Finite Element Methods In Linear Structural Mechanics

Univ. Prof. Dr. Techn. G. MESCHKE

SHORT PRESENTATION IN

ADAPTIVE FINITE ELEMENT

By

Abdullah ALSAHLY   Shorash MIRO

Computational Engineering

Ruhr Universität Bochum

Outline

Introduction

- Motivation and general concepts
- Major steps of Adaptive finite element

Error Measures and Adaptivity

- Modeling and Errors
- Errors control and Accuracy

Adaptive Mesh Refinement

- Mesh adaptation
- Refinement options

Motivation and general concepts

- Engineering problems are complex ⇒ no analytical solutions are available for real-life problems ⇒ **numerical methods**

  ⇒ **need to** mesh the domain / need to pick a method (FEM) to discretize the PDEs.

- Generally speaking, with more elements in a mesh, the solution is more precise:
  - There are more nodes that are available for calculating response and thus the solution is more precise
  - More elements means smaller elements so discretization error is minimized

Major steps of Adaptive finite element

- The usual finite element analysis would proceed from the selection of a mesh and basis to the generation of a solution to an accuracy appraisal and analysis. Experience is the traditional method of determining whether or not the mesh and basis will be optimal or even adequate for the analysis at hand.

- Accuracy appraisals typically require the generation of a second solution on a finer mesh or with a higher order polynomial and a comparison of the two solutions

Major steps of Adaptive finite element

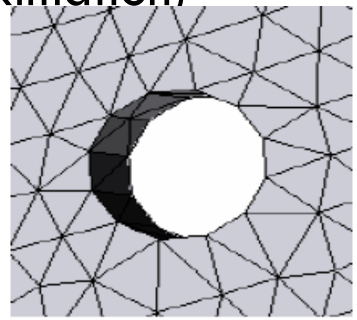
- Adaptive procedures try to automatically refine, coarsen or relocate a mesh and/or adjust the basis to achieve a solution having a specified accuracy in an optimal fashion

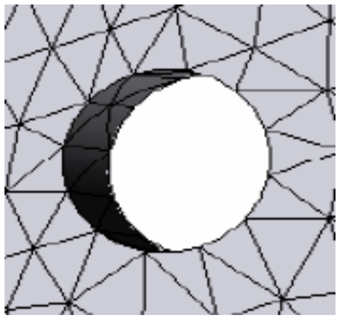
- In order to make Algorithm we must specify
  - a discretization method
  - a solver for the discrete problems
  - an error estimator
  - a refinement strategy which determines which elements have to be refined or coarsened and how this has to be done.

Modeling and Errors

Errors :-

- Modelling Errors
  - Errors due to assumptions in mathematical model
  - Errors related to data uncertainty and model accuracy
- Numerical Discretization Error (i.e. due to piecewise approximation)
  - The calculated shape of an element edge is limited by it's order (Linear = 1st Order or Parabolic = 2nd Order)
  - Therefore, curvature, either initial or calculated, must be approximated with straight facets in a linear (1st Order) mesh and with 2nd order segments in a high quality mesh
  - The difference between the ideal curvature and the modeled curvature is called **Discretization Error**





Errors control and Accuracy

- The computation typically begins with a trial solution generated on a coarse mesh with a low-order basis. The error of this solution is appraised. If it fails to satisfy the prescribed accuracy, adjustments are made with the goal of obtaining the desired solution with minimal effort. For example, we might try to reduce the discretization error to its desired level using the fewest degrees of freedom
- A posteriori error estimates in a particular norm were computed by summing their elemental contributions as

$$||E||^2 = \sum_{e=1}^{N_\Delta} ||E||_e^2$$

where N_Δ is the number of elements in the mesh and $||E||_e^2$ is the restriction of the erro $||E||^2$ mate to Element e

Errors control and Accuracy

- The most popular method of determining where adaptivity is needed is to use $||E||_e$ as an enrichment indicator.
- Thus, we assume that large errors come from regions where the local error estimate $||E||_e$ is large and this is where we should refine or concentrate the mesh and/or increase the method order.
- Correspondingly, the mesh would be coarsened or the polynomial degree of the basis lowered in regions where $||E||_e$ is small.
- The practical limit where further mesh size reductions add no benefit to the solution can be found in the **Convergence process.**

Mesh adaptation

- A **numerical solution can be improved by**
  – Either refining the grid (i.e., putting more mesh cells)
  – Or increasing the local expansion order (i.e., polynomial order)
  – Or both ! (a finer mesh represents a larger solution space)
- How to **select meshes (cells) for refinement???**
  – The refinement **must be driven by the accuracy required (user's choice)**
  – We will need several cycles to reach the final mesh adaptively from an initial mesh
- Additionally, **do we need accuracy everywhere in the mesh ???**
  – Not always, obtaining a highly accurate solution everywhere may not be desired from an engineering Point of View .

