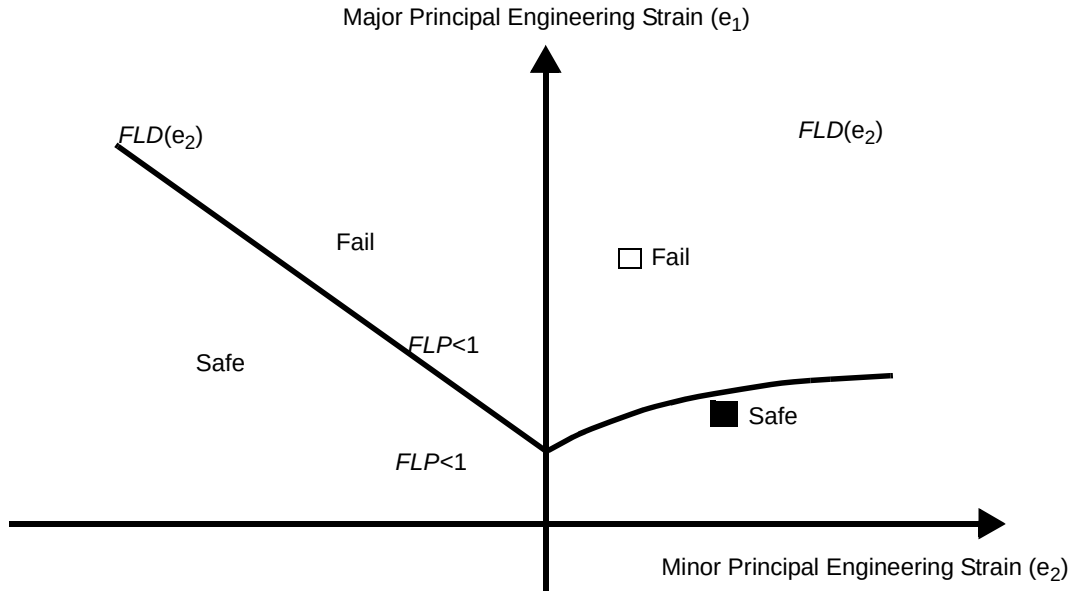


Figure 12-1 The Definition of Forming Limit Parameter (FLP)

The methods employed in MSC.Marc to define the Forming Limit Diagram are:

1. Fitted function definition
2. Predicted function definition
3. TABLE definition (accepts piecewise linear FLD curves)

The details of the above methods are described below:

1. *Fitted function:*

By this method, the polynomial functions are utilized to fit the FLD curve.

The functions are shown in Equations 12-16a and 12-9b:

$$FLD(e_2) = C_0 + D_1 e_2 + D_2 e_2^2 + D_3 e_2^3 + D_4 e_2^4 \quad (e_2 \leq 0) \quad (12-16a)$$

$$FLD(e_2) = C_0 + C_1 e_2 + C_2 e_2^2 + C_3 e_2^3 + C_4 e_2^4 \quad (e_2 \geq 0) \quad (12-9b)$$

2. *Predicted function:*

The predicted functions are generated based on the theories of local necking and diffuse necking. Both theories assume that the material obeys the power-law strain hardening, $\bar{\sigma} = K\bar{\epsilon}^n$.

Local necking: The critical strain for the onset of local necking is influenced by strain ratio ρ , where $\rho = \epsilon_2/\epsilon_1$, $\rho \leq 0$. Equation shows the relationship of critical strain versus strain ratio (by Hill).

$$\epsilon^* = \frac{n}{(1 + \rho)} \quad \rho \leq 0 \quad (12-10)$$

For uniaxial tension, $\rho = -0.5$. For plane strain, $\rho = 0$.

Diffuse necking: Diffuse necking is the phenomenon that while necking happens, deformation continues. The critical strain for the diffuse necking to happen is determined by Equation 12-11 (by Swift):

$$\epsilon^* = \frac{2n(1 + \rho + \rho^2)}{(1 + \rho)(2\rho^2 - \rho + 2)} \quad \rho \leq 0 \quad (12-11)$$

Note: For the case of $\rho < 0$, the predicted critical strain by Equation 12-11 is well below the experiment data. Therefore, for the purpose of prediction, the diffuse necking theory and the local necking theory are chosen for cases of $\rho > 0$ and $\rho \leq 0$, respectively.

Experiments (mainly with low carbon steels) showed that the predicted FLD usually is at a level below the experimental FLD curve. The experimental forming limit diagram is characterized by the value of ϵ^* corresponding to plane strain. This value is referred at FLD_0 as shown in Figure 12-2. This value increases with the strain-hardening exponent, n , and the strain-rate exponent, m . The value of ϵ^* is observed to rise as the thickness increases. This phenomenon is referred as thickness effect and it is characterized as thickness coefficient t_c . Experiments by Keeler tended to express this relationship as shown in Equation 12-12.

$$FLD_0 = Q \cdot (0.233 + t_c \cdot t) \quad (12-12)$$

where $Q = n/0.21$ if n is less than 0.21. Otherwise, $Q = 1.0$. T is the thickness of the sheet metal. The thickness coefficient t_c is set as 3.59 if the unit used to define the thickness is “Inch”. If the unit used is “mm”, t_c is set as 0.141. Similarly, if the unit “cm” is used, t_c is then set as 1.41, etc. It should be emphasized that, in most cases, friction is also a very important parameter that will affect the occurrence of necking. On the other hand, Equation 12-12 is mainly from data on low-carbon steels that have relatively high strain-rate dependence. In materials with low strain-rate dependence (for example, aluminum alloys), the thickness effect should be much less.

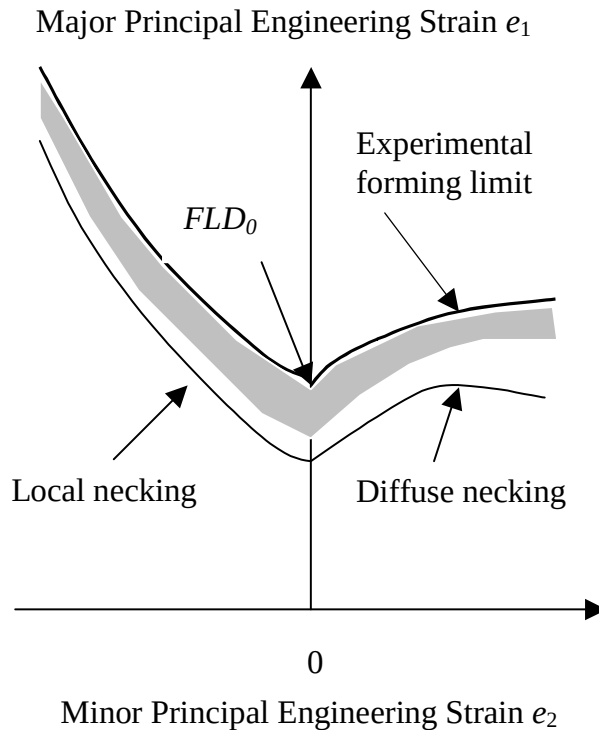


Figure 12-2 Predicted FLD and Experimental FLD Curves

Finally, it is suggested that for safety purposes, a safety zone, as shown as the shadow area in [Figure 12-2](#), must be set. For points above the safety zone, material is considered as failure. For points below the safety zone, material is considered as safe. If a point locates in the safety zone, the material is in the marginal status. Usually, the thickness of the safety zone is set as 0.1 or 10%.

3. TABLE definition:

The TABLE function in MSC.Marc allows users to define any curves through the [TABLE](#) model definition option. For example, if the user has FLD point of material, it is possible to define the FLD as piecewise linear curve. For details, refer the *MSC.Marc Volume C: Program Input*, Chapter 3 Model Definition Options, [TABLE](#).

Program Messages

The messages provided by MSC.Marc at various points in the output show the current status of the problem solution. Several of these messages are listed below.

- START OF INCREMENT x indicates the start of increment number x.
- START OF ASSEMBLY indicates MSC.Marc is about to enter the stiffness matrix assembly.
- START OF MATRIX SOLUTION indicates the start of the solution of the linear system.
- SINGULARITY RATIO prints yyy, where yyy is an indication of the conditioning of the matrix. This value is typically in the range 10^{-6} to 1. If yyy is of the order of machine accuracy (10^{-6} for most machines), the equations might be considered singular and the solution unreliable.
- END OF MATRIX SOLUTION indicates the end of matrix decomposition.
- END OF INCREMENT x indicates the completion of increment number x.
- RESTART DATA at INCREMENT x ON UNIT 8 indicates that the restart data for increment number x has been written (saved) on Unit 8.
- POST DATA at INCREMENT x on UNIT y indicates that the post data for increment number x has been written (saved) on Unit y.

In addition to these MSC.Marc messages, exit messages indicate normal and abnormal exists from MSC.Marc. [Table 12-2](#) shows the most common exit messages.

Table 12-2 MSC.Marc Exit Messages

Message	Meaning
MARC EXIT 3001	Normal exit.
MARC EXIT 3004	Normal exit.
MARC EXIT 13	Input data errors were detected by the program.
MARC EXIT 2004	Operator matrix (for example, stiffness matrix in stress analysis) has become non-positive definite and the analysis terminated.
MARC EXIT 3002	Convergence has not occurred within the allowable number of recycles.

MSC.Marc HyperMesh Results Interface

MSC.Marc now outputs results for postprocessing with HyperMesh. The writing of the HyperMesh results file is invoked by the [HYPERMESH](#) model definition option described in detail in *MSC.Marc Volume C: Program Input*. MSC.Marc Mentat can also be used to write the necessary commands into the MSC.Marc data file. The related button can be found one level below each of the ANALYSIS CLASS buttons within the JOBS menu. The binary file created as a result has the title `jobid.hmr` where `jobid` is the name of the data file for the job.

The geometry data may be imported into HyperMesh from a MSC.Marc data file. Alternatively, it can be created by HyperMesh or read in from other formats via the HyperMesh import capability.

The types of data that may be selected for writing into the HyperMesh results file are listed in *MSC.Marc Volume C: Program Input* under the [HYPERMESH](#) model definition option. It is important to note that, in case eigenvectors for buckling or eigenfrequency analysis are to be written into the results file, the corresponding MSC.Marc file should have the [BUCKLE INCREMENT](#) or [MODAL INCREMENT](#) model definition option, as appropriate. The [BUCKLE](#), [MODAL SHAPE](#), and [RECOVER](#) history definition options are not to be used in these cases.

Since HyperMesh allows only one deformed shape plot per simulation, each eigenvector of an eigenvalue analysis is saved as a separate simulation. Thus, when using HyperMesh, these eigenvectors can be plotted by skipping to the next simulation rather than skipping to the next data type of a simulation. The contour plots can be obtained for all data types including eigenvectors.

In case the number of requested eigenvalues is more than the number extracted, the data type in the HyperMesh “deformed” screen will inform you for those modes that are not extracted. The “next” button may need to be clicked to see the data type in the “deformed” mode of plotting.

MSC.Marc SDRC I-DEAS Results Interface

A facility exists in MSC.Marc to write out an SDRC I-DEAS universal file of the analysis results, in order for postprocessing by the I-DEAS program. The writing of the universal file is invoked by the [SDRC](#) model definition option described in detail in *MSC.Marc Volume C: Program Input*. MSC.Marc Mentat can also be used to write the necessary commands into the MSC.Marc data file. The related button can be found one level below each of the ANALYSIS CLASS buttons within the JOBS menu. The formatted ASCII file created as a result has the title `jobid.unv`, where `jobid` is the name of the data file for the job.

The model data is imported into I-DEAS by way of the universal file as well.

The types of data that may be selected for writing into the I-DEAS universal file are listed in *MSC.Marc Volume C: Program Input* under the [SDRC](#) model definition option.

MSC.Marc ADAMS Results Interface

The MSC.Marc ADAMS results interface allows a finite element model from MSC.Marc to be transferred to MSC.ADAMS/Flex. A modal analysis is first performed in MSC.Marc. Then, the interface program converts the results into a modal neutral file (MNF), which is used by MSC.ADAMS/Flex as the foundation for a flexible body.

The following describes the necessary steps needed for using this feature.

The MSC.Marc model should not have any loading or kinematic boundary conditions. A modal analysis using the Lanczos method should be performed. This should be the first and only loadcase in the model. The post file must be saved with a post revision equal to or greater than 9 (which includes the default in the current release).

Before the job is started, [SUPERELEM](#) model definition option must be inserted into the MSC.Marc input file (jobname.dat):

First card series:

SUPERELEM

Second card series:

0,0,1

Third card series:

list of degrees of freedom to be exported to adams

Fourth card series:

list of nodes to be exported to adams

the third and fourth card series can be repeated if needed

Example:

```
superelem
```

```
0,0,1
```

```
1,2,3,4,5,6
```

```
1,16
```

```
1,2,3,4,5,6
```

```
321,336
```

When the job is run successfully, it gives the exit number 3018 and a post file is written. This post file contains the information that is converted into the MNF file. To convert the model into the MNF, the user needs to run the program *marctoadams* located in the bin directory of the MSC.Marc installation. This program starts by asking some questions about the model with suggested options where applicable:

Type of post file to read (binary or formatted)

Name of post file to read

Model units

The model units given should be the ones used in the MSC.Marc model. This allows MSC.ADAMS to interpret the MNF correctly. If the *marctoadams* program runs successfully, you will obtain a file called `filename.mnf`, where `filename` is the model name of the post file being used.

The current version does not support preloads, modal loads, or modal stresses or strains.

Status File

The status file contains summarized information of the analysis. Each line within this file includes the load case number, the associated increment number, and the number of cycles, contact separations, and cutbacks associated with that increment; the accumulated total cycles, increment splittings, separations, cutbacks, and remeshing times numbers in the analysis as well as the time step size of each increment and the overall time achieved by the analysis. MSC.Marc reports all this information on one line upon completion of each increment. If the increment is partially completed (for example, the completion of one part of a split increment or one part of the whole increment due to cutback), MSC.Marc also reports one line for each part of the increment.

