Computational Biomechanics 2017

Lecture 3:

Intro to FEA

Ulli Simon, Martin Pietsch, Lucas Engelhardt

Scientific Computing Centre Ulm, UZWR
Ulm University

Contents

Detailed Schedule Summer 2017

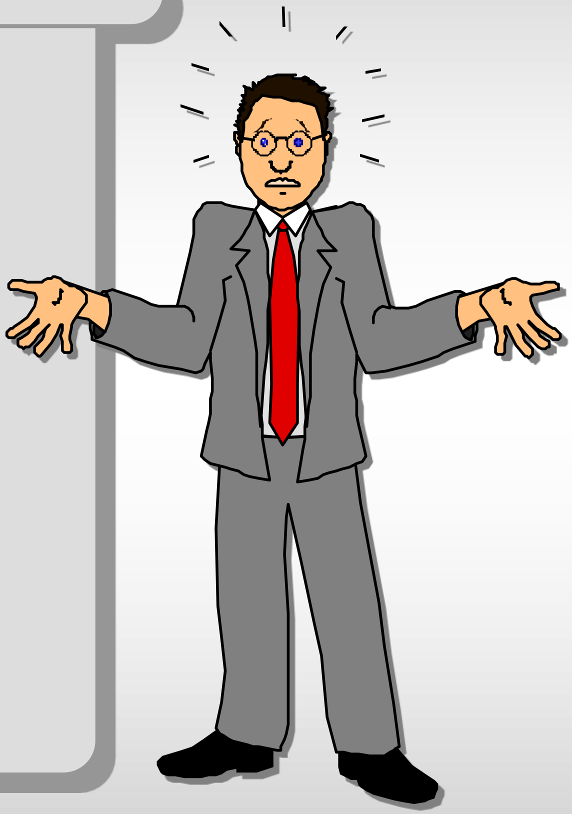
SW	Day	Date	Topic	Lecturer
01	Mo	17 Apr	- Holiday -	-
02	Mo	24 Apr	L01: Intro to Biomechanics; Mech 1: Statics	Ulli
03	Mo	01 May	- Holiday -	-
04	Mo	08 May	L02: Mech 2: Elastostatics; Lab: Intro to Ansys WB	Ulli
05	Mo	15 May	L03: Mech 3: Mat. Props. Biol. Tissues, Intro FEA, Lab: Trabec. B. 1	Ulli