Refinement options

The **refinements can either be**

- **Uniform :**

  All mesh cells are divided, polynomial order is uniformly increased.

- **Selective:**

  Only selected cells will be divided/will have their polynomial order increased.

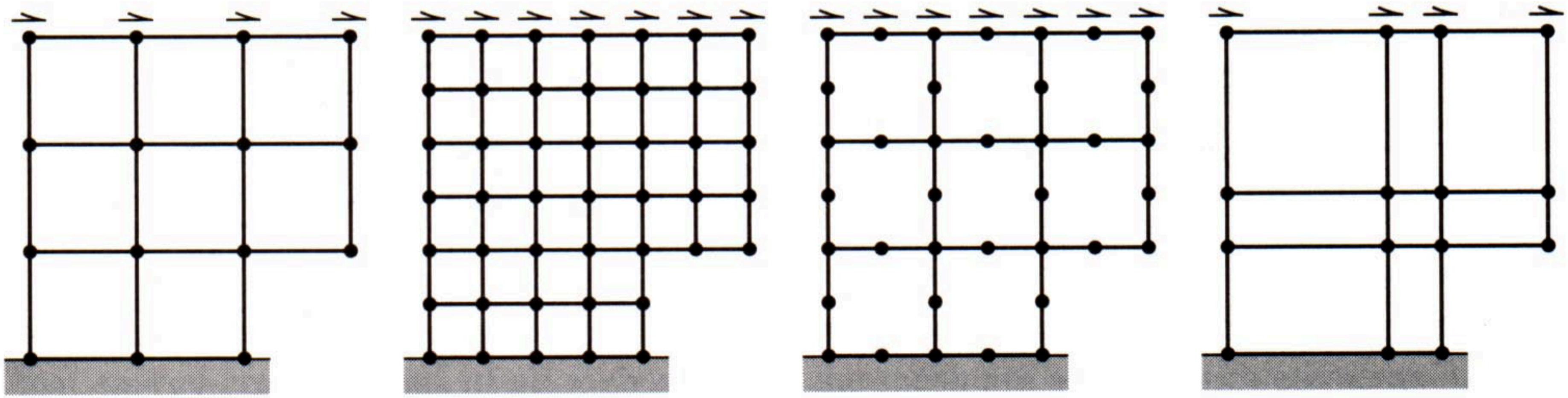
- **Remark**

  Uniform refinement is CPU expensive, not the best use of an engineer's time !!!

Mesh adaptation

- Generally, there are four ways to refine a mesh:

  1. ***h-refinement*** *(changing the element size)*
  2. ***p-refinement*** *(changing to higher* order polynomial interpolations)
  3. ***hp-refinement*** *(combination of h and p refinements)*
  4. ***r-refinement*** *(keep the number of nodes constant and adjust their positions)*



Original mesh — *A* uniform *h*-refinement — *A* uniform *p*-refinement — *A* possible *r*-refinement

Refinement options

h-refinement:

- Selective subdivision of the mesh, the polynomial order is kept fixed.

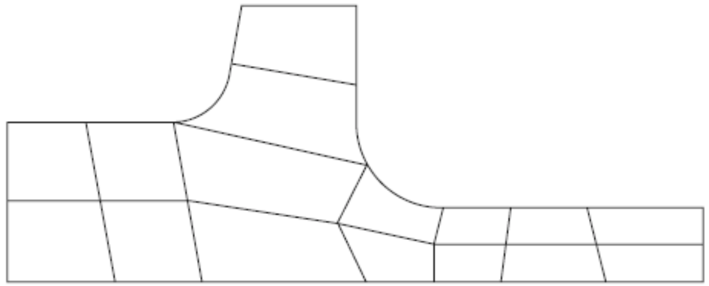
We will illustrate two typical methods of h-refinement:

1- Element subdivision (enrichment) in which the existing elements are simply divided into smaller ones keeping the original element boundaries intact.