CHAPTER 13

Parallel Processing

- Different Types of Machines
- Supported and Unsupported Features
- Matrix Solvers
- Contact
- Domain Decomposition

MSC.Marc can make use of multiple processors when performing an analysis. Most, but not all, features of MSC.Marc are supported in parallel mode. The type of parallelism used is based upon *domain decomposition*. A commonly used name for this is the Domain Decomposition Method (DDM). The model is decomposed into domains of elements, where each element is part of one and only one domain. The nodes which are located on domain boundaries are duplicated in all domains at the boundary. These nodes are referred to as inter-domain nodes below. The total number of elements is thus the same as in a serial (nonparallel) run but the total number of nodes can be larger. The computations in each domain are done by separate processes

on the machine used. At various stages of the analysis, the processes need to communicate data between each other. This is handled by means of a communication protocol called MPI (Message Passing Interface). MPI is a standard for how this communication is to be done and MSC.Marc makes use of different implementations of MPI on different platforms. MSC.Marc uses MPI regardless of the type of machine used. The types of machines supported are shared memory machines, which are single machines with multiple processors and a memory which is shared between the processors, and cluster of separate workstations connected with some network. Each machine (node) of a cluster can also be a multiprocessor machine.

Different Types of Machines

As mentioned above, MSC.Marc can run on a shared memory machine and on a cluster of workstations. The main reason for running a job in parallel on a shared memory machine is speed. Since all processes run on the same machine sharing the same memory, the processes all compete for the same memory. There is an overhead in memory usage so some parts of the analysis need more memory for a parallel run than a serial job. The matrix solver, on the other hand, needs less memory in a parallel job. Less memory is usually needed to store and solve several smaller systems than one large.

In the case of a cluster, the picture is somewhat different. Suppose a number of workstations are used in a run and one process is running on each workstation. The process then has full access to the memory of the workstation. If a job does not fit into the memory of one workstation, the job could be run on, say, two workstations and the combined memory of the machines may be sufficient.

The amount of speed-up that can be achieved depends on a number of factors including the type of analysis, the type of machine used, the size of the problem, and the performance of communications. For instance, a shared memory machine usually has faster communication than a cluster (for example, communicating over a standard Ethernet). On the other hand, a shared memory machine may run slower if it is used near its memory capacity due to memory access conflicts and cache misses etc.

Supported and Unsupported Features

Most of the main features of MSC.Marc are supported in parallel mode. For instance, a fully coupled thermomechanical contact analysis can be performed in parallel. A notable feature which is not yet supported is automatic remeshing.

The list of unsupported features is

- Acoustics
- Auto therm creep
- Beam-to-beam contact
- Bearing
- Buckling
- Convective terms in heat transfer
- Design sensitivity and optimization
- Eigenvalue extraction
- Electromagnetics
- Explicit dynamics
- Fluids and its coupled analysis
- Gap elements (but the normal contact is supported)
- Harmonics
- Hydrodynamics
- Insert
- J-integral extraction
- Radiation
- Remeshing and rezoning
- Response spectrum
- Steady state rolling analysis
- Superelements

Matrix Solvers

All matrix solvers in MSC.Marc are supported in parallel mode with the exception of the unsymmetric solver. Only the multifrontal solver (solver type 8) supports out-of-core solution in parallel. The matrix solvers work somewhat different in parallel mode as compared to serial mode.

For the direct solvers, a two-stage approach is used. In a mechanical analysis, all inter-domain nodes are first given fixed displacements and each domain solves the resulting equation system independently. The second stage operates on the interdomain nodes to relax the extra imposed fixed displacements. This stage is not necessarily if there are no connections between the meshes in the different domains. An iterative procedure based upon the conjugate gradient method is performed for this stage. This

procedure operates as mentioned on the inter-domain nodes. The number of conjugate gradient iterations needed to converge to the final solution is reported in the MSC.Marc output file as “nn inter-domain iterations”.

The conjugate gradient iterative solver (solver type 2) operates simultaneously on the whole model. It works to a large extent like in a serial run. For each iteration cycle, there is a need to synchronize the residuals from the different domains.

The third type of solver supported is the hardware solver (HP, Sun, and SGI). These solvers are parallelized themselves using multithreading. They do not support runs over a network so they are only available on shared memory machines. For this case, the global stiffness matrix (or the corresponding matrix for nonstructural analysis) is gathered to the parent process (where the job was started). The original child processes are put to sleep and the solver creates new processes (multiple threads) to solve the equation system in parallel. When the solution is finished, the child processes are awakened and the data related to each domain are distributed.

Contact

Contact analysis poses a special challenge for parallelization due to its global nature. The current version of MSC.Marc fully supports contact analysis including thermal and electrical contact but excluding beam-to-beam contact. One restriction is specific to contact analysis in parallel: the nodal numbering must be consecutive.

In order to enable a fast contact search procedure, some contact data are duplicated across all domains. A node in one domain must be able to find contact with a segment in a different domain. The duplicated data includes nodes on the exterior boundary of deformable contact bodies. This data duplication is not necessary in a pure rigid contact job, so it is important to define a contact table to indicate that no deformable contact can take place in this case. The default, if no contact table is used, is that all deformable bodies check for contact with all other bodies including itself.

If a node touches a segment which is located in a different domain, the nodes making up the segment are imported into the domain of the touching node. If the node separates, the nodes are deleted from the domain. This means that the number of nodes in a domain can dynamically change. This procedure allows the creation of tyings spanning domains. This also has implications on how the domains should be created. One should avoid having contact bodies with a large number of nodes coming into contact across different domains. An extreme case would be two sheets of equal number of shell elements coming into contact on top of each other. If two domains are used and each sheet is in a separate domain, the number of nodes in one domain would double.

While a parallel run should, in general, produce exactly the same results as the corresponding serial run (apart from differences in round-off errors), there is an exception in contact analysis including self contact. The way the contact search algorithm works, the setting up of contact tying relations depends on the nodal numbering. Since the nodal numbering may be different in a parallel run as compared to a serial run (depending on the decomposition), the contact tying settings could be different and the results may vary. A serial run with self contact can also be made to behave differently by using directed renumbering.

Domain Decomposition

The way the model is decomposed into domains, in general, affects the performance of the parallel run. As mentioned above, this decomposition is, in this release, performed in the preprocessor stage. Both MSC.Mentat and MSC.Patran use Metis software for creating the domain decomposition. The general goal in creating a good decomposition is to minimize the number of inter-domain nodes. Keeping this number low minimizes the overhead in having duplicate nodes and also minimizes the size of the system solved in the interdomain part of the direct solver. It is important to get a good balance between the workload in the different domains if, for instance, different types of elements are used or if the nodes of a cluster have different performance.

Another aspect of deformable contact as mentioned above is to avoid having a large number of nodes touching segments of a different domain.

Some different methods for creating the decomposition are available in MSC.Mentat and MSC.Patran. One should note that some of them are based on element connectivity so they tend to put separate contact bodies in separate domains which is not always optimal.

The optimal number of domains to use for a given job is pretty hard to estimate. It obviously depends on the number of processors available. There is, in general, an optimal number of processors to use for a given job and a given machine or cluster of machines. A general rule is that the larger the model is, the more it pays off to use more processors.

Running a Parallel Job

Running a MSC.Marc parallel job is not much different from running a serial job. In the current release of MSC.Marc, the model is decomposed into domains by the preprocessor (MSC.Marc Mentat or MSC.Patran) and a separate input file is created for each domain. From a user's perspective, the extra steps consist of activating the procedure for decomposing the model and activating the parallel run. For the case of a run on a cluster, one would additionally have to define which machines make up the

cluster (unless an automatic cluster tool is used). The run is started as usual and separate results files are created for each domain. The postprocessor automatically picks up the results from the different domains and combines them into a single model which looks the same as in a serial run. One can also choose to look at the results of each domain separately, which is particularly useful for large jobs.

When running a parallel job from MSC.Marc Mentat or MSC.Patran, one has to activate the button for parallel and start the job as for a serial job. From the command line, one needs to use an extra argument to specify the number of domains:

```
run_marc -v n -b n -j jobname -np 5
```

for running a job with five domains. User subroutines are handled in the same way as in a serial run. See *MSC.Marc Volume D: User Subroutines and Special Routines* for a description on special considerations for user subroutines in a parallel run.

Running on a cluster (a set of connected computers) requires that the machines communicate properly. The actual connection could be anything from a telephone modem up to a high speed direct connected cluster. A standard 100 Mbit Ethernet network is usually sufficient for reasonable performance, although it can become bottlenecked with fast processors. Faster connections give better performance on the communication part.

In order to perform an analysis over a network, one needs to specify which machines are to be part of the analysis. This is done through a so-called host file, which is generic for a MSC.Marc run. The MSC.Marc startup script converts it into a format that the current MPI implementation expects. The host file is a simple text file that lists the computers that are to be used in the run and some info about the run. The host file typically looks something like (using a Unix system)

```
host1 2
host2 1 /users/joe/run /programs/marc2003
host3 2 /usr/people/joe /programs/marc/marc2003
```

Here host1, host2, and host3 are the host names of the respective machines. It specifies a job with five domains where two processes are run on host1 and host3 and one on host2. /users/joe/run is the directory where the job is run on host2. This is where the input files should be and where the results files are created. This directory could be a local directory on a hard drive on host2 or it could be a shared directory (via NFS for example). This directory must be accessible from host2 (but not necessarily from the other hosts). The fourth entry of the host file specifies where the MSC.Marc installation directory is located. This can also be a local directory (if MSC.Marc is installed on the local machine) or it can point to a shared directory. It is, in general, recommended that the run directory is a local directory to avoid I/O traffic

over the network. Please note that not all MPI implementations currently supported uses the fourth entry. This is for instance true for MP-MPICH that is used on Windows systems. Here the MSC.Marc installation directory must be shared.

If local run directories are used, one must make sure that the input files are available in the run directory of each host. This can be done by means of a built-in file copy program that is automatically launched when the MSC.Marc job is started. Before the analysis starts, the necessary input files are copied over to the respective host. The file copy program uses the same MPI communication as the main MSC.Marc program. When the analysis is finished, the file copy program copies the post files back to the directory on `host1` where the job was started. Please note that only the input files and the post files (not output files, etc.) are copied. This copying can be suppressed by buttons in MSC.Mentat and MSC.Patran or by a command line option if MSC.Marc is manually started. If a user subroutine is used together with local run directories, the new executable automatically transfers to all hosts before the MSC.Marc job starts.

One can also set up MSC.Marc to run via external cluster tools. A useful run time option in this context is the command line option `-dir`. It specifies the run directory for the job and is where the scratch files and results files are created. Suppose MSC.Marc is set up to run on a cluster using some kind of cluster tool to distribute the processes and each machine in the cluster has a local disk called `/scratch`. By using the option `-dir /scratch` from the command line, the run uses the local directory during the run. The input files are still read from where the job is started (which could be a shared directory on a control workstation which is not even part of the run).

The following command line options related to parallel processing are available for the MSC.Marc start-up script `run_marc`:

<code>-np nn</code>	specify <code>nn</code> number of processes.
<code>-ci y n</code>	if <code>y</code> copy input files to local run directories; if <code>n</code> , do not.
<code>-cr y n</code>	if <code>y</code> copy post files from local run directories; if <code>n</code> , do not.
<code>-ho hostfile</code>	specify the host file that describes the hosts to use in a network run.
<code>-dir rundir</code>	specify that the run directory of the run is <code>rundir</code> .

APPENDIX

A

Finite Element Technology in MSC.Marc

- Governing Equations of Various Structural Procedures
- System and Element Stiffness Matrices
- Load Vectors
- References

Governing Equations of Various Structural Procedures

This section describes the basic concepts of finite element technology. For more information, you are urged to refer to the finite element text [\[Ref. 5\]](#). MSC.Marc was developed on the basis of the displacement method. The stiffness methodology used in MSC.Marc addresses force-displacement relations through the stiffness of the system. The force-displacement relation for a linear static problem can be expressed as

$$Ku = f \quad (A-1)$$

where K is the system stiffness matrix, u is the nodal displacement, and f is the force vector.

Assuming that the structure has prescribed boundary conditions both in displacements and forces, the governing Equation A-1 can be written as

$$\begin{bmatrix} K_{11} & K_{12} \\ K_{21} & K_{22} \end{bmatrix} \begin{Bmatrix} u_1 \\ u_2 \end{Bmatrix} = \begin{Bmatrix} f_1 \\ f_2 \end{Bmatrix} \quad (\text{A-2})$$

u_1 is the unknown displacement vector, f_1 is the prescribed force vector, u_2 is the prescribed displacement vector, and f_2 is the reaction force. After solving for the displacement vector u , the strains in each element can be calculated from the strain-displacement relation in terms of element nodal displacement as

$$\varepsilon_{el} = \beta u_{el} \quad (\text{A-3})$$

The stresses in the element are obtained from the stress-strain relations as

$$\sigma_{el} = L \varepsilon_{el} \quad (\text{A-4})$$

where σ_{el} and ε_{el} are stresses and strains in the elements, and u_{el} is the displacement vector associated with the element nodal points; β and L are strain-displacement and stress-strain relations, respectively.

In a dynamic problem, the effects of mass and damping must be included in the system. The equation governing a linear dynamic system is

$$M\ddot{u} + D\dot{u} + Ku = f \quad (\text{A-5})$$

where M is the system mass matrix, D is the damping matrix, Equation A-6 is the acceleration vector, and u is the velocity vector. The equation governing an undamped dynamic system is

$$M\ddot{u} + Ku = f \quad (\text{A-6})$$

The equation governing undamped free vibration is

$$M\ddot{u} + Ku = 0 \quad (\text{A-7})$$

Natural frequencies and modal shapes of the structural system are calculated using this equation.

$$K\phi - \omega^2 M\phi = 0 \quad (\text{A-8})$$

The equations governing some other procedures are similar. For example, the governing equation of transient heat transfer analysis is

$$C^T \dot{T} + K^T T = Q \quad (\text{A-9})$$

where C^T is the heat capacity matrix, K^T is the thermal conductivity matrix, Q is the thermal load vector (flux), T is the nodal temperature vector, and t is the time derivative of the temperature. Equation A-9 reduces to

$$K^T T = Q \quad (\text{A-10})$$

for the steady-state problem. Note that the equation governing steady-state heat transfer (Equation A-10) and the equation of static stress analysis (Equation A-1) take the same form. Similarly, the hydrodynamic bearing problem is analogous to a steady-state heat transfer problem. This problem is governed by an equation similar to Equation A-10.

The matrix equation of the electrical problem in coupled thermo-electrical analysis is

$$K^E(T)V = I \quad (\text{A-11})$$

The equation governing the thermal problem is:

$$C^T(T)\dot{T} + K^T(T)T = Q + Q^E \quad (\text{A-12})$$

in both Equation A-11 and Equation A-12, v is the voltage, $K^E(T)$ is the temperature-dependent electrical-conductivity matrix, I is the nodal-current vector, $C^T(T)$ is the temperature-dependent, heat-capacity matrix, and $K^T(T)$ is the thermal-conductivity matrix. T is the nodal temperature vector, Q is the heat-flux vector, and Q^E is the internal heat-generation vector that results from the electrical current. Electrical and thermal problems are coupled through $K^E(T)$ and Q^E .