FE Explanation in one sentence

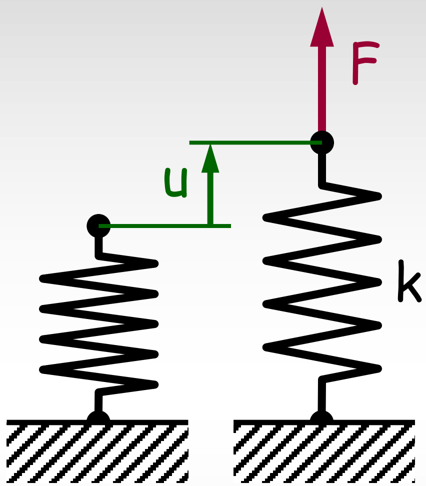
Finite Element Methode

=

Numerical Method
to solve partial differential
equations (PDEs) approximately

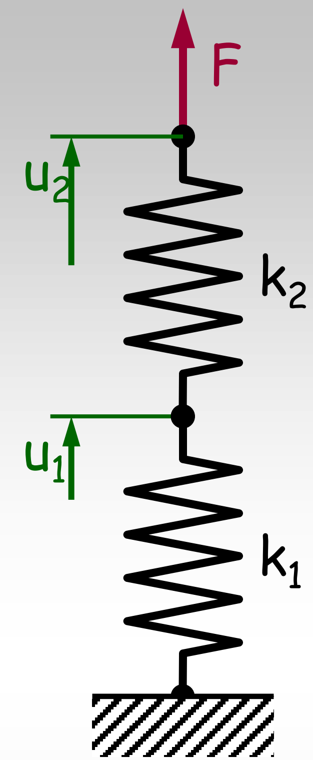


FE Explanation on one slide



$$k \cdot u = F$$

$$u = k^{-1} F$$



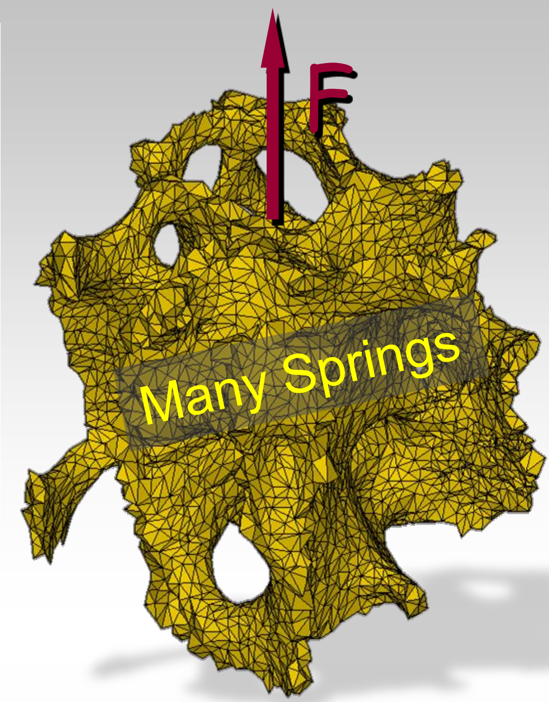
$$k_1 u_1 = k_2 (u_2 - u_1)$$

$$k_2 (u_2 - u_1) = F$$

$$\underbrace{\begin{bmatrix} k_1 + k_2 & -k_2 \\ -k_2 & k_2 \end{bmatrix}}_{\underline{\underline{K}}} \cdot \underbrace{\begin{bmatrix} u_1 \\ u_2 \end{bmatrix}}_{\underline{u}} = \underbrace{\begin{bmatrix} 0 \\ F \end{bmatrix}}_{\underline{F}}$$

$$\underline{u} = \underline{\underline{K}}^{-1} \underline{F}$$

$\Leftrightarrow$



FE-Software

$$\underline{\underline{K}} \cdot \underline{u} = \underline{F}$$

FE-Software

$$\underline{u} = \underline{\underline{K}}^{-1} \underline{F}$$

Fields

Dynamics
- Implicit: Modal analysis
- Explicit: transient time dependent (crash)

Statics, Elasticity
- Stresses, Strains

Nonlinearities:
- Contact, Friction
- Plasticity, Hardening
- Fatigue, Fracture mechanics
- Shape optimization