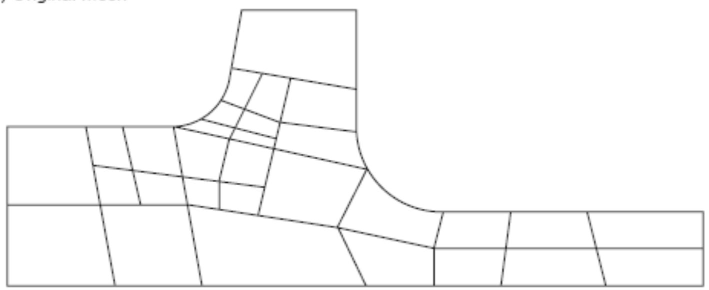
2- Mesh regeneration (Remeshing) here, on the basis of a given solution, a new element size is predicted in all the domain and a totally new mesh is generated

Refinement options

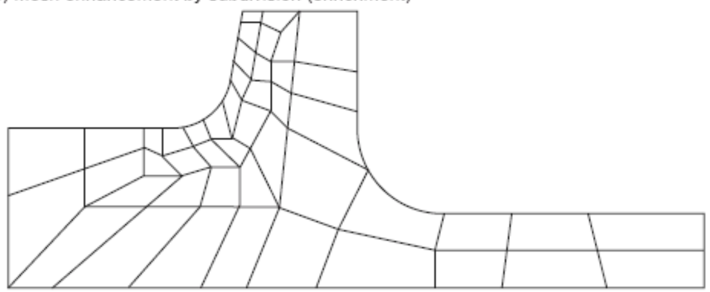
h-refinement:



(a) Original mesh



(b) Mesh enhancement by subdivision (enrichment)



(c) Mesh enhancement by remeshing

Refinement options

h-refinement:

- **How to predict required element size in h-adaptivity ?**

To decide which element size is required and where,

there are many strategies and procedures, one of them

and may be the simplest one is :

making the relative energy norm percentage

$\eta = \frac{||\mathbf{e}||}{||\mathbf{u}||} \times 100\%$ less than some specified value $\bar{\eta}$

here we have used the following formula to get the error

$$||\mathbf{e}||^2 = ||\mathbf{u}||^2 - ||\hat{\mathbf{u}}||^2$$

Refinement options

h-refinement:

So, the permissible error $\bar{\eta}||\mathbf{u}|| \approx \bar{\eta}\left(||\hat{\mathbf{u}}||^2 + ||\mathbf{e}||^2\right)^{1/2}$

Accordingly, the error in any element **k** should be

$$||\mathbf{e}||_k < \bar{\eta}\left(\frac{||\hat{\mathbf{u}}||^2 + ||\mathbf{e}||^2}{m}\right)^{1/2} \equiv \bar{e}_m$$