The matrix equations for the thermal-mechanical problem are as follows:

$$M\ddot{u} + D\dot{u} + K(T,u,t)u = F \quad (\text{A-13})$$

$$C^T(T)\dot{T} + K^T(T)T = Q + Q^I + Q^F \quad (\text{A-14})$$

In [Equation A-13](#) and [Equation A-14](#) the damping matrix D , stiffness matrix K , heat-capacity matrix C^T and thermal-conductivity matrix K^T are all dependent on temperature. Q^I is the internal heat generated due to inelastic deformation. The coupling between the heat transfer problem and the mechanical problem is due to the temperature-dependent mechanical properties and the internal heat generated. If an updated Lagrangian analysis is performed, K and K^T are dependent upon prior displacement.

The governing equations described above are either sets of algebraic equations ([Equation A-1](#), [Equation A-10](#), and [Equation A-11](#)), or sets of ordinary differential equations ([Equation A-5](#) through [Equation A-8](#) and all equations through [Equation A-4](#)). The time variable is a continuous variable for the ordinary differential equations. Select an integration operator (for example, Newmark-beta, Houbolt, or central difference for dynamic problems, and backward difference for heat transfer) to reduce the set of differential equations to a set of algebraic equations. The final form of governing equations of all analysis procedures is, therefore, a set of algebraic equations.

System and Element Stiffness Matrices

The previous section presented system matrices assembled from element matrices in the system. For example, the system stiffness matrix K is expressed in terms of the element stiffness matrix K_i^{el} as

$$K = \sum_{i=1}^N K_i^{el} \quad (A-15)$$

where N is the number of elements in the system. The system stiffness matrix is a symmetric-banded matrix. See [Figure A-1](#) for a schematic of the assemblage of element matrices. When the element-by-element iterative solver is used, the total stiffness matrix is never assembled.

The element stiffness matrix can be expressed as

$$K^{el} = \int_{v^{el}} \beta^T L \beta dv^{el} \quad (A-16)$$

where v^{el} is the volume of the element, β is the strain-displacement relation, and L is the stress-strain relation.

$$\varepsilon = \beta u \quad (A-17)$$

$$\sigma = L \varepsilon \quad (A-18)$$

The mass matrix can be expressed as

$$M^{el} = \int_{v^{el}} N^T \rho N dv \quad (A-19)$$

The integration in [Equation A-16](#) is carried out numerically in MSC.Marc, and is dependent on the selection of integration points. The element stiffness matrix can be fully or under integrated. The mass matrix is always fully integrated.

Note: The β matrix is directly associated with the shape functions and geometry of each element. Shape functions associated with different element types are explained in [MSC.Marc Volume B: Element Library](#). Stress-strain relations L are discussed in [Chapter 7](#) of this manual.

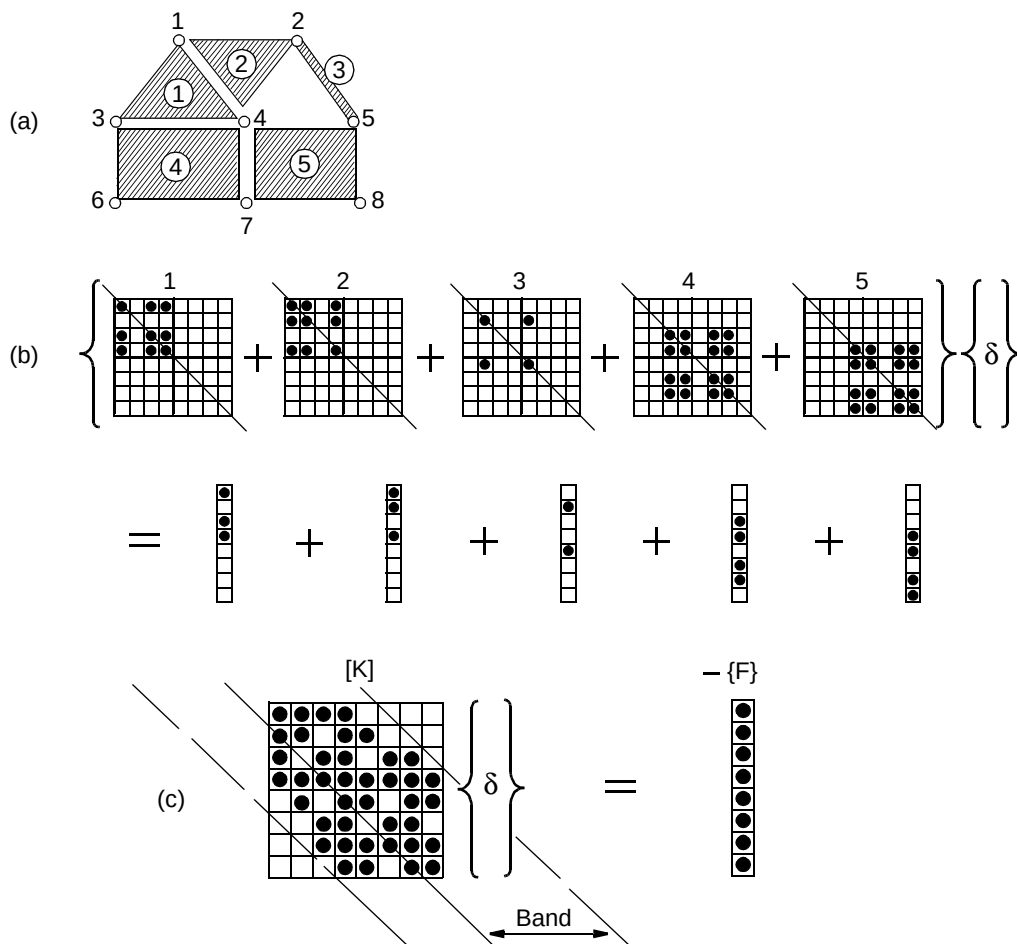


Figure A-1 Schematic of Matrix Assemblage

Load Vectors

The nodal force vector f in [Equation A-1](#) includes the contributions of various types of loading.

$$f = f_{point} + f_{surface} + f_{body} + f^* \quad (A-20)$$

where f_{point} is the point load vector, $f_{surface}$ is the surface load vector, f_{body} is the body (volumetric) load vector, and f^* represents all other types of load vectors (for example, thermal and creep strains, and initial stress).

The point load is associated with nodal degrees-of-freedom and can be added to the nodal force vector directly. Equivalent nodal force vectors $f_{surface}$, f_{body} must be calculated from the distributed (surface/volumetric) load first and then added to the nodal force vector. In MSC.Marc, the computation of equivalent nodal forces is carried out through numerical integration of the distributed load over the surface area of volume to which the load is applied. This can be expressed as

$$f_{surface} = \int_A N^T p dA \quad (A-21)$$

$$f_{body} = \int_V N^T p dV \quad (A-22)$$

where p is the pressure.

[Figure A-1](#) shows the assemblage of the nodal force vector.

References

1. Zienkiewicz, O. C. and R. L. Taylor. *The Finite Element Method (4th ed.) Vol. 1. Basic Formulation and Linear Problems (1989), Vol. 2. Solid and Fluid Mechanics, Dynamics, and Nonlinearity (1991)* McGraw-Hill Book Co., London, U. K.
2. Bathe, K. J. *Finite Element Procedures*, Prentice-Hall, Englewood Cliffs, NJ, 1995.
3. Hughes, T. J. R. *The Finite Element Method—Linear Static and Dynamic Finite Element Analysis*, Prentice-Hall, Englewood Cliffs, NJ. 1987.
4. Ogden, R. W. “Large Deformation Isotropic Elasticity: On The Correlation of Theory and Experiment for Incompressible Rubberlike Solids,” *Proceedings of the Royal Society*, Vol. A (326), pp. 565-584, 1972.
5. Cook, R. D., D. S. Malkus, and M. E. Plesha, *Concepts and Applications of Finite Element Analysis* (3rd ed.), John Wiley & Sons, New York, NY, 1989.

APPENDIX

B

Finite Element Analysis of NC Machining Processes

- General Description
- NC Files (Cutter Shape and Cutter Path Definition)
- Intersection Between Finite Element Mesh and Cutter
- Deactivation of Elements
- References

General Description

Machining (or Metal Cutting) is a type of material removal processes widely used to produce a final part of the desired geometry. After removal of the machined material, re-establishment of equilibrium within the remaining part of the structure causes some distortion due to the relief of the residual stresses in the machined materials.

Depending on the stress level and geometry of the part, the severe distortion due to machining can result in high scrap rates and increased manufacturing costs, especially for structures with thin wall or large thin plates.

A numerical analysis procedure has been developed in MSC.Marc to predict the distortion during the 3-D bulk machining so that engineers can optimize their structure design and machining processes. This procedure is based on the assumption that effects introduced by the cutting process are local and small compared to those due to pre-existing residual stresses.

The simulation procedure uses the cutter shape and path information obtained from NC machining data. Cutter motion information is used to predict the material to be removed by automatically deleting finite elements that are located within the cutter path. The distortion of the remaining part can be predicted by re-calculating the equilibrium. An interface has been developed in MSC.Marc to convert cutter path data into the finite elements to be removed. The cutter path data is stored in either the Automatically Programmed Tools (APT) source or the Cutter Location (CL) data format. The APT source is the NC data output by CAD software CATIA. The CL data is the cutter location data provided by APT compilers.

For details on how to use this functionality, refer to *MSC.Marc Volume C: Program Input* for the [MACHINING](#) parameter and the [DEACTIVATE](#) model definition option and the [DEACTIVATE](#) history definition option.

NC Files (Cutter Shape and Cutter Path Definition)

The cutter shape and the cutter path are the key parameters to determining the volume of the material to be cut. The cutter shape is defined by the CUTTER statement in the APT source or CL files. The expanded form of the CUTTER statement is as follows:

CUTTER/d,r,e,f,α,β,h

where the parameters have the following meaning as shown in [Figure B-1](#).

- d* The tool diameter.
- r* The radius of the corner circle; it can be zero or larger than $d/2$.
- e* The radial distance from the tool axis to the center of the corner circle.
- f* The distance from the tool endpoint to the center of the corner circle measured parallel with the tool axis.
- α The angle from a radial line through the tool endpoint to the lower line segment; it is in range of zero to 90°, and positive.

- β The angle between the upper segment and the tool axis; it is in the ranges of -90° to 90° .
- h The cutter height measured from the tool endpoint along the tool axis.

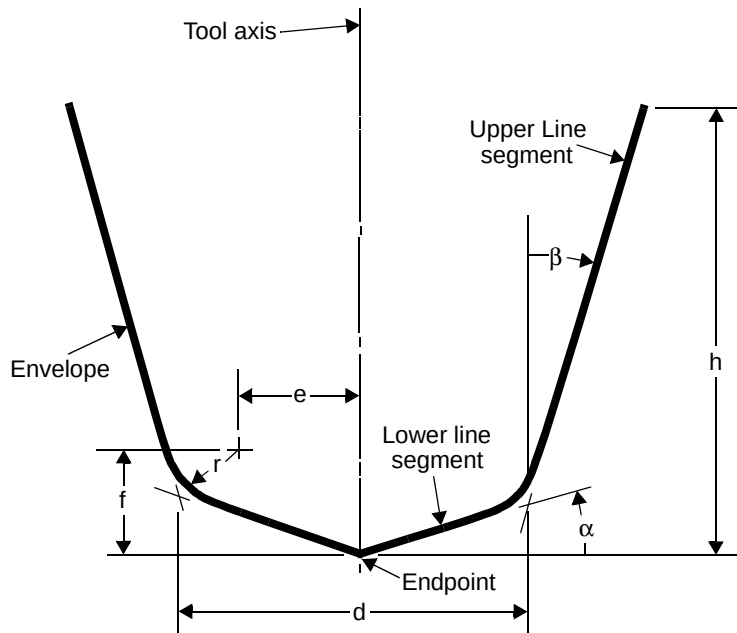


Figure B-1 Definition of Cutter Geometry

The cutter path is calculated based on the motion of the cutter and cutter shape. The motion is defined by cutter motion statements like GOTO/, GODLTA/, CYCLE/, and DRILL/, etc. In MSC.Marc, these motion types are translated into point-to-point, circular, or drilling motion modes, respectively. For details, refer to references by Irvin H. Kral and Dassault Systems.

- Notes:**
1. MSC.Marc accepts APT data files written by CATIA V4 and the corresponding CL data files by APT compilers.
 2. The cutter can be of either multi-axis or fixed axis.
 3. Two cutter motion types: Point-To-Point (for example, GOTO/ and GODLTA/ statements) and Cycle (DRILL/...) are directly supported. Circular motions (either in-plane or out-of-plane) can be interpolated into Point-To-Point motions. The interpolation can be done inside CATIA by choosing the proper option when writing out APT source file.

4. In addition, some statements in the APT/CL files are skipped. If the user uses statements like TRACUT/, COPY/, /MACRO, or /CALL to define cutter motions, they must be converted into explicit motion statements like Point-to-Point or Cycle (DRILL/) motions before writing out the APT data file.
5. It is assumed that the part and cutter are defined in the same coordinate system. So, if a part needs to be flipped over between two machining stages, the user can rotate the cutter but keep the part unchanged in space.
6. If the user wants to change the boundary conditions for some nodes during the cutting process defined by one APT/CL file, the cutting process must be split into multiple processes. Each process, when split, has its own APT/CL file. This requires each process to have its own set of boundary conditions.

Intersection Between Finite Element Mesh and Cutter

When the cutter moves, its path forms the volume of the material to be cut. This volume is calculated based on the cutter geometry and its path in space. The finite element mesh is checked against this volume to decide the intersection between the volume and the finite element nodes and elements.

Deactivation of Elements

The automatic deactivation of finite elements depends on the intersection status between the cutter path and the mesh. The approach is designed for elements that are fully or partially located inside the cut volume. Elements that are fully located inside the cut volume are automatically deactivated and the stress that existed before deactivation is released. For elements that are partially intersected by the cut volume, integration points are checked. In the current version, the check of integration points is limited to the element centroid only. If element centroid is found inside the cut volume, the associated element is deactivated. For deactivated elements, the stresses/strains are set to zero permanently.

After the deactivation of elements along the motion of cutter, the part will distort progressively. The updated geometry of the part is used for the next cutting step. Using this method, the whole machining process is simulated with MSC.Marc. After the machining is finished, the final distortion of the work piece can be calculated by spring back analysis. The spring back analysis is conducted by removing all fixturing restraints except for those needed to remove rigid body motion.