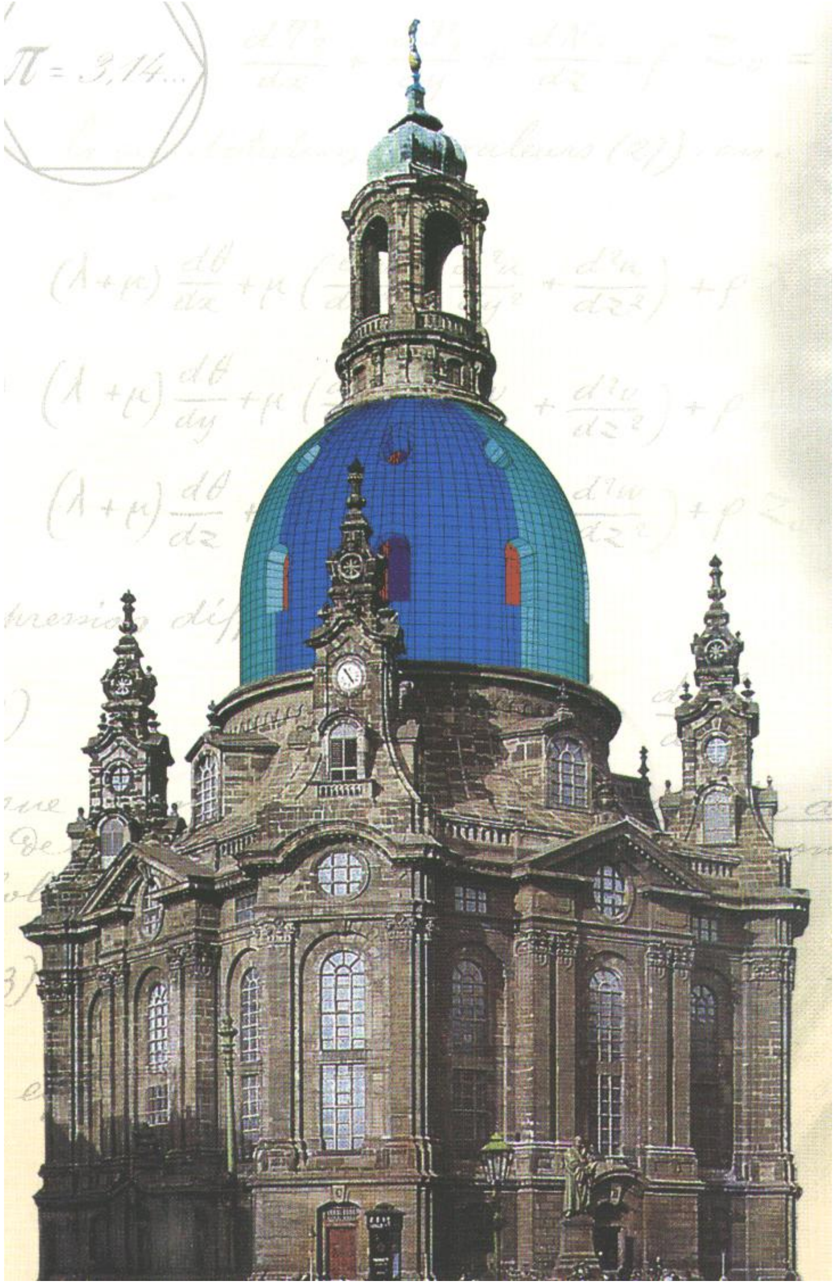
Accustics

FEM

Heat Transfer, Diffusion

Fluid flow
- Air planes
- Weather prediction