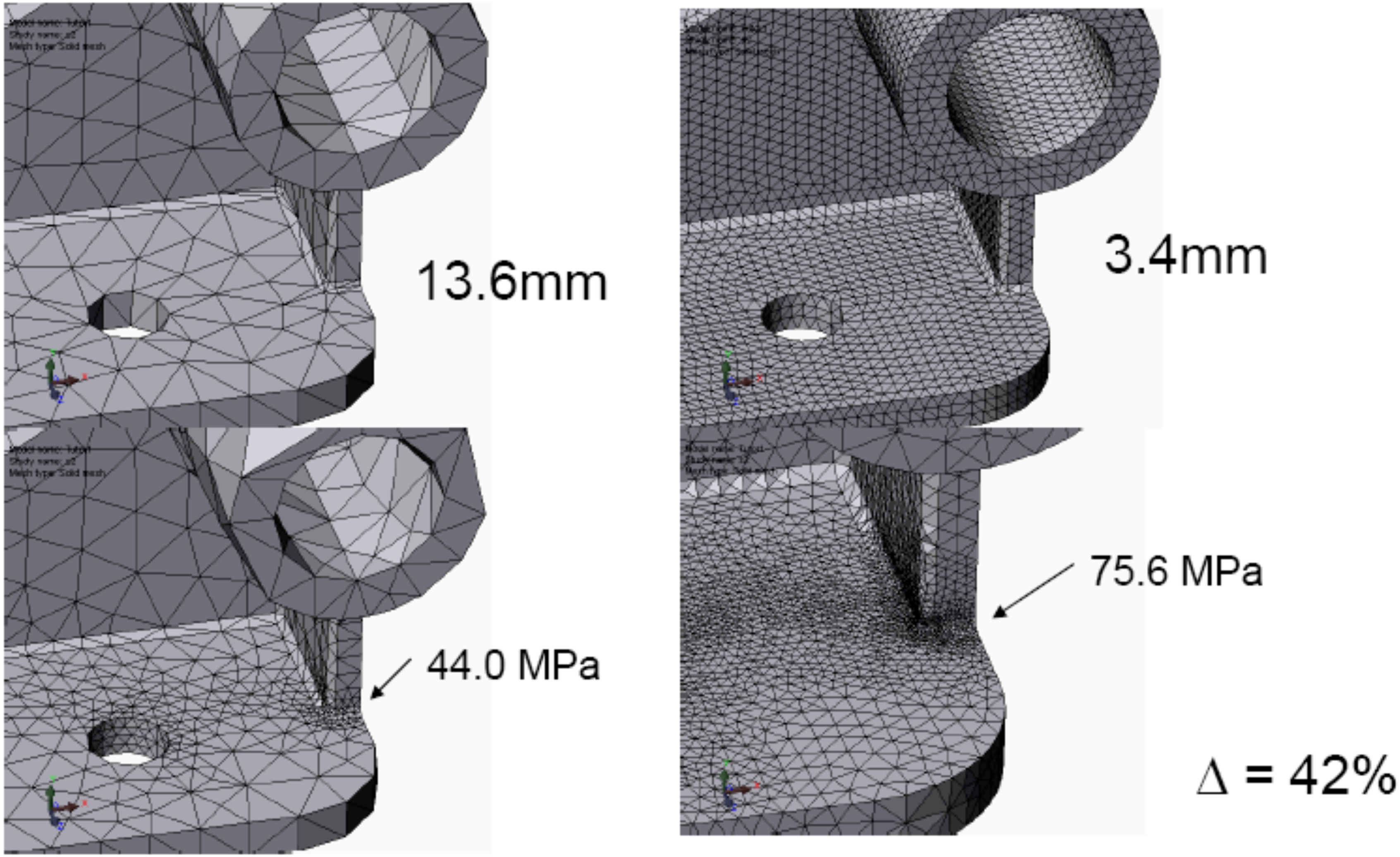
Element in which the previous formula is not satisfied are candidates for refinement, thus if we define the ratio

$$\frac{||\mathbf{e}||_k}{\bar{e}_m} = \xi_k$$

we shall refine whenever

$$\xi_k > 1$$

h-refinement (numerical example with linear approximation)



h-refinement (numerical example)

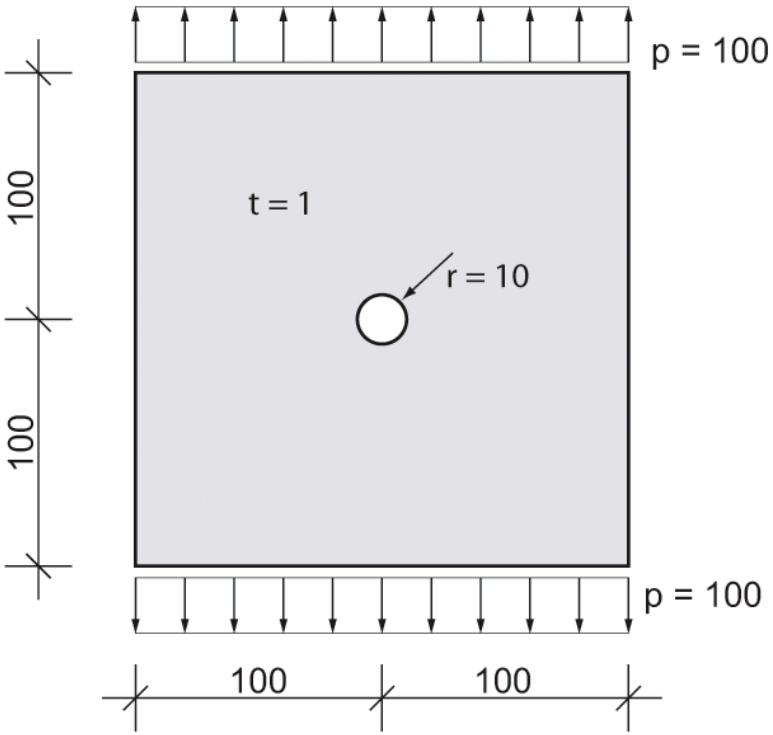
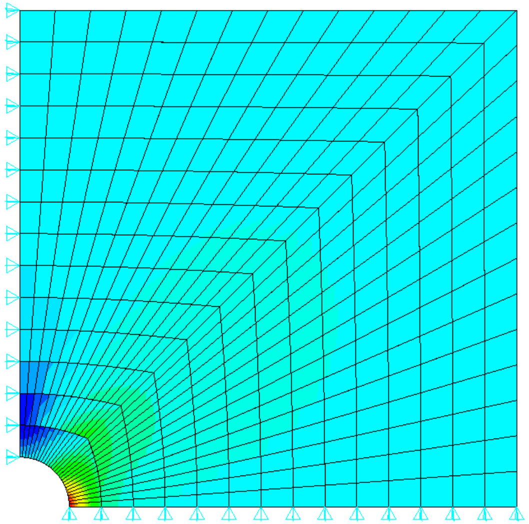
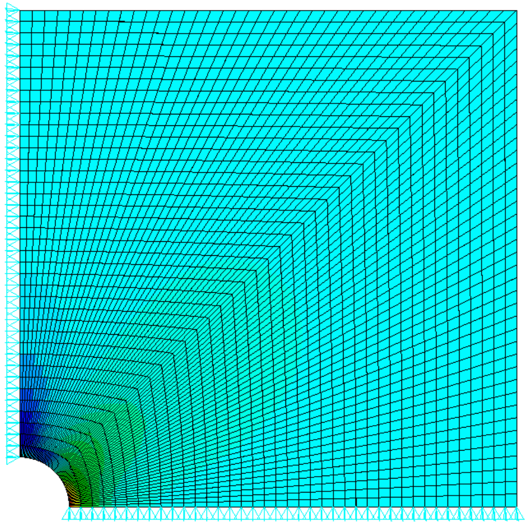