References

1. Irvin H. Kral, *Numerical Control Programming in APT*, Prentice-Hall, 1986.
2. Dassault Systems, *CATIA Training Guide*, Version 4, Release 1.6, April, 1996.
3. Houtzeel Manufacturing systems, Inc., *APT Language Reference Manual*, March, 1997.

I N D E X

MSC.Marc Volume A: Theory and User Information

A B C D E F G H I J K L M N O P Q R S T U V W X Y Z

A

acoustic

- analysis 6-49, 7-160, 9-19, 10-24, 12-19
- element types 6-49, 6-55
- input options 9-19

ACOUSTIC (parameter) 6-50

ACTIVATE (history definition option) 4-7

ADAPT GLOBAL (model definition option) 4-41

ADAPTIVE (model definition option) 4-38, 5-4, 5-5, 10-43

ADAPTIVE (parameter) 8-40, 8-41, 10-43

adaptive mesh

- contact 8-41
- refinement 10-43

adaptive time control 5-53

ALL POINTS (parameter) 6-22, 10-3

analysis 6-76, 7-158

- acoustic 6-49, 7-160, 9-19, 10-24, 12-19
- contact 12-19
- coupled 6-67, 8-31
- coupled acoustic-structural 6-51
- coupled contact 8-31
- coupled thermo-electrical (Joule heating) 6-73
- coupled thermo-mechanical 6-69
- crack 5-61
- creep 11-5
- design sensitivity 5-102
- dynamic 12-18
- eigenvalue 5-70
- electrical 9-18
- electromagnetic 6-39, 7-159, 9-20, 10-26, 12-19
- electrostatic 7-159, 9-18, 10-24, 12-19

error 4-33

fluid 6-56, 7-160, 10-28

fluid/solid interaction 6-71, 7-159, 10-24

Fourier 5-6

harmonic 6-42

heat transfer 6-3, 7-158, 9-16, 12-19

hydrodynamic bearing 6-25, 7-158, 9-17, 10-25, 12-19

large strain 10-32, 10-33

linear 5-3

magnetostatic 6-33, 7-159, 10-25, 12-19

nonlinear 5-10

perturbation 5-31

piezoelectric 6-44, 9-19, 10-29, 12-19

post buckling 11-5

rigid cavity acoustic 6-49

rigid-plastic 5-90, 10-33, 12-19

snap-through 11-5

soil 5-97, 7-136, 10-27

steady state rolling 5-118

stress 12-18

thermal contact 8-34

thermo-electrical 7-158

analytical contact 8-49

ANELAS (user subroutine) 5-4, 7-5, 7-8

ANEXP (user subroutine) 5-4, 7-8, 7-129

ANISOTROPIC (model definition option) 5-4, 6-44, 7-5, 7-7, 7-66, 7-83

anisotropic viscoelastic material 7-112

ANKOND (user subroutine) 6-4, 7-158

ANPLAS (user subroutine) 7-8, 7-66

application phase 5-69

APPROACH (history definition option) 8-12

A B C D E F G H I J K L M N O P Q R S T U V W X Y Z

Arbitrary Eularian-Lagrangian (AEL)
 formulation 5-27
 arc-length methods 11-27
 ASSUMED STRAIN (parameter) 10-40
 assumed strain formulation 10-40
 ATTACH EDGE (model definition option) 4-36
 AUTO CREEP (history definition option) 5-53,
 5-55, 5-58, 5-59, 5-60, 7-107, 11-4, 11-5, 11-6
 AUTO INCREMENT (history definition option)
 4-27, 5-30, 7-135, 9-9, 11-4, 11-5, 11-6, 11-7,
 11-9
 AUTO INCREMENT (model definition option)
 11-31
 AUTO LOAD (history definition option) 4-27,
 7-119, 9-9, 11-4, 11-6
 AUTO STEP (history definition option) 5-76, 6-6,
 6-10, 7-112, 7-119, 7-135, 8-49, 8-61, 11-4,
 11-5, 11-6, 11-9, 11-11, 11-12, 11-13, 11-14,
 11-15
 AUTO THERM (history definition option) 9-14,
 11-4, 11-6
 AUTO THERM CREEP (history definition option)
 5-57, 5-58, 11-5, 11-6
 Automatic Thermally Loaded Elastic-Creep/
 Elastic-Plastic-Creep Stress Analysis (AUTO
 THERM CREEP) 5-57
 Automatically Programmed Tools (APT) B-2
 AXITO3D (model definition option) 5-115

B

bandwidth optimization 4-24
 Barlat's (1991) yield function 7-68
 beam elements 10-21, 12-16
 BEAM SECT (parameter) 4-30, 10-21
 beam-to-beam intersection tying constraints 9-32
 BEARING (parameter) 6-26
 behavior
 incompressible 5-33

 nearly incompressible 5-36
 nonlinear 11-4
 Bergan and Mollestad 11-15
 B-H RELATION (electromagnetic, model
 definition option) 6-43, 7-159
 B-H RELATION (magnetostatic, model definition
 option) 6-34, 6-35, 6-36, 7-159
 Bingham fluid 6-58
 block definition 4-9
 body motion 8-10
 boundary conditions
 input options 9-22
 loading types 9-3
 time-dependent 5-83
 treatment of 5-37
 BUCKLE (history definition option) 5-30, 5-31
 BUCKLE (parameter) 5-9, 5-30
 BUCKLE INCREMENT (model definition option)
 5-9, 5-30, 5-31
 buckling analysis 5-30

C

cable element 10-31
 Cam-Clay model 7-137
 Carreau model 6-59
 CASE COMBIN (model definition option) 5-4, 5-6,
 11-17
 cavity
 pressure loading 9-9
 radiating 6-13
 rigid, acoustic analysis 6-49
 surface elements 9-14
 central difference operator 5-77, 5-81
 CENTROID (parameter) 5-20, 5-29, 6-22, 9-15,
 10-3, 11-17, 11-36, 12-13, 12-18
 Chaboche 7-88, 7-105
 CHANGE PORE (model definition option) 5-97
 CHANGE STATE (history definition option) 5-57,

A B C D E F G H I J K L M N O P Q R S T U V W X Y Z

- 5-58, 11-6
- CHANGE STATE (model definition option) 6-4, 6-22, 9-14
- CHANNEL (model definition option) 6-4, 6-68
- closest point projection procedure 5-47
- combined hardening 7-80
- common material characteristics 5-32
- COMPOSITE (model definition option) 5-112, 7-6, 7-7
- composite continuum elements 7-7
- composite material 7-6
- computers 2-2
- conditioning number 11-42
- CONN FILL (model definition option) 4-23
- CONN GENER (model definition option) 4-23
- CONNECT (model definition option) 4-15
- CONNECTIVITY (model definition option) 4-9, 4-23, 6-70, 6-77
- Conrad Gap 6-19
- CONRAD GAP (model definition option) 6-4
- CONSTANT DILATATION (parameter) 5-18, 10-34
- constant dilatation elements 10-34
- constitutive relations 7-84
- constraint
 - rigid link 9-35
- CONSTRAINT (model definition option) 4-15
- contact
 - adaptive meshing 8-41
 - analysis 12-19
 - beams 8-36
 - bodies 8-5
 - constraint implementation 8-18
 - coupled 8-31
 - deformable 8-54
 - detection 8-13
 - dynamic 8-39
 - friction modeling 8-21
 - mathematical aspects 8-46
 - neighbor relations 8-17
 - quadratic 8-35
 - release 8-30
 - separation 8-29
 - shell 8-15, 8-38
 - thermal 8-34
 - tolerance values 8-43
- CONTACT (model definition option) 5-27, 5-28, 5-37, 5-77, 6-5, 8-4, 8-5, 8-7, 8-36, 9-2, 9-39, 11-11
- CONTACT CHANGE (rezoning option) 8-40
- CONTACT NODE (model definition option) 8-5
- contact penetration criteria, remeshing 4-45
- CONTACT TABLE (history definition option) 8-12, 8-13, 8-30
- CONTACT TABLE (model definition option) 8-20, 8-27, 8-28, 8-33, 8-34, 8-57
- CONTINUE (history definition option) 3-4, 3-8, 3-10
- continuum elements 10-20
- CONTROL (electromagnetic, model definition option) 6-39
- CONTROL (heat transfer, model definition option) 6-70
- CONTROL (history definition option) 5-58, 6-6, 11-4
- CONTROL (magnetostatic, model definition option) 6-34
- CONTROL (model definition option) 5-20, 5-31, 6-5, 6-6, 6-14, 8-44, 11-4, 11-11, 12-8
- control algorithm 4-27
- convergence controls 5-73
- CONVERT (model definition option) 6-71, 6-78
- COORDINATES (model definition option) 4-9, 4-22, 4-23, 12-16
- Coulomb model, friction 8-21
- COUPLE (parameter) 6-70, 6-77
- coupled acoustic-structural analysis 6-51
 - acoustic medium 6-54
 - divergence theorem of Green 6-52

A B C D E F G H I J K L M N O P Q R S T U V W X Y Z

fluid pressure [6-53](#)
 harmonic analyses [6-52](#)
 coupled analysis [6-67](#), [8-31](#)
 coupled contact analysis [8-31](#)
 coupled electrical-thermal-mechanical [6-76](#)
 coupled electrical-thermal-mechanical analysis [6-76](#)
 coupled thermo-electrical analysis (Joule heating) [6-73](#)
 coupled thermo-mechanical analysis [6-69](#)
 crack analysis [5-61](#)
 CRACK DATA (model definition option) [7-133](#)
 cracking
 closure [7-135](#)
 low tension [7-134](#)
 uniaxial [7-133](#)
 creep [5-52](#)
 buckling [5-57](#)
 control tolerances [5-54](#)
 implicit formulation [7-103](#)
 Maxwell model [7-97](#)
 CREEP (model definition option) [5-59](#), [7-97](#), [7-98](#), [7-103](#), [7-104](#), [7-107](#), [7-128](#)
 CREEP (parameter) [5-56](#), [5-59](#), [5-60](#), [7-96](#), [7-103](#), [7-104](#), [7-107](#)
 creep analysis [11-5](#)
 CREEP INCREMENT (history definition option) [11-4](#), [11-6](#)
 Crisfield [11-30](#), [11-32](#)
 criteria
 equivalent values [4-39](#)
 mean strain energy [4-38](#)
 node within a box [4-40](#)
 nodes in contact [4-40](#)
 previously refined mesh [4-40](#)
 remeshing [4-44](#)
 temperature gradient [4-40](#)
 user-defined [4-40](#)
 Zienkiewicz-Zhu [4-38](#)

 Zienkiewicz-Zhu creep strain [4-39](#)
 CRPLAW (user subroutine) [5-53](#), [5-54](#), [5-55](#), [5-59](#), [5-60](#), [7-97](#), [7-98](#), [7-102](#), [7-105](#), [7-107](#), [7-128](#)
 CRPVIS (user subroutine) [5-59](#), [7-96](#), [7-107](#)
 crushing [7-135](#)
 curved-pipe element [10-30](#)
 CURVES (model definition option) [4-36](#)
 Cutter Location (CL) [B-2](#)
 cyclic symmetry [9-38](#)
 CYCLIC SYMMETRY (model definition option) [9-39](#), [9-40](#)
 cyclic symmetry tying constraints [9-38](#)
 CYLINDRICAL (model definition option) [4-7](#)

D

DAMAGE (model definition option) [7-157](#)
 damage models [7-152](#)
 damping [5-83](#)
 DAMPING (model definition option) [5-74](#), [5-83](#), [5-86](#)
 DAMPING COMPONENTS (history definition option) [6-27](#)
 data transfer, axisymmetric analysis to 3-D analysis [5-115](#)
 DEFINE (model definition option) [3-3](#), [3-4](#), [3-5](#)
 deformable contact, solution strategy [8-56](#)
 deformable-deformable contact [8-54](#)
 degree of freedom transformation [9-22](#)
 DENSITY EFFECTS (model definition option) [7-74](#)
 DESIGN DISPLACEMENT CONSTRAINTS (model definition option) [5-112](#)
 DESIGN FREQUENCY CONSTRAINTS (model definition option) [5-112](#)

A B C D E F G H I J K L M N O P Q R S T U V W X Y Z

DESIGN OBJECTIVE (model definition option) 5-112
design optimization 5-106
DESIGN OPTIMIZATION (parameter) 5-111, 7-4
DESIGN SENSITIVITY (parameter) 5-111, 7-4
design sensitivity analysis 5-102
design space 5-107
DESIGN STRAIN CONSTRAINTS (model definition option) 5-112
DESIGN STRESS CONSTRAINTS (model definition option) 5-112
DESIGN VARIABLES (model definition option) 5-112
dilatational creep 7-103
direct
 constraints 8-4
 integration 5-76
 mesh definition input 4-3
 methods 11-41
DISP CHANGE (history definition option) 5-58, 6-70, 6-78
DIST CHARGE (electromagnetic, model definition option) 6-39
DIST CHARGE (model definition option) 6-45
DIST CHARGES (electrostatic, model definition option) 6-31
DIST CURRENT (electromagnetic, model definition option) 6-39
DIST CURRENT (history definition option) 6-78
DIST CURRENT (Joule, history definition option) 6-74
DIST CURRENT (Joule, model definition option) 6-74
DIST CURRENT (magnetostatic, model definition option) 6-34
DIST CURRENT (model definition option) 6-77
DIST FLUXES (history definition option) 6-74, 6-78

DIST FLUXES (model definition option) 5-97, 6-4, 6-10, 6-11, 6-26, 6-74, 6-78
DIST LOADS (history definition option) 5-58, 6-78
DIST LOADS (model definition option) 5-97, 6-45, 6-78, 9-6, 9-9
DIST LOADS (parameter) 5-111
DIST SOURCE (model definition option) 6-50
Domain Decomposition Method (DDM) 13-1
ductile metals 7-152
dynamic
 analysis 12-18
 crack propagation 5-68
 fracture methodology 5-69
 impact 8-39
DYNAMIC (parameter) 5-9, 5-68, 5-70, 5-74, 5-76
DYNAMIC CHANGE (dynamic, history definition option) 6-70
DYNAMIC CHANGE (electromagnetic, history definition option) 6-39, 6-42
DYNAMIC CHANGE (history definition option) 5-74, 5-76, 5-82, 6-50, 6-72, 11-4, 11-6

E

Edge I-J orientation 7-11
effects of temperature and strain rates on yield stress 7-75
eigenvalue analysis 5-70
ELASTIC (parameter) 4-34, 5-3, 5-4
elastic models 7-136
ELASTICITY (parameter) 5-4, 5-23
elastomeric material 7-48
elastomers 7-155
electrical
 analysis 9-18
 input options 9-18
ELECTRO (parameter) 6-31