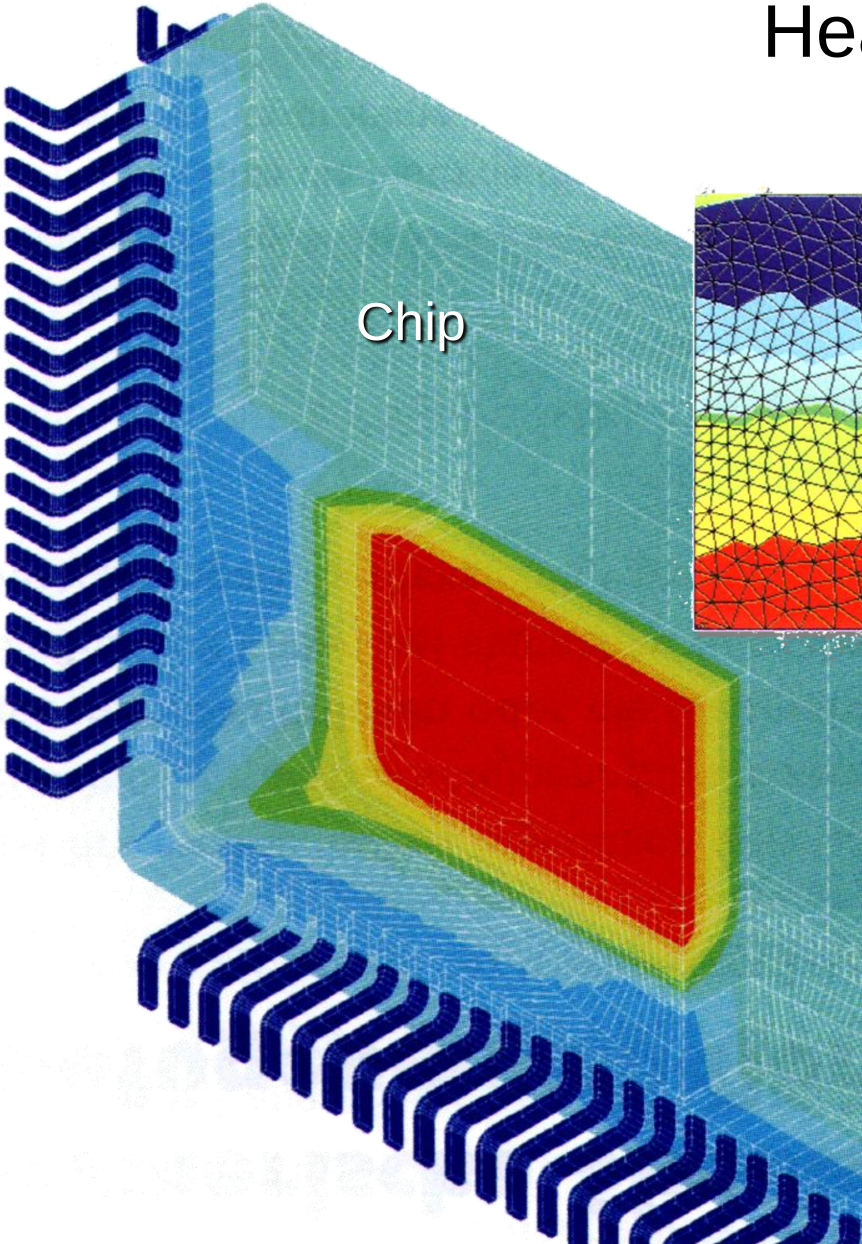
Electromagnetic fields

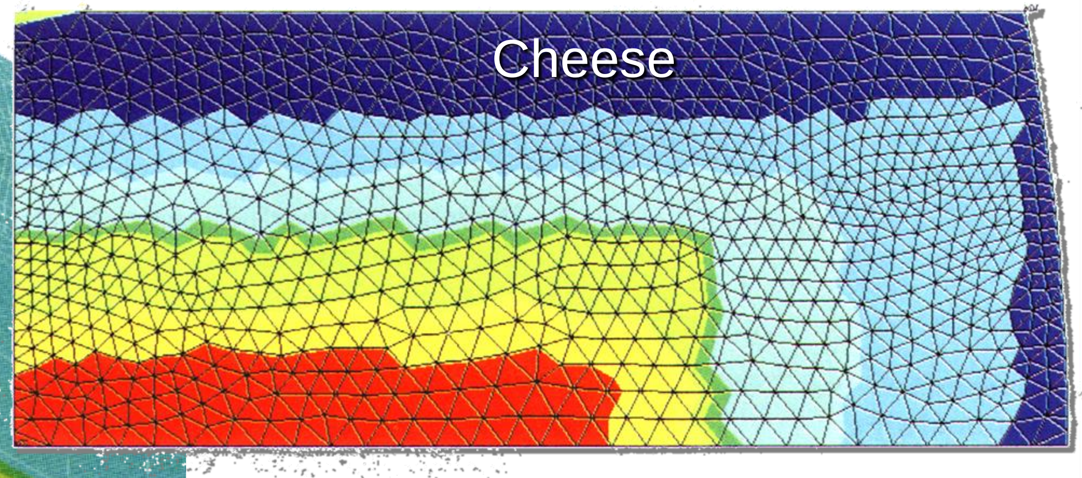