Plate with circular void



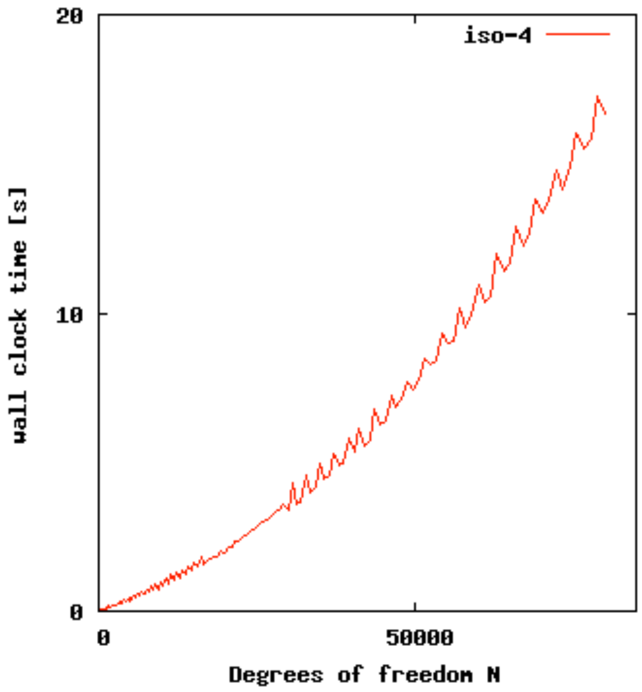
Coarse mesh



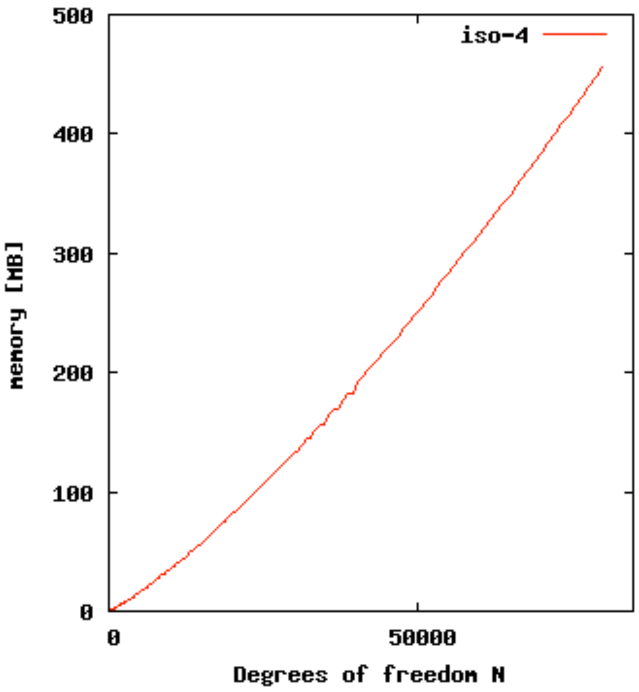
Refined mesh

h-refinement (numerical example)

Uniform refinement is CPU expensive, not the best use of an engineer's time !!!