A B C D E F G H I J K L M N O P Q R S T U V W X Y Z

electromagnetic
 analysis 6-39, 7-159, 9-20, 10-26, 12-19
 element types 6-39
 input options 6-40, 9-20
 electrostatic
 analysis 7-159, 9-18, 10-24, 12-19
 element types 6-30
 input options 9-19
 ELEM SORT (model definition option) 12-10
 element distortion criteria, remeshing 4-44
 elements
 acoustic analysis 6-49, 6-55
 beam 10-21, 12-16
 beams, contact 8-36
 cable 10-31
 classes of stress-strain relations 7-8
 composite continuum 7-7
 connectivity data 4-3
 constant dilatation 10-34
 contact 8-35
 continuum 10-20
 curved-pipe 10-30
 electromagnetic 6-39
 electrostatic 6-30
 fluid 6-65
 Fourier 10-37
 Fourier analysis 5-7
 gap-and-friction 10-30, 12-17
 heat transfer 10-24, 12-17
 heat transfer convection 6-23
 hydrodynamic bearing 6-25
 incompressible 10-32
 magnetostatic 6-33
 membrane 10-19
 piezoelectric 6-44
 pipe-bend 10-30
 plate 10-22
 quarter point 5-67
 rebar 10-31

reduced integration 10-35
 semi-infinite 10-38
 shear panel 10-30
 shell 8-38, 10-23
 special 10-30
 structural classification 10-10
 truss 10-18
 types library 10-3
 ELEMENTS (parameter) 10-2
 ELEVAR (user subroutine) 12-11, 12-15
 ELEVEC (user subroutine) 12-11
 EL-MA (parameter) 6-39
 ELSTO (parameter) 2-3, 2-5, 2-6, 2-7, 12-3
 END (parameter) 3-6, 3-8
 END OPTION (model definition option) 3-2, 3-7, 3-8, 3-10, 5-111, 6-50
 END REZONE (rezoning option) 3-10
 Equivalent Values Criterion 4-39
 error analysis 4-33
 ERROR ESTIMATE (model definition option) 4-33, 5-5, 5-37
 Eularian formulation 5-25
 examples of lists input 3-5
 EXCLUDE (history definition option) 8-15
 EXTENDED (parameter) 3-3

F

FAIL DATA (model definition option) 7-22, 7-133
 file units 2-6
 FILM (user subroutine) 6-4, 6-10, 6-11, 6-12, 6-74, 6-78, 9-17
 FILMS (model definition option) 6-4, 6-10, 6-74, 6-78, 9-17
 FINITE (parameter) 5-16, 5-18, 5-23, 5-36, 7-30, 7-74, 7-78
 finite element technology
 basic concepts A-1

A B C D E F G H I J K L M N O P Q R S T U V W X Y Z

FIXED DISP (model definition option) 4-9, 4-15,
5-97, 6-45, 6-70, 6-72, 6-78

FIXED POTENTIAL (electromagnetic, model
definition option) 6-39

FIXED POTENTIAL (electrostatic, model
definition option) 6-31

FIXED POTENTIAL (magnetostatic, model
definition option) 6-34

FIXED PRESSURE (model definition option)
5-97, 6-26, 6-50

FIXED TEMPERATURE (model definition option)
6-4, 6-10, 6-66, 6-70, 6-74, 6-78

FIXED VELOCITY (model definition option) 6-66

FLD (Forming Limit Diagram) 12-22

flow

diagram 11-47

rule 7-83

FLP (Forming Limit Parameter) 12-22

fluid

analysis 6-56, 7-160, 10-28

elements 6-65

FLUID (parameter) 10-28

FLUID DRAG (model definition option) 9-6

fluid drag and wave

input options 9-7

loads 9-6

fluid mechanics 6-56

Bingham fluid 6-58

Carreau model 6-59

element types 6-65

piecewise non-Newtonian flow 6-58

power law fluid 6-59

FLUID SOLID (model definition option) 6-72

fluid/solid interaction analysis 6-71, 7-159,
10-24

FLUX (user subroutine) 6-4, 6-10, 6-11, 6-26,
6-31, 6-34, 6-45, 6-50, 6-74, 6-78, 9-17

FOLLOW FOR (parameter) 5-20, 5-29, 7-74, 9-9,
9-26, 10-41

follow force stiffness contribution 10-41

FORCDT (user subroutine) 5-83, 6-10, 6-11, 6-27,
6-31, 6-34, 6-45, 6-66

force components 9-4

FORCEM (user subroutine) 4-36, 5-83, 6-39, 6-45

Forming Limit Diagram (FLD) 12-22

Forming Limit Parameter (FLP) 12-22

FOUNDATION (model definition option) 5-4, 5-29

FOURIER (model definition option) 5-4, 5-6, 5-9

FOURIER (parameter) 5-6, 5-9

Fourier analysis 5-6

coefficients 5-7

elements 10-37

elements used 5-7

fractal dimension 11-42

fracture mechanics 5-61

friction modeling 8-21

FXORD (model definition option) 4-17, 4-22,
10-23

G

gap-and-friction element 10-30, 12-17

gasket 7-23

GASKET (model definition option) 7-27

generation phase 5-69

geometric domains 10-16

GEOMETRY (model definition option) 4-30,
5-112, 6-26, 10-22, 10-34, 10-40, 12-16

global

meshing 8-40

remeshing 4-41

ZX plane orientation 7-12

glue model, friction 8-28

gradient

objective or response function 5-102

pressure 5-98

Gurson model, modified 7-152

A B C D E F G H I J K L M N O P Q R S T U V W X Y Z

H

harmonic
 analysis [6-42](#)
 response [5-84](#)
 HARMONIC (electromagnetic, history definition option) [6-43](#)
 HARMONIC (history definition option) [5-85](#)
 HARMONIC (parameter) [5-85](#), [5-86](#), [5-88](#)
 HEAT (parameter) [6-22](#), [6-23](#), [10-24](#)
 heat fluxes [8-31](#), [9-16](#)
 (special) conditions [9-17](#)
 input options [9-16](#)
 heat transfer
 analysis [6-3](#), [7-158](#), [9-16](#), [12-19](#)
 convection [6-22](#)
 convection elements [6-23](#)
 elements [10-24](#), [12-17](#)
 temperature effects [6-8](#)
 hereditary integral model [7-108](#)
 Hill
 failure criterion [7-17](#)
 yield function [7-66](#)
 history definition options [3-8](#)
 Hoffman failure criterion [7-18](#)
 HOOKLW (user subroutine) [5-4](#), [7-5](#), [7-8](#)
 HOOKVI (user subroutine) [7-113](#)
 Houbolt operator [5-77](#)
 hydrodynamic bearing [12-17](#)
 analysis [7-158](#), [9-17](#), [10-25](#), [12-19](#)
 elements [6-25](#)
 mass fluxes [9-17](#)
 pump pressures [9-17](#)
 restrictors [9-17](#)
 HYPELA (user subroutine) [5-20](#), [7-30](#), [7-136](#)
 HYPELA2 (user subroutine) [7-30](#), [7-136](#)
 HYPOELASTIC (model definition option) [7-30](#)

I

IBOOC (parameter) [2-4](#), [2-5](#)
 IMPD (user subroutine) [12-11](#)
 incompressible
 elements [10-32](#)
 Herrmann elastic formulation [5-35](#)
 material behavior [5-33](#)
 material modeled [5-35](#)
 incremental mesh generators [4-23](#)
 initial conditions [5-83](#)
 INITIAL DISP (model definition option) [5-83](#)
 INITIAL PC (model definition option) [5-97](#)
 initial plastic strain [9-15](#)
 INITIAL PORE (model definition option) [5-97](#)
 INITIAL POROSITY (model definition option) [5-97](#)
 initial stress and initial plastic strain input options [9-16](#)
 INITIAL TEMP (model definition option) [6-3](#), [6-10](#), [6-74](#), [6-78](#)
 INITIAL VEL (model definition option) [5-83](#)
 INITIAL VOID RATIO (model definition option) [5-97](#)
 input conventions [3-2](#)
 input options
 acoustic analysis sources [9-19](#)
 boundary conditions [9-22](#)
 electrical [9-18](#)
 electromagnetic [6-40](#), [9-20](#)
 electrostatic [9-19](#)
 fluid drag and wave loads [9-7](#)
 heat fluxes [9-16](#)
 initial stress and initial plastic strain [9-16](#)
 magnetostatic currents [9-20](#)
 mass fluxes and restrictors [9-18](#)
 mechanical loads [9-8](#)
 thermal loads [9-15](#)
 INSERT (model definition option) [9-37](#)
 INTCRD (user subroutine) [12-11](#)

A B C D E F G H I J K L M N O P Q R S T U V W X Y Z

interface

Adams results [12-29](#)

hyperMesh results [12-27](#)

SDRC I-DEAS results [12-28](#)

interpolation (shape) functions, type of [10-17](#)

inverse power sweep [5-71](#)

ISOTROPIC (acoustic, model definition option) [6-50](#), [7-160](#)

ISOTROPIC (electromagnetic, model definition option) [6-39](#), [7-159](#)

ISOTROPIC (electrostatic, model definition option) [6-31](#), [7-159](#)

ISOTROPIC (fluid, model definition option) [6-72](#), [7-159](#), [7-160](#)

ISOTROPIC (heat transfer, model definition option) [6-74](#), [7-158](#), [9-15](#)

ISOTROPIC (hydrodynamic, model definition option) [7-158](#)

ISOTROPIC (magnetostatic, model definition option) [6-33](#), [7-159](#)

ISOTROPIC (model definition option) [3-3](#), [5-112](#), [6-3](#), [6-8](#), [6-26](#), [6-44](#), [6-66](#), [7-4](#), [7-7](#), [7-27](#), [7-65](#), [7-66](#), [7-71](#), [7-72](#), [7-75](#), [7-76](#), [7-77](#), [7-78](#), [7-80](#), [7-82](#), [7-83](#), [7-98](#), [7-102](#), [7-103](#), [7-106](#), [7-110](#), [7-112](#), [7-126](#), [7-127](#), [7-133](#), [9-8](#), [9-9](#)

isotropic hardening [7-78](#)

J

J-INTEGRAL (model definition option) [5-4](#)

Joule

heating [6-73](#)

structural [6-76](#)

JOULE (model definition option) [6-74](#), [6-78](#)

JOULE (parameter) [6-74](#), [6-77](#)

K

Kachanov factor [7-157](#)

Kelvin-Voigt model [7-107](#)

kinematic

constraints [9-21](#)

hardening [7-80](#)

L

Lagrange multipliers [8-3](#)

Lagrangian formulation [5-15](#)

Lanczos method [5-72](#)

large deformations [5-33](#)

LARGE DISP (parameter) [5-16](#), [5-18](#), [5-20](#), [5-23](#), [5-29](#), [5-30](#), [5-36](#), [7-30](#), [7-74](#), [7-77](#), [7-78](#)

large strain

analysis [10-32](#), [10-33](#)

elasticity [5-33](#)

plasticity [5-36](#)

linear analysis [5-3](#)

linear elastic material [7-3](#)

linear fracture mechanics [5-61](#)

linear springs and elastic foundations input options [9-37](#)

linked/unlinked [5-112](#)

list of items, input [3-3](#)

load

AUTO STEP scheme [11-9](#)

automatic incrementation [11-7](#)

cavity pressure [9-9](#)

fixed [11-7](#)

fluid drag and wave [9-6](#)

history definition options [11-6](#)

increment size [11-7](#)

mass fluxes [9-17](#)

mechanical [9-8](#)

mechanical and thermal increments [11-4](#)

other AUTO STEP options [11-13](#)

piezoelectric [9-19](#)

A B C D E F G H I J K L M N O P Q R S T U V W X Y Z

pump pressures [9-17](#)
 recycling criterion [11-9](#)
 residual [12-18](#)
 residual load correction [11-16](#)
 restrictors [9-17](#)
 surface/volumetric [9-4](#)
 thermal [9-14](#)
 types [9-3](#)
 user-defined physical criteria [11-12](#)
 vectors [A-7](#)
 LOAD COR (parameter) [9-9](#)
 load incrementation
 fixed scheme [11-4](#)
 load incrementqtation
 adaptive scheme [11-4](#)
 LORENZI (model definition option) [5-68](#)

M

machining (or metal cutting) [B-1](#)
 MAGNETO (parameter) [6-33](#)
 magnetostatic
 analysis [7-159](#), [9-19](#), [10-25](#), [12-19](#)
 current input options [9-20](#)
 elements types [6-33](#)
 MANY TYPES (model definition option) [4-15](#)
 mass fluxes and restrictors [9-17](#)
 mass fluxes and restrictors input options [9-18](#)
 MASSES (model definition option) [5-86](#)
 material
 dependent failure criteria [7-14](#)
 instabilities [5-37](#)
 low tension [7-133](#)
 powder [7-73](#)
 preferred direction [7-10](#)
 shape memory [7-43](#)
 material model
 composite [7-6](#)
 elastomers [7-48](#)

 gasket [7-23](#)
 linear elastic [7-3](#)
 Mohr-Coulomb (hydrostatic yield
 dependence) [7-70](#)
 nonlinear hypoelastic [7-30](#)
 time-independent inelastic [7-61](#)
 maximum
 strain failure criterion [7-16](#)
 stress criterion [7-14](#)
 Mean Strain Energy Criterion [4-38](#)
 mean-normal method [5-47](#)
 mechanical
 loads [9-8](#)
 loads input options [9-8](#)
 membrane elements [10-19](#)
 MERGE (model definition option) [4-10](#)
 mesh
 definition [4-2](#)
 refinement tying constraints [9-29](#)
 severe distortion [5-37](#)
 mesh generation, remeshing techniques [4-46](#)
 advancing front meshing [4-46](#)
 Delaunay Triangulation [4-48](#)
 overlay meshing [4-47](#)
 MESH2D [4-9](#)
 MESH2D (model definition option) [4-15](#)
 modal
 stresses and reactions [5-73](#)
 superposition [5-74](#)
 MODAL INCREMENT (model definition option)
 [5-9](#)
 MODAL SHAPE (history definition option) [5-71](#),
 [5-73](#), [6-50](#), [6-72](#)
 model definition options [3-7](#)
 Mohr-Coulomb
 linear material [7-70](#)
 parabolic material [7-72](#)
 Monte Carlo method [6-16](#)