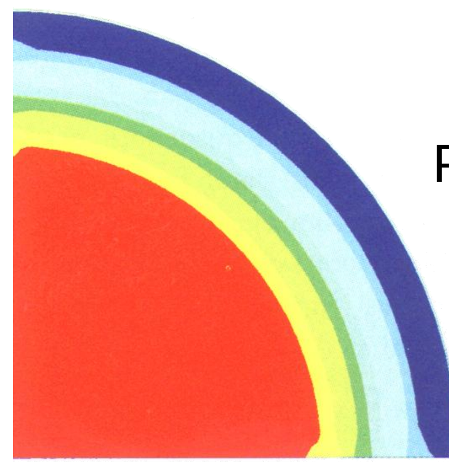
Statics, Elasticity

Frauenkirche, Dresden

Heat Transfer and Diffusion

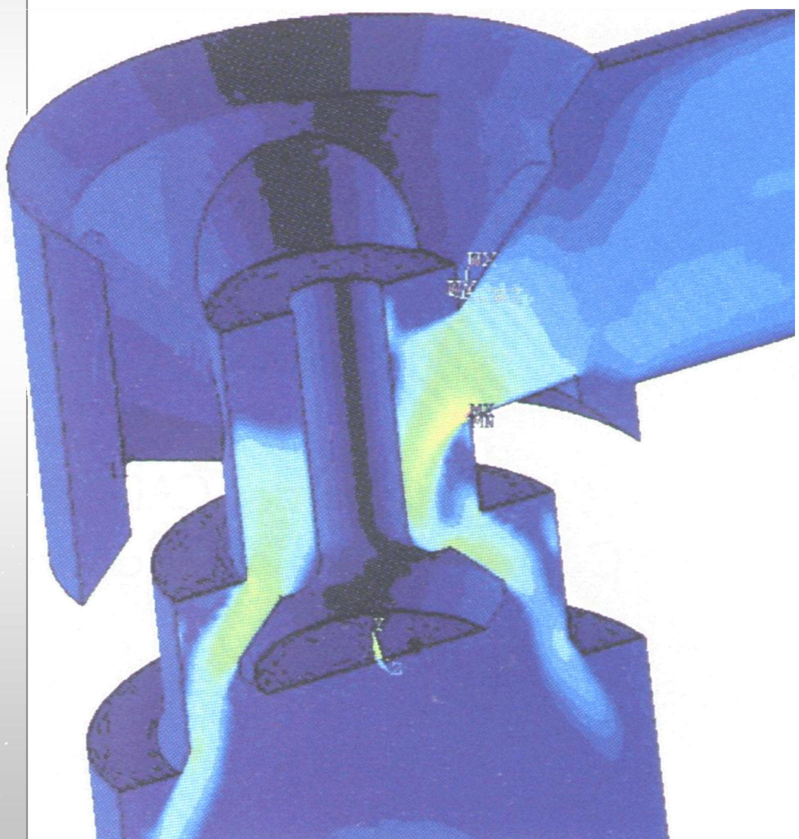