Computation time



Memory

h-refinement (remarks)

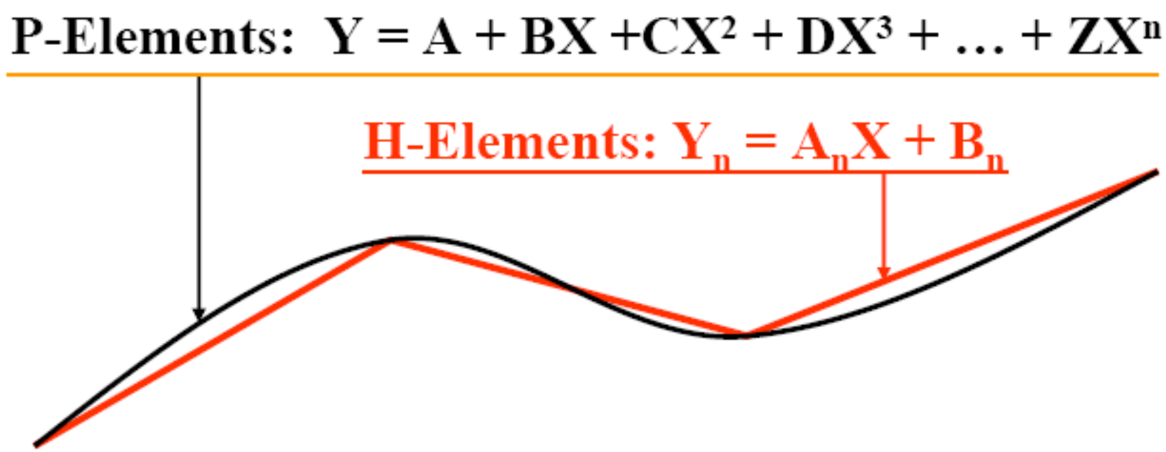
- The mesh should be refined until convergence is achieved (i.e. the results change very little from the previous refinement).

- Not optimal for regions where the solution is smooth (e.g., if the true solution is quadratic, a linear approximation would still require some mesh subdivision to be performed)