A B C D E F G H I J K L M N O P Q R S T U V W X Y Z

MOONEY (model definition option) 7-52, 7-57, 7-119

Mooney-Rivlin 7-52, 7-58

MOTION (2-D, user subroutine) 8-10

MOTION CHANGE (history definition option) 8-12, 8-30

MPI (Message Passing Interface) 13-2

MSC.Marc

- host systems 2-2

- overview 1-3

MSC.Nastran RBE2 and RBE3 9-41

Mullin's effect 7-155

multiplicative decomposition 5-18, 5-42, 7-60

multistage return mapping 5-49

N

Narayanaswamy model 5-58, 7-120

NASSOC (user subroutine) 5-59, 5-60, 7-105

Navier-Stokes 6-56, 6-60, 6-66, 6-68

NC machining data B-2

- cutter path B-3

- cutter shape B-2

- element deactivation B-3

- intersection B-3

nearly incompressible behavior 5-36

neighbor relations 8-17

Neo-Hookean 7-52

Newmark-beta operator 5-76, 5-78

Newton Cooling 7-131

Newton-Raphson method

- full 11-19

- modified 11-21

NiTi alloys 7-31

nodal degrees of freedom tying constraints 9-34

NODE CIRCLE (model definition option) 4-23

NODE FILL (model definition option) 4-23

NODE GENER (model definition option) 4-23

NODE MERGE (model definition option) 4-23

NODE SORT (model definition option) 12-10

Node Within A Box Criterion 4-40

Nodes In Contact Criterion 4-40

nodes, merging of 4-10

nonlinear

- analysis 5-10

- fracture mechanics 5-63

nonlinearities, three types 5-15

non-Newtonian 6-58

nonstructural materials 7-158

NURBS 8-6, 8-7, 8-18, 8-19, 8-49, 8-50

- trimmed 8-6, 8-42

O

Oak Ridge National Laboratory (ORNL) 5-53, 7-75, 7-101

objective function 5-102, 5-104, 5-110, 5-111, 5-113, 5-114

objective stress rates 5-39

OGDEN (model definition option) 7-53, 7-57, 7-119, 7-157

operating systems 2-2

operator

- central difference 5-77

- Houbolt 5-77

- Newmark-beta 5-76

- single step Houbolt 5-77

OPHTHOTROPIC (model definition option) 7-82

optimization algorithm 5-110

OPTIMIZE (model definition option) 4-24

Option

- ACTIVATE (history definition) 4-7

- ADAPT GLOBAL (model definition) 4-41

- ADAPTIVE (model definition) 4-38, 5-4, 5-5, 10-43

- ANISOTROPIC (model definition) 5-4, 6-44, 7-5, 7-7, 7-66, 7-83

- APPROACH (history definition) 8-12

A B C D E F G H I J K L M N O P Q R S T U V W X Y Z

ATTACH EDGE (model definition) 4-36
 AUTO CREEP (history definition) 5-53, 5-55,
 5-58, 5-59, 5-60, 7-107, 11-4, 11-5, 11-6
 AUTO INCREMENT (history definition) 4-27,
 5-30, 7-135, 9-9, 11-4, 11-5, 11-6, 11-7,
 11-9
 AUTO INCREMENT (model definition) 11-31
 AUTO LOAD (history definition) 4-27, 7-119,
 9-9, 11-4, 11-6
 AUTO STEP (history definition) 5-76, 6-6,
 6-10, 7-112, 7-119, 7-135, 8-49, 8-61, 11-4,
 11-5, 11-6, 11-9, 11-11, 11-12, 11-13,
 11-14, 11-15
 AUTO THERM (history definition) 9-14, 11-4,
 11-6
 AUTO THERM CREEP (history definition)
 5-57, 5-58, 11-5, 11-6
 AXITO3D (model definition) 5-115
 B-H RELATION (electromagnetic, model
 definition) 6-43, 7-159
 B-H RELATION (magnetostatic, model
 definition) 6-34, 6-35, 6-36, 7-159
 BUCKLE (history definition) 5-30, 5-31
 BUCKLE INCREMENT (model definition) 5-9,
 5-30, 5-31
 CASE COMBIN (model definition) 5-4, 5-6,
 11-17
 CHANGE PORE (model definition) 5-97
 CHANGE STATE (history definition) 5-57,
 5-58, 11-6
 CHANGE STATE (model definition) 6-4, 6-22,
 9-14
 CHANNEL (model definition) 6-4, 6-68
 COMPOSITE (model definition) 5-112, 7-6,
 7-7
 CONN FILL (model definition) 4-23
 CONN GENER (model definition) 4-23
 CONNECT (model definition) 4-15

CONNECTIVITY (model definition) 4-9, 4-23,
 6-70, 6-77
 CONRAD GAP (model definition) 6-4
 CONSTRAINT (model definition) 4-15
 CONTACT (model definition) 5-27, 5-28,
 5-37, 5-77, 6-5, 8-4, 8-5, 8-7, 8-36, 9-2,
 9-39, 11-11
 CONTACT CHANGE (rezoning) 8-40
 CONTACT NODE (model definition) 8-5
 CONTACT TABLE (history definition) 8-12,
 8-13, 8-30
 CONTACT TABLE (model definition) 8-20,
 8-27, 8-28, 8-33, 8-34, 8-57
 CONTINUE (history definition) 3-4, 3-8, 3-10
 CONTROL (electromagnetic, model
 definition) 6-39
 CONTROL (heat transfer, model definition)
 6-70
 CONTROL (history definition) 5-58, 6-6, 11-4
 CONTROL (magnetostatic, model definition)
 6-34
 CONTROL (model definition) 5-20, 5-31, 6-5,
 6-6, 6-14, 8-44, 11-4, 11-11, 12-8
 CONVERT (model definition) 6-71, 6-78
 COORDINATES (model definition) 4-9, 4-22,
 4-23, 12-16
 CRACK DATA (model definition) 7-133
 CREEP (model definition) 5-59, 7-97, 7-98,
 7-103, 7-104, 7-107, 7-128
 CREEP INCREMENT (history definition) 11-4,
 11-6
 CURVES (model definition) 4-36
 CYCLIC SYMMETRY (model definition) 9-39,
 9-40
 CYLINDRICAL (model definition) 4-7
 DAMAGE (model definition) 7-157
 DAMPING (model definition) 5-74, 5-83, 5-86

A B C D E F G H I J K L M N O P Q R S T U V W X Y Z

DAMPING COMPONENTS (history definition) 6-27

DEFINE (model definition) 3-3, 3-4, 3-5

DENSITY EFFECTS (model definition) 7-74

DESIGN DISPLACEMENT CONSTRAINTS (model definition) 5-112

DESIGN FREQUENCY CONSTRAINTS (model definition) 5-112

DESIGN OBJECTIVE (model definition) 5-112

DESIGN STRAIN CONSTRAINTS (model definition) 5-112

DESIGN STRESS CONSTRAINTS (model definition) 5-112

DESIGN VARIABLES (model definition) 5-112

DISP CHANGE (history definition) 5-58, 6-70, 6-78

DIST CHARGE (electromagnetic, model definition) 6-39

DIST CHARGE (model definition) 6-45

DIST CHARGES (electrostatic, model definition) 6-31

DIST CURRENT (electromagnetic, model definition) 6-39

DIST CURRENT (history definition) 6-78

DIST CURRENT (Joule, history definition) 6-74

DIST CURRENT (Joule, model definition) 6-74

DIST CURRENT (magnetostatic, model definition) 6-34

DIST CURRENT (model definition) 6-77

DIST FLUXES (history definition) 6-74, 6-78

DIST FLUXES (model definition) 5-97, 6-4, 6-10, 6-11, 6-26, 6-74, 6-78

DIST LOADS (history definition) 5-58, 6-78

DIST LOADS (model definition) 5-97, 6-45, 6-78, 9-6, 9-9

DIST SOURCE (model definition) 6-50

DYNAMIC CHANGE (history definition) 5-74, 5-76, 5-82, 6-50, 6-72, 11-4, 11-6

DYNAMIC CHANGE (history definition, dynamic) 6-70

DYNAMIC CHANGE (history definition, electromagnetic) 6-39, 6-42

ELEM SORT (model definition) 12-10

END OPTION (model definition) 3-2, 3-7, 3-8, 3-10, 5-111, 6-50

END REZONE (rezoning) 3-10

ERROR ESTIMATE (model definition) 4-33, 5-5, 5-37

EXCLUDE (history definition) 8-15

FAIL DATA (model definition) 7-22, 7-133

FILMS (model definition) 6-4, 6-10, 6-74, 6-78, 9-17

FIXED DISP (model definition) 4-9, 4-15, 5-97, 6-45, 6-70, 6-72, 6-78

FIXED POTENTIAL (electromagnetic, model definition) 6-39

FIXED POTENTIAL (electrostatic, model definition) 6-31

FIXED POTENTIAL (magnetostatic, model definition) 6-34

FIXED PRESSURE (model definition) 5-97, 6-26, 6-50

FIXED TEMPERATURE (model definition) 6-4, 6-10, 6-66, 6-70, 6-74, 6-78

FIXED VELOCITY (model definition) 6-66

FLUID DRAG (model definition) 9-6

FLUID SOLID (model definition) 6-72

FOUNDATION (model definition) 5-4, 5-29

FOURIER (model definition) 5-4, 5-6, 5-9

FXORD (model definition) 4-17, 4-22, 10-23

GASKET (model definition) 7-27

GEOMETRY (model definition) 4-30, 5-112, 6-26, 10-22, 10-34, 10-40, 12-16

HARMONIC (electromagnetic, history definition) 6-43

A B C D E F G H I J K L M N O P Q R S T U V W X Y Z

HARMONIC (history definition) 5-85
 HYPOELASTIC (model definition) 7-30
 INITIAL DISP (model definition) 5-83
 INITIAL PC (model definition) 5-97
 INITIAL PORE (model definition) 5-97
 INITIAL POROSITY (model definition) 5-97
 INITIAL TEMP (model definition) 6-3, 6-10, 6-74, 6-78
 INITIAL VEL (model definition) 5-83
 INITIAL VOID RATIO (model definition) 5-97
 INSERT (model definition) 9-37
 ISOTROPIC (acoustic, model definition) 6-50, 7-160
 ISOTROPIC (electromagnetic, model definition) 6-39, 7-159
 ISOTROPIC (electrostatic, model definition) 6-31, 7-159
 ISOTROPIC (fluid, model definition) 6-72, 7-159, 7-160
 ISOTROPIC (heat transfer, model definition) 6-74, 7-158, 9-15
 ISOTROPIC (hydrodynamic, model definition) 7-158
 ISOTROPIC (magnetostatic, model definition) 6-33, 7-159
 ISOTROPIC (model definition) 3-3, 5-112, 6-3, 6-8, 6-26, 6-44, 6-66, 7-4, 7-7, 7-27, 7-65, 7-66, 7-71, 7-72, 7-75, 7-76, 7-77, 7-78, 7-80, 7-82, 7-83, 7-98, 7-102, 7-103, 7-106, 7-110, 7-112, 7-126, 7-127, 7-133, 9-8, 9-9
 J-INTEGRAL (model definition) 5-4
 JOULE (model definition) 6-74, 6-78
 LORENZI (model definition) 5-68
 MANY TYPES (model definition) 4-15
 MASSES (model definition) 5-86
 MERGE (model definition) 4-10
 MESH2D (model definition) 4-15
 MODAL INCREMENT (model definition) 5-9

MODAL SHAPE (history definition) 5-71, 5-73, 6-50, 6-72
 MOONEY (model definition) 7-52, 7-57, 7-119
 MOTION CHANGE (history definition) 8-12, 8-30
 NODE CIRCLE (model definition) 4-23
 NODE FILL (model definition) 4-23
 NODE GENER (model definition) 4-23
 NODE MERGE (model definition) 4-23
 NODE SORT (model definition) 12-10
 OGDEN (model definition) 7-53, 7-57, 7-119, 7-157
 OPTIMIZE (model definition) 4-24
 ORIENTATION (model definition) 7-5, 7-7, 7-10, 7-66, 7-113, 10-3
 ORTHO TEMP (model definition) 6-3, 6-74, 7-7, 7-75, 7-76, 7-126, 7-127
 ORTHOTROPIC (electrical, model definition) 7-159
 ORTHOTROPIC (electromagnetic, model definition) 6-39, 7-159
 ORTHOTROPIC (electrostatic, model definition) 6-31
 ORTHOTROPIC (magnetostatic, model definition) 6-33, 7-159
 ORTHOTROPIC (model definition) 5-4, 5-112, 6-3, 7-5, 7-7, 7-65, 7-66, 7-75, 7-77, 7-78, 7-80, 7-82, 7-83, 7-103, 7-106, 7-110, 7-112, 7-126, 7-127, 7-129, 9-8, 9-9
 ORTHOTROPIC (model definition) 6-44
 ORTHOTROPIC (thermal, model definition) 6-74, 7-158, 9-15
 PARAMETERS (history definition) 5-79
 PARAMETERS (model definition) 5-90, 5-92, 6-66, 8-19, 8-20, 8-33
 PERMANENT (magnetostatic, model definition) 6-34, 6-35, 6-36
 PIEZOELECTRIC (model definition) 6-44

A B C D E F G H I J K L M N O P Q R S T U V W X Y Z

POINT CHARGE (model definition) [6-31](#), [6-45](#)
POINT CURRENT (electromagnetic, history definition) [6-39](#)
POINT CURRENT (history definition) [6-78](#)
POINT CURRENT (Joule, history definition) [6-74](#)
POINT CURRENT (Joule, model definition) [6-74](#)
POINT CURRENT (magnetostatic, model definition) [6-34](#)
POINT CURRENT (model definition) [6-77](#)
POINT FLUX (history definition) [6-74](#), [6-78](#)
POINT FLUX (model definition) [5-97](#), [6-4](#), [6-10](#), [6-26](#), [6-66](#), [6-74](#), [6-78](#)
POINT LOAD (history definition) [5-58](#), [6-78](#)
POINT LOAD (model definition) [3-3](#), [5-97](#), [6-45](#), [6-78](#), [8-10](#)
POINT SOURCE (model definition) [6-50](#)
POST (model definition) [5-58](#), [6-4](#), [6-22](#), [11-7](#), [12-21](#)
POWDER (model definition) [7-74](#)
PRINT CHOICE (model definition) [5-88](#), [7-20](#), [12-9](#), [12-10](#), [12-12](#), [12-13](#), [12-15](#), [12-18](#)
PRINT ELEMENT (model definition) [7-20](#), [12-9](#), [12-12](#), [12-13](#), [12-15](#)
PRINT NODE (model definition) [12-9](#), [12-10](#), [12-12](#), [12-18](#)
PRINT VMASS (model definition) [12-10](#)
PROPORTIONAL INCREMENT (history definition) [4-27](#), [5-58](#), [5-60](#), [9-9](#)
RADIATING CAVITY (model definition) [6-13](#)
REAUTO (model definition) [5-53](#), [11-17](#)
REBAR (model definition) [7-133](#)
RECOVER (history definition) [5-73](#)
RELATIVE DENSITY (model definition) [7-74](#)
RELEASE (history definition) [8-30](#)
RESTART (model definition) [2-6](#), [5-4](#), [5-53](#), [11-17](#), [11-18](#), [12-10](#), [12-12](#)