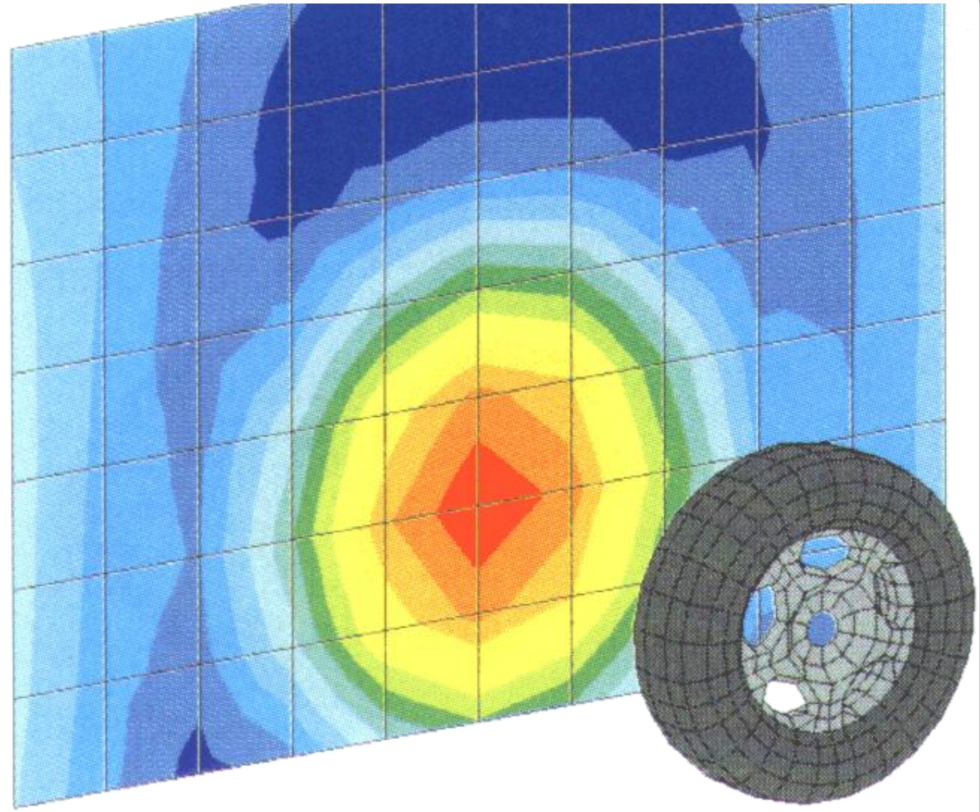




Pasta

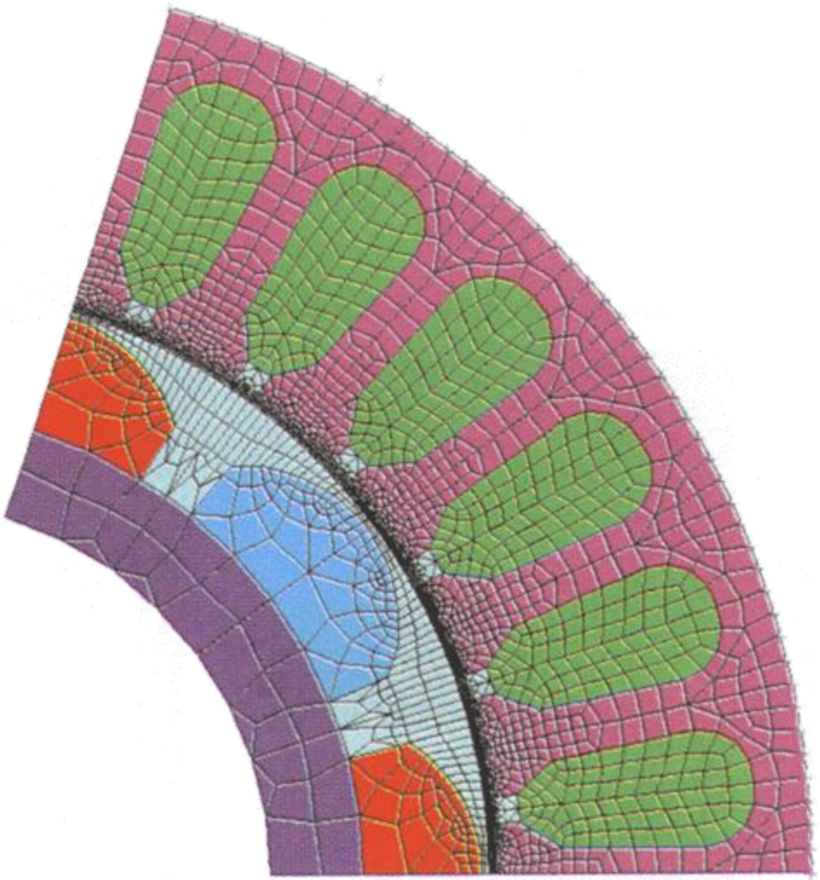
Fluid Flow