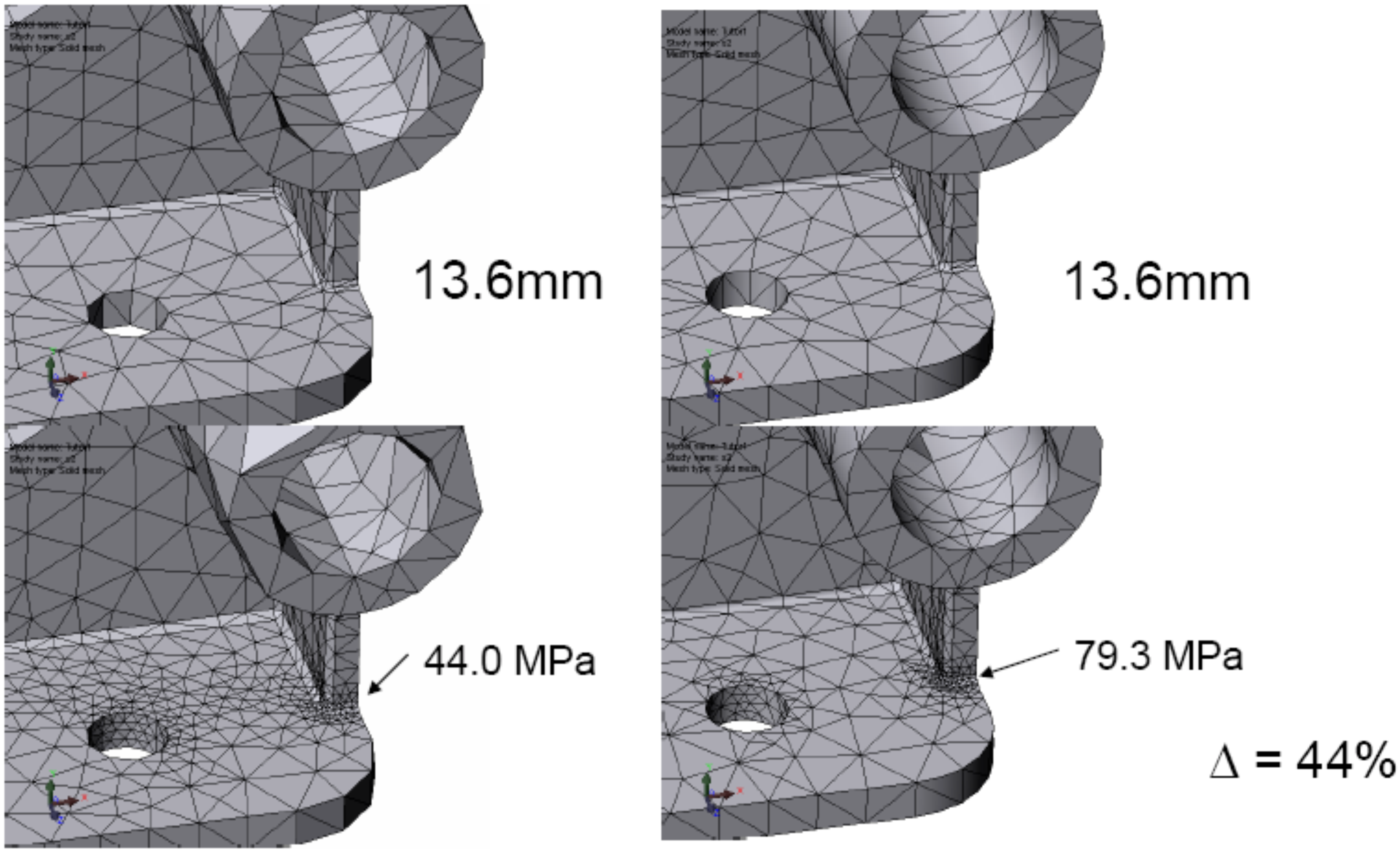
Refinement options

p-refinement:

- The initial mesh is kept unchanged, selective increase in the polynomial order
- Good for computing a smooth solution with large mesh cells
- Can capture more deformation with larger elements but are more computationally intensive than h-elements



p-refinement (numerical example)