RESTART INCREMENT (history definition) [11-18](#)
RESTART LAST (model definition) [11-18](#)
RESTRICTOR (model definition) [6-26](#)
REZONE (rezoning) [3-10](#)
ROTATION A (model definition) [9-8](#)
SERVO LINK (model definition) [9-2](#), [9-28](#)
SHELL TRANSFORMATION (model definition) [9-23](#), [9-26](#)
SHIFT FUNCTION (model definition) [5-58](#), [7-116](#)
SOIL (model definition) [5-97](#), [7-136](#)
SOLVER (model definition) [10-41](#), [11-41](#), [11-43](#)
SPECIFIED NODES (model definition) [4-15](#)
SPLINE (model definition) [8-6](#), [8-54](#)
SPRINGS (model definition) [5-4](#), [5-29](#), [5-86](#)
START NUMBER (model definition) [4-15](#)
STEADY STATE (electrostatic, history definition) [6-31](#)
STEADY STATE (history definition) [6-6](#), [6-63](#), [6-74](#)
STEADY STATE (magnetostatic, history definition) [6-34](#)
STIFFNS COMPONENTS (history definition) [6-27](#)
STIFFSCALE (model definition) [4-29](#)
STRAIN RATE (model definition) [5-92](#), [6-66](#), [7-7](#), [7-76](#)
STRAIN RATE (model definition, fluid) [7-160](#)
SUBSTRUCTURE (model definition) [4-27](#)
SUMMARY (model definition) [12-10](#)
SUPERINPUT (model definition) [4-27](#)
SURFACES (model definition) [4-36](#)
SYNCHRONIZED (history definition) [8-12](#)
TABLE (model definition) [7-25](#), [11-8](#), [11-9](#)
TEMP CHANGE (history definition) [6-4](#), [6-10](#), [6-66](#), [6-70](#), [6-74](#), [6-78](#)

A B C D E F G H I J K L M N O P Q R S T U V W X Y Z

TEMPERATURE EFFECTS (coupled thermal-stress, model definition) [7-158](#)

TEMPERATURE EFFECTS (heat transfer, model definition) [6-74](#), [7-158](#)

TEMPERATURE EFFECTS (hydrodynamic, model definition) [7-158](#)

TEMPERATURE EFFECTS (model definition) [5-92](#), [6-3](#), [6-8](#), [6-26](#), [7-7](#), [7-52](#), [7-74](#), [7-75](#), [7-76](#), [7-110](#), [7-116](#), [7-126](#), [7-127](#), [7-129](#), [7-160](#)

THERMAL CONTACT (model definition) [6-5](#)

THICKNESS (model definition) [6-26](#)

THICKENS CHANGE (history definition) [6-26](#)

TIME STEP (history definition) [7-76](#), [7-112](#), [7-119](#)

TIME-TEMP (model definition) [7-130](#), [7-131](#)

TRANSFORMATION (model definition) [9-22](#), [9-32](#), [12-18](#)

TRANSIENT (history definition) [6-4](#), [6-6](#), [6-7](#), [6-10](#), [6-74](#), [11-5](#), [11-6](#)

TYING (model definition) [6-4](#), [6-72](#), [9-2](#), [9-28](#)

TYING CHANGE (history definition) [9-28](#)

UFCONN (model definition) [4-8](#)

UFXORD (model definition) [4-8](#)

UTRANFORM (model definition) [9-22](#)

VELOCITY (model definition) [6-4](#), [6-23](#), [6-26](#)

VELOCITY CHANGE (history definition) [6-4](#), [6-23](#), [6-66](#)

VIEW FACTOR (model definition) [6-17](#), [9-17](#)

VISCEL EXP (model definition) [5-59](#), [7-124](#)

VISCELMOON (model definition) [5-59](#), [7-52](#), [7-119](#)

VISCELOGDEN (model definition) [5-59](#), [7-53](#), [7-119](#), [7-157](#)

VISCELORTH (model definition) [7-112](#)

VISCPLPROP (model definition) [7-112](#)

VOLTAGE (model definition) [6-74](#), [6-77](#)

VOLTAGE CHANGE (history definition) [6-74](#), [6-78](#)

WORK HARD (model definition) [5-92](#), [7-7](#), [7-76](#), [7-78](#), [7-80](#), [7-82](#)

ORIENT (user subroutine) [5-4](#), [7-5](#), [7-10](#), [7-13](#), [7-66](#), [7-113](#)

ORIENTATION (model definition option) [7-5](#), [7-7](#), [7-10](#), [7-66](#), [7-113](#), [10-3](#)

ORTHO TEMP (model definition option) [6-3](#), [6-74](#), [7-7](#), [7-75](#), [7-76](#), [7-126](#), [7-127](#)

ORTHOTROPIC (electrical, model definition option) [7-159](#)

ORTHOTROPIC (electromagnetic, model definition option) [6-39](#), [7-159](#)

ORTHOTROPIC (electrostatic, model definition option) [6-31](#)

ORTHOTROPIC (magnetostatic, model definition option) [6-33](#), [7-159](#)

ORTHOTROPIC (model definition option) [5-4](#), [5-112](#), [6-3](#), [6-44](#), [7-5](#), [7-7](#), [7-65](#), [7-66](#), [7-75](#), [7-77](#), [7-78](#), [7-80](#), [7-83](#), [7-103](#), [7-106](#), [7-110](#), [7-112](#), [7-126](#), [7-127](#), [7-129](#), [9-8](#), [9-9](#)

ORTHOTROPIC (thermal, model definition option) [6-74](#), [7-158](#), [9-15](#)

out-of-core storage options [2-5](#)

output results

workspace information [12-3](#)

P

parallel processing [13-1](#)

contact analysis [13-4](#)

domain decomposition [13-5](#)

solvers [13-3](#)

types of machines [13-2](#)

unsupported features [13-3](#)

Parameter

ACOUSTIC [6-50](#)

ADAPTIVE [8-40](#), [8-41](#), [10-43](#)

ALL POINTS [6-22](#), [10-3](#)

ASSUMED STRAIN [10-40](#)

A B C D E F G H I J K L M N O P Q R S T U V W X Y Z

BEAM SECT 4-30, 10-21
BEARING 6-26
BUCKLE 5-9, 5-30
CENTROID 5-20, 5-29, 6-22, 9-15, 10-3,
11-17, 11-36, 12-13, 12-18
CONSTANT DILATATION 5-18, 10-34
COUPLE 6-70, 6-77
CREEP 5-56, 5-59, 5-60, 7-96, 7-103, 7-104,
7-107
DESIGN OPTIMIZATION 5-111, 7-4
DESIGN SENSITIVITY 5-111, 7-4
DIST LOADS 5-111
DYNAMIC 5-9, 5-68, 5-70, 5-74, 5-76
ELASTIC 4-34, 5-3, 5-4
ELASTICITY 5-4, 5-23
ELECTRO 6-31
ELEMENTS 10-2
EL-MA 6-39
ELSTO 2-3, 2-5, 2-6, 2-7, 12-3
END 3-6, 3-8
EXTENDED 3-3
FINITE 5-16, 5-18, 5-23, 5-36, 7-30, 7-74,
7-78
FLUID 10-28
FOLLOW FOR 5-20, 5-29, 7-74, 9-9, 9-26,
10-41
FOURIER 5-6, 5-9
HARMONIC 5-85, 5-86, 5-88
HEAT 6-22, 6-23, 10-24
IBOOC 2-4, 2-5
JOULE 6-74, 6-77
LARGE DISP 5-16, 5-18, 5-20, 5-23, 5-29,
5-30, 5-36, 7-30, 7-74, 7-77, 7-78
LOAD COR 9-9
MAGNETO 6-33
PIEZO 6-44
PLASTICITY 5-18, 5-19, 5-23, 5-49, 7-78
PRINT 5-31, 12-20
RADIATION 6-13, 6-17, 7-158, 9-17

RESTRICTOR 6-26
REZONING 4-25
R-P FLOW 5-90, 11-26
SCALE 11-4
SHELL SECT 10-22
SIZING 8-40, 10-2, 12-6
STATE VARS 6-26
STOP 2-5, 8-45
SUBSTRUC 4-27
SUPER 4-27
TSHEAR 7-20, 12-16
T-T-T 7-130
UPDATE 5-16, 5-18, 5-23, 5-36, 5-77, 6-71,
7-30, 7-74, 7-78, 7-139, 9-26
VISCO ELAS 5-59
parameters 3-6
PARAMETERS (history definition option) 5-79
PARAMETERS (model definition option) 5-90,
5-92, 6-66, 8-19, 8-20, 8-33
penalty 5-90, 6-62
 methods 8-3
PERMANENT (magnetostatic, model definition
 option) 6-34, 6-35, 6-36
perturbation analysis 5-31
piecewise linear representation 7-126
PIEZO (parameter) 6-44
piezoelectric
 analysis 6-44, 7-159, 9-19, 10-29, 12-19
 coupling 6-47
 element types 6-44
 loads 9-19
PIEZOELECTRIC (model definition option) 6-44
pipe bend element tying constraints 9-33
pipe-bend element 10-30
PLASTICITY (parameter) 5-18, 5-19, 5-23, 5-49,
7-78
plate elements 10-22
PLOTV (user subroutine) 12-21

A B C D E F G H I J K L M N O P Q R S T U V W X Y Z

POINT CHARGE (model definition option) 6-31, 6-45

POINT CURRENT (electromagnetic, history definition option) 6-39

POINT CURRENT (history definition option) 6-78

POINT CURRENT (Joule, history definition option) 6-74

POINT CURRENT (Joule, model definition option) 6-74

POINT CURRENT (magnetostatic, model definition option) 6-34

POINT CURRENT (model definition option) 6-77

POINT FLUX (history definition option) 6-74, 6-78

POINT FLUX (model definition option) 5-97, 6-4, 6-10, 6-26, 6-66, 6-74, 6-78

POINT LOAD (history definition option) 5-58, 6-78

POINT LOAD (model definition option) 3-3, 5-97, 6-45, 6-78, 8-10

POINT SOURCE (model definition option) 6-50

Poisson's 5-3, 7-3, 7-4, 7-5, 7-7, 7-73, 7-126, 7-130, 7-133, 7-136, 7-139

POST (model definition option) 5-58, 6-4, 6-22, 11-7, 12-21

post buckling analysis 11-5

post file 12-21

POWDER (model definition option) 7-74

powder material 7-73

Power Law Fluid 6-59

preconditioners, iterative solvers 11-42

Previously Refined Mesh Criterion 4-40

Principal Engineering Strains 12-22

PRINT (parameter) 5-31, 12-20

PRINT CHOICE (model definition option) 5-88, 7-20, 12-9, 12-10, 12-12, 12-13, 12-15, 12-18

PRINT ELEMENT (model definition option) 7-20, 12-9, 12-12, 12-13, 12-15

PRINT NODE (model definition option) 12-9, 12-10, 12-12, 12-18

PRINT VMASS (model definition option) 12-10
program

initiation 2-9

messages 12-26

progressive composite failure 7-22

PROPORTIONAL INCREMENT (history definition option) 4-27, 5-58, 5-60, 9-9

PSPG 6-60

PWL (piece-wise linear) 8-6

Q

quarter point elements 5-67

R

radial return 5-46

RADIATING CAVITY (model definition option) 6-13

RADIATION (parameter) 6-13, 6-17, 7-158, 9-17

radiation viewfactors 6-12

RBE2 and RBE3 9-41

reaction forces 12-18

REAUTO (model definition option) 5-53, 11-17

REBAR (model definition option) 7-133

REBAR (user subroutine) 7-133

rebar elements 10-31

RECOVER (history definition option) 5-73

reduced integration elements 10-35

RELATIVE DENSITY (model definition option) 7-74

RELEASE (history definition option) 8-30

remeshing

criteria 4-44

global 4-41

A B C D E F G H I J K L M N O P Q R S T U V W X Y Z

remeshing techniques [4-46](#)
 mesh generation [4-46](#)
 outline extraction and repair [4-49](#)
 target number of elements [4-51](#)
Residual Load Correction [11-26](#)
residual load correction [11-16](#)
RESTART (model definition option) [2-6](#), [5-4](#),
 [5-53](#), [11-17](#), [11-18](#), [12-10](#), [12-12](#)
restart analysis [11-17](#)
RESTART INCREMENT (history definition option)
 [11-18](#)
RESTART LAST (model definition option) [11-18](#)
RESTRICTOR (model definition option) [6-26](#)
RESTRICTOR (parameter) [6-26](#)
result evaluation, contact [8-42](#)
REZONE (rezoning option) [3-10](#)
rezoning [4-25](#), [8-40](#)
REZONING (parameter) [4-25](#)
rigid contact, solution strategy [8-51](#)
rigid link constraint [9-36](#)
rigid-plastic analysis [5-90](#), [10-33](#), [12-19](#)
Riks-Ramm [11-32](#), [11-33](#)
ROTATION A (model definition option) [9-8](#)
R-P FLOW (parameter) [5-90](#), [11-26](#)
rubber foam model [7-56](#)
RVCNST [8-22](#), [8-23](#)