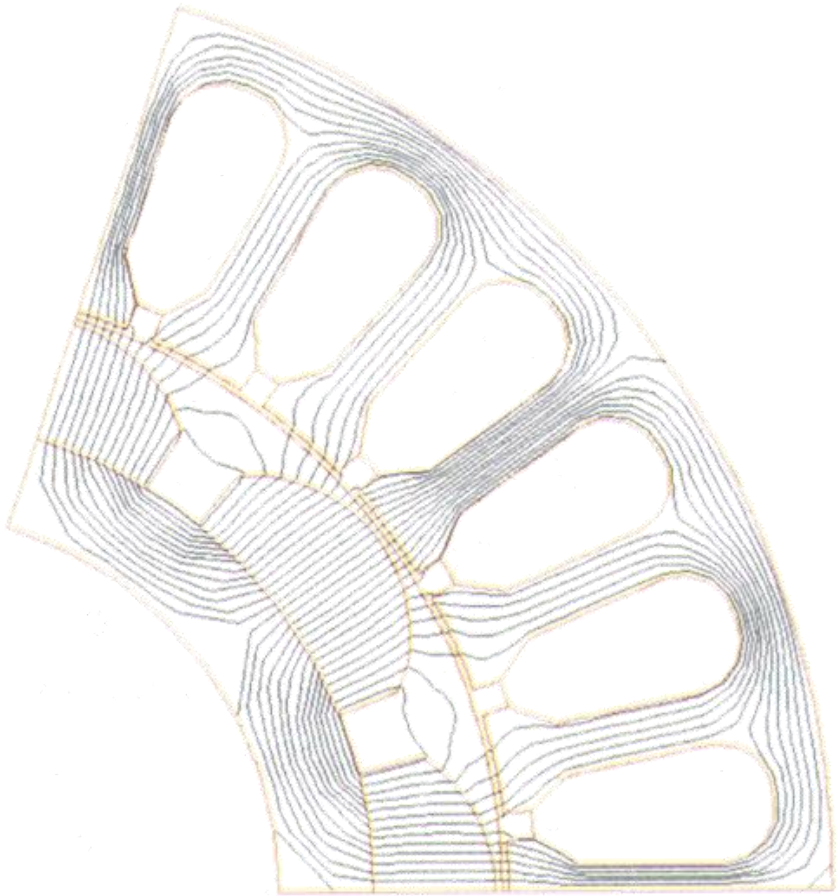




Accustics

Electromagnetic Fields

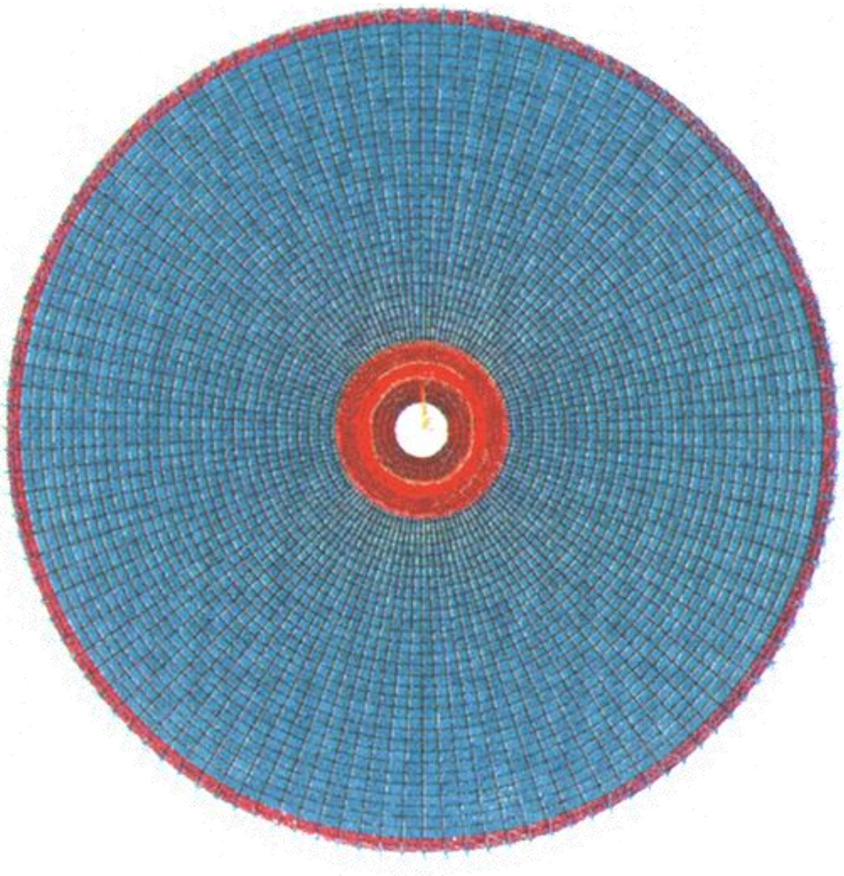