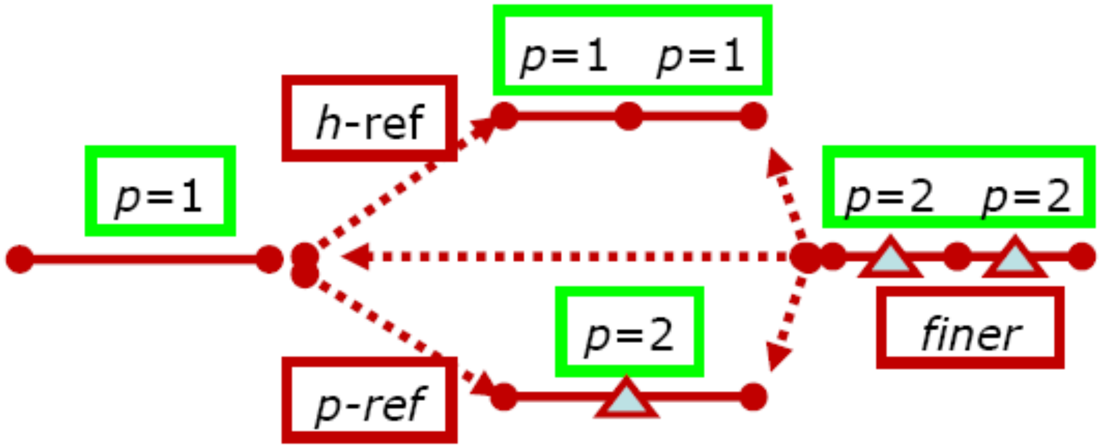


Refinement options

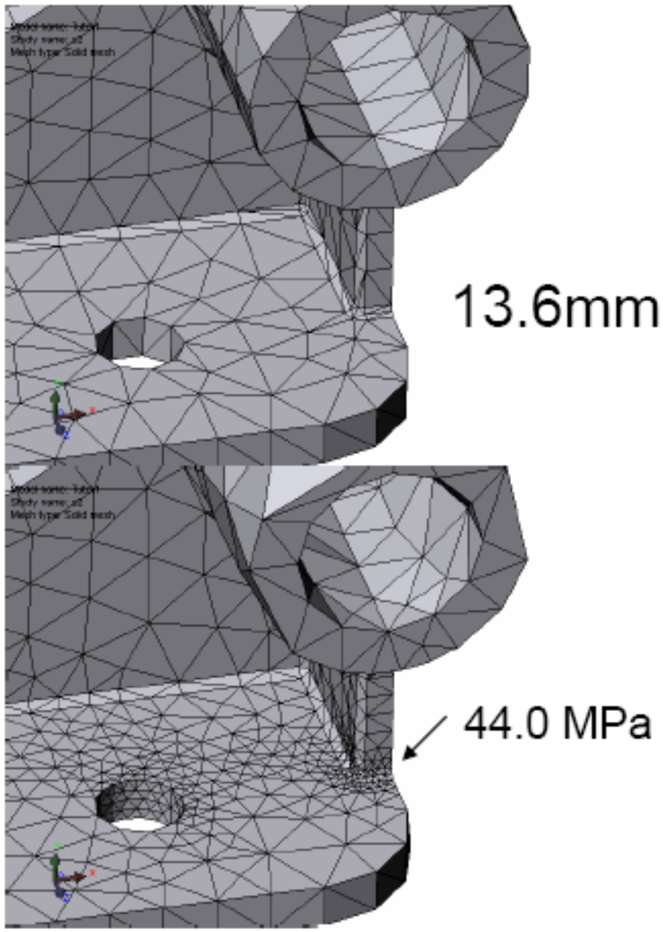
hp-refinement:

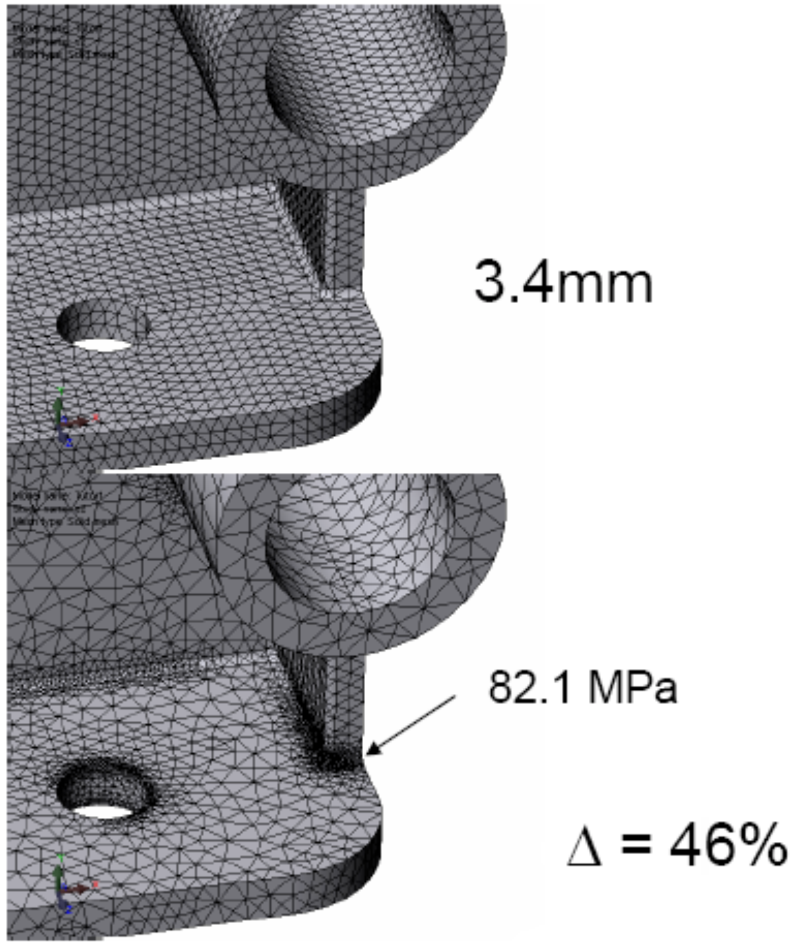
- the element size *h* as well as the polynomial degree *p* are adapted .

- Illustration of hp-refinement :



hp-refinement (numerical example)





Linear approximation

Higher order approximation

Conclusion

”Do not forget that you can eat an elephant if you slice it into small enough portions” (The principle of FEM)

Russian saying