S

SCALE (parameter) [11-4](#)
scaling, linear-elastic solution [11-4](#)
Secant method [11-24](#)
semi-infinite elements [10-38](#)
sensitivity [5-102](#)
separation, contact [8-29](#)
SEPFOR (user subroutine) [8-52](#)
SEPSTR (user subroutine) [8-52](#)
Sequential Quadratic Programming [5-110](#)
SERVO LINK (model definition option) [9-2](#), [9-28](#)

shape memory material [7-43](#)
shear panel element [10-30](#)
shell
 contact [8-15](#)
 elements [8-38](#), [10-23](#)
 stiffener tying constraints [9-34](#)
 transformation [9-23](#)
SHELL SECT (parameter) [10-22](#)
shell stiffener tying constraints [9-34](#)
SHELL TRANSFORMATION (model definition
 option) [9-23](#), [9-26](#)
shell-to-shell intersection tying constraints [9-31](#)
shell-to-solid intersections tying constraints [9-32](#)
SHIFT FUNCTION (model definition option) [5-58](#),
 [7-116](#)
single step Houbolt operator [5-77](#)
SIZING (parameter) [8-40](#), [10-2](#), [12-6](#)
snap-through analysis [11-5](#)
SOIL (model definition option) [5-97](#), [7-136](#)
soil analysis [5-97](#), [7-136](#), [10-27](#)
solver
 conjugate gradient iterative [13-4](#)
 direct [13-3](#)
 hardware [13-4](#)
 matrix [13-3](#)
 multifrontal [13-3](#)
SOLVER (model definition option) [10-41](#), [11-41](#),
 [11-43](#)
special elements [10-30](#)
SPECIFIED NODES (model definition option)
 [4-15](#)
spectrum response [5-88](#)
SPLINE (model definition option) [8-6](#), [8-54](#)
SPRINGS (model definition option) [5-4](#), [5-29](#),
 [5-86](#)
springs, linear and nonlinear [12-17](#)
SSTRAN (user subroutine) [4-27](#)
START NUMBER (model definition option) [4-15](#)
STATE VARS (parameter) [6-26](#)

A B C D E F G H I J K L M N O P Q R S T U V W X Y Z

status file [12-31](#)

STEADY STATE (electrostatic, history definition option) [6-31](#)

STEADY STATE (history definition option) [6-6](#), [6-63](#), [6-74](#)

STEADY STATE (magnetostatic, history definition option) [6-34](#)

steady state rolling analysis [5-118](#)

Stefan-Boltzmann [9-17](#)

stick-slip model, friction [8-24](#)

STIFFNS COMPONENTS (history definition option) [6-27](#)

STIFFSCALE (model definition option) [4-29](#)

STOP (parameter) [2-5](#), [8-45](#)

storage methods [11-43](#)

strain correction method [11-22](#)

STRAIN RATE (model definition option) [5-92](#), [6-66](#), [7-7](#), [7-76](#)

STRAIN RATE (model definition, fluid option) [7-160](#)

strain rate effects on elastic-plastic constitutive relation [7-87](#)

stress

analysis [12-18](#)

intensity factor defined [5-62](#)

intensity factors [5-62](#)

SUBSTRUC (parameter) [4-27](#)

SUBSTRUCTURE (model definition option) [4-27](#)

substructuring [4-26](#)

SUMMARY (model definition option) [12-10](#)

SUPER (parameter) [4-27](#)

SUPERINPUT (model definition option) [4-27](#)

SUPG [6-60](#)

SURFACES (model definition option) [4-36](#)

symmetries [4-14](#)

SYNCHRONIZED (history definition option) [8-12](#)

system and element stiffness matrices [A-5](#)

T

TABLE (model definition option) [7-25](#), [11-8](#), [11-9](#)

TEMP CHANGE (history definition option) [6-4](#), [6-10](#), [6-66](#), [6-70](#), [6-74](#), [6-78](#)

TEMPERATURE EFFECTS (coupled thermal-stress, model definition option) [7-158](#)

TEMPERATURE EFFECTS (heat transfer, model definition option) [6-74](#), [7-158](#)

TEMPERATURE EFFECTS (hydrodynamic, model definition option) [7-158](#)

TEMPERATURE EFFECTS (model definition option) [5-92](#), [6-3](#), [6-8](#), [6-26](#), [7-7](#), [7-52](#), [7-74](#), [7-75](#), [7-76](#), [7-110](#), [7-116](#), [7-126](#), [7-127](#), [7-129](#), [7-160](#)

Temperature Gradient Criterion [4-40](#)

temperature-dependent

creep [7-128](#)

material properties [7-126](#)

tension softening [7-134](#)

thermal

loads [9-14](#)

loads input options [9-15](#)

THERMAL CONTACT (model definition option) [6-5](#)

thermal contact analysis [8-34](#)

thermal expansion coefficient [7-128](#)

thermo-electrical [7-158](#)

thermo-mechanical shape memory model [7-31](#)

constitutive theory [7-34](#)

phase transformation strains [7-35](#)

transformation induced deformation [7-32](#)

Thermo-Rheologically Simple material [7-114](#)

THICKNESS (model definition option) [6-26](#)

THICKNS CHANGE (history definition option) [6-26](#)

TIME STEP (history definition option) [7-76](#), [7-112](#), [7-119](#)

time step definition [5-82](#)

A B C D E F G H I J K L M N O P Q R S T U V W X Y Z

time-dependent

boundary conditions [5-83](#)

inelastic behavior [7-92](#)

time-independent, inelastic behavior [7-61](#)

TIME-TEMP (model definition option) [7-130](#),
[7-131](#)

Time-Temperature-Transformation (T-T-T)
[7-130](#)

tolerance values, contact [8-43](#)

total Lagrangian procedure [5-19](#)

TRANSFORMATION (model definition option)
[9-22](#), [9-32](#), [12-18](#)

TRANSIENT (history definition option) [6-4](#), [6-6](#),
[6-7](#), [6-10](#), [6-74](#), [11-5](#), [11-6](#)

TRSFAC (user subroutine) [7-116](#)

truss elements [10-18](#)

Tsai-Wu failure criterion [7-19](#)

TSHEAR (parameter) [7-20](#), [12-16](#)

T-T-T (parameter) [7-130](#)

TYING (model definition option) [6-4](#), [6-72](#), [9-2](#),
[9-28](#)

TYING CHANGE (history definition option) [9-28](#)

tying constraints

beam-to-beam intersection [9-32](#)

cyclic symmetry [9-38](#)

mesh refinement [9-29](#)

nodal degrees of freedom [9-34](#)

pipe bend element [9-33](#)

shell stiffener [9-34](#)

shell-to-shell intersection [9-31](#)

shell-to-solid intersections [9-32](#)

tying, shell-to-solid [9-36](#)

U

UADAP (user subroutine) [4-40](#)

UBEAR (user subroutine) [6-27](#)

UCOMPL (user subroutine) [5-87](#)

UCOORD (user subroutine) [4-36](#)

UCRACK (user subroutine) [7-133](#)

UCRPLW (user subroutine) [5-53](#), [5-56](#), [5-57](#),
[7-104](#), [7-128](#)

UELDAM (user subroutine) [7-157](#)

UENERG (user subroutine) [7-57](#)

UEPS (user subroutine) [6-31](#), [6-33](#), [6-39](#), [7-159](#)

UFAIL (user subroutine) [7-14](#), [7-20](#)

UFCONN (model definition option) [4-8](#)

UFCONN (user subroutine) [4-8](#)

UFINITE (user subroutine) [7-65](#)

UFORMS (user subroutine) [9-28](#), [9-35](#)

UFOUR (user subroutine) [5-6](#)

UFOXORD (user subroutine) [4-17](#)

UFRIC (user subroutine) [8-27](#)

UFXORD (model definition option) [4-8](#)

UFXORD (user subroutine) [4-8](#)

UGROOV (user subroutine) [6-26](#)

UHTCOE (user subroutine) [8-33](#), [8-34](#)

UHTCON (user subroutine) [8-33](#), [8-34](#)

UHTNRC (user subroutine) [8-33](#), [8-34](#)

UMOONY (user subroutine) [7-52](#)

UMU (user subroutine) [6-39](#), [7-159](#)

UNEWTN (user subroutine) [5-92](#)

uniaxial cracking data [7-133](#)

UOGDEN (user subroutine) [7-53](#)

UPDATE (parameter) [5-16](#), [5-18](#), [5-23](#), [5-36](#),
[5-77](#), [6-71](#), [7-30](#), [7-74](#), [7-78](#), [7-139](#), [9-26](#)

updated Lagrange Formulation, nonlinear
elasticity [7-59](#)

updated Lagrangian Procedure [5-23](#)

UPNOD (user subroutine) [5-91](#)

UPOSTV (user subroutine) [12-21](#)

UPSTRECH (user subroutine) [7-57](#)

URESTR (user subroutine) [6-26](#)

URPFLO (user subroutine) [5-92](#)

User subroutine

ANELAS [5-4](#), [7-5](#), [7-8](#)

ANEXP [5-4](#), [7-8](#), [7-129](#)

ANKOND [6-4](#), [7-158](#)

A B C D E F G H I J K L M N O P Q R S T U V W X Y Z

ANPLAS 7-8, 7-66
 CRPLAW 5-53, 5-54, 5-55, 5-59, 5-60, 7-97,
 7-98, 7-102, 7-105, 7-107, 7-128
 CRPVIS 5-59, 7-96, 7-107
 ELEVAR 12-11, 12-15
 ELEVEC 12-11
 FILM 6-4, 6-10, 6-11, 6-12, 6-74, 6-78, 9-17
 FLUX 6-4, 6-10, 6-11, 6-26, 6-31, 6-34, 6-45,
 6-50, 6-74, 6-78, 9-17
 FORCDT 5-83, 6-10, 6-11, 6-27, 6-31, 6-34,
 6-45, 6-66
 FORCEM 4-36, 5-83, 6-39, 6-45
 HOOKLW 5-4, 7-5, 7-8
 HOOKVI 7-113
 HYPELA 5-20, 7-30, 7-136
 HYPELA2 7-30, 7-136
 IMPD 12-11
 INTCRD 12-11
 MOTION (2-D) 8-10
 NASSOC 5-59, 5-60, 7-105
 ORIENT 5-4, 7-5, 7-10, 7-13, 7-66, 7-113
 PLOTV 12-21
 REBAR 7-133
 SEPFOR 8-52
 SEPSTR 8-52
 SSTRAN 4-27
 TRSFAC 7-116
 UADAP 4-40
 UBEAR 6-27
 UCOMPL 5-87
 UCOORD 4-36
 UCRACK 7-133
 UCRPLW 5-53, 5-56, 5-57, 7-104, 7-128
 UELDAM 7-157
 UENERG 7-57
 UEPS 6-31, 6-33, 6-39, 7-159
 UFAIL 7-14, 7-20
 UFCONN 4-8
 UFINITE 7-65

UFORMS 9-28, 9-35
 UFOUR 5-6
 UFRIC 8-27
 UFXORD 4-8, 4-17
 UGROOV 6-26
 UHTCOE 8-33, 8-34
 UHTCON 8-33, 8-34
 UHTNRC 8-33, 8-34
 UMOONY 7-52
 UMU 6-39, 7-159
 UNEWTN 5-92
 UOGDEN 7-53
 UPNOD 5-91
 UPOSTV 12-21
 UPSTRECH 7-57
 URESTR 6-26
 URPFLO 5-92
 USIGMA 6-39, 7-159
 USINC 5-83, 6-10
 USPRNG 5-29, 9-38
 USSD 5-89
 UTHICK 6-26, 6-27
 UTIMESTEP 11-13
 UTRANS 9-22
 UVELOC 6-4, 6-23, 6-26
 UVSCPL 5-54, 5-55, 5-60, 7-106
 VSWELL 5-54, 5-55, 7-103, 7-107
 WKSLP 7-76, 7-78, 7-80, 7-82
 YIEL 5-60, 7-105
 ZERO 5-60, 7-105
 user subroutine input 4-8
 User-defined Criterion 4-40
 user-defined failure criteria 7-20
 USIGMA (user subroutine) 6-39, 7-159
 USINC (user subroutine) 5-83, 6-10
 USPRNG (user subroutine) 5-29, 9-38
 USSD (user subroutine) 5-89
 UTHICK (user subroutine) 6-26, 6-27
 UTIMESTEP (user subroutine) 11-13

A B C D E F G H I J K L M N O P Q R S T U V W X Y Z

UTRANFORM (model definition option) [9-22](#)
UTRANS (user subroutine) [9-22](#)
UVELOC (user subroutine) [6-4](#), [6-23](#), [6-26](#)
UVSCPL (user subroutine) [5-54](#), [5-55](#), [5-60](#),
[7-106](#)

V

VELOCITY (model definition option) [6-4](#), [6-23](#),
[6-26](#)
VELOCITY CHANGE (history definition option)
[6-4](#), [6-23](#), [6-66](#)
VIEW FACTOR (model definition option) [6-17](#),
[9-17](#)
viewfactor calculation
 Adaptive Contour Integration [6-16](#)
 Direct Adaptive Integration [6-12](#)
 Monte Carlo Method [6-16](#)
VISCEL EXP (model definition option) [5-59](#),
[7-124](#)
VISCELMOON (model definition option) [5-59](#),
[7-52](#), [7-119](#)
VISCELOGDEN (model definition option) [5-59](#),
[7-53](#), [7-119](#), [7-157](#)
VISCELOORTH (model definition option) [7-112](#)
VISCELPROP (model definition option) [7-112](#)
VISCO ELAS (parameter) [5-59](#)
viscoelastic material
 incompressible isotropic [7-113](#)
 isotropic [7-110](#)
 Thermo-Rheologically Simple [7-114](#)
viscoelasticity [5-58](#)
viscoplasticity
 explicit formulation [7-104](#), [7-106](#)
 explicit method [5-59](#)
 implicit method [5-60](#)
VOLTAGE (model definition option) [6-74](#), [6-77](#)
VOLTAGE CHANGE (history definition option)
[6-74](#), [6-78](#)

von Mises [5-46](#), [5-47](#), [5-55](#), [5-56](#), [5-107](#), [7-64](#),
[7-68](#), [7-73](#), [7-78](#), [7-79](#), [7-80](#), [7-83](#), [7-84](#), [7-86](#),
[7-87](#), [7-99](#), [12-10](#), [12-13](#), [12-14](#)
VSWELL (user subroutine) [5-54](#), [5-55](#), [7-103](#),
[7-107](#)

W

weighted mesh [4-14](#)
Williams-Landel-Ferry (WLF) [5-58](#), [7-116](#)
WKSLP (user subroutine) [7-76](#), [7-78](#), [7-80](#), [7-82](#)
WORK HARD (model definition option) [5-92](#), [7-7](#),
[7-76](#), [7-78](#), [7-80](#), [7-82](#)
workhardening
 isotropic [7-78](#)
 rules [7-76](#)
 slope [7-62](#)
workspace allocation [2-3](#), [8-45](#)

Y

YIEL (user subroutine) [5-60](#), [7-105](#)
Young's [7-2](#), [7-3](#), [7-7](#), [7-21](#), [7-58](#), [7-63](#), [7-78](#),
[7-126](#), [7-130](#), [7-133](#), [7-134](#), [7-136](#)

Z

ZERO (user subroutine) [5-60](#), [7-105](#)
Zienkiewicz-Zhu Creep Strain Criterion [4-39](#)
Zienkiewicz-Zhu Criterion [4-38](#)