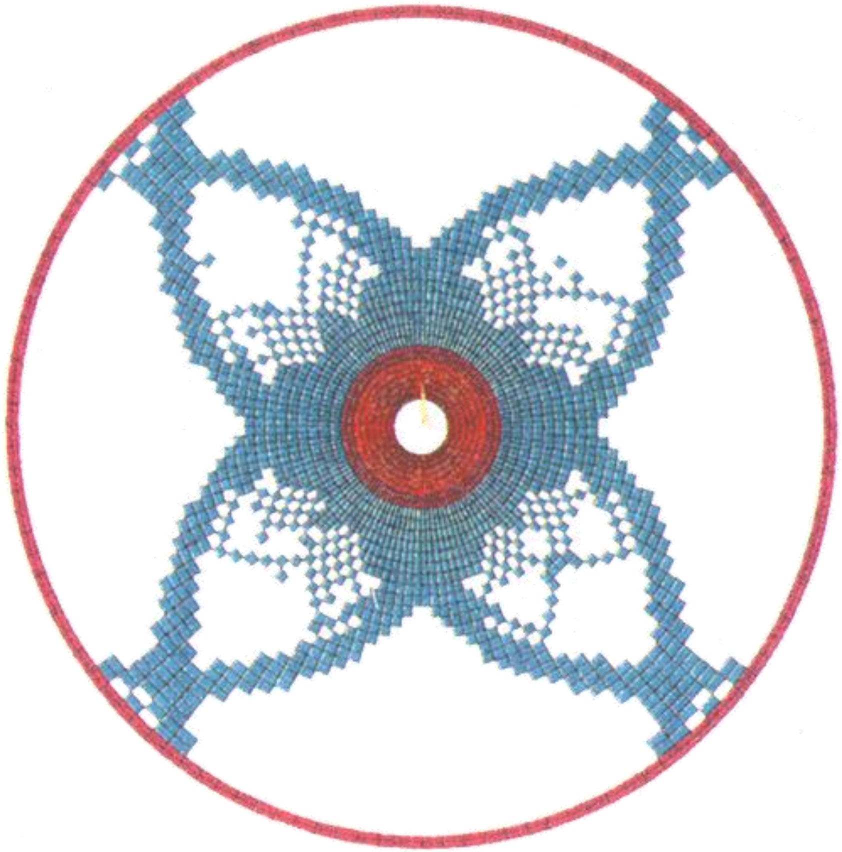


Model: electric motor

Solution: streamlines of magnetic flow

Shape optimization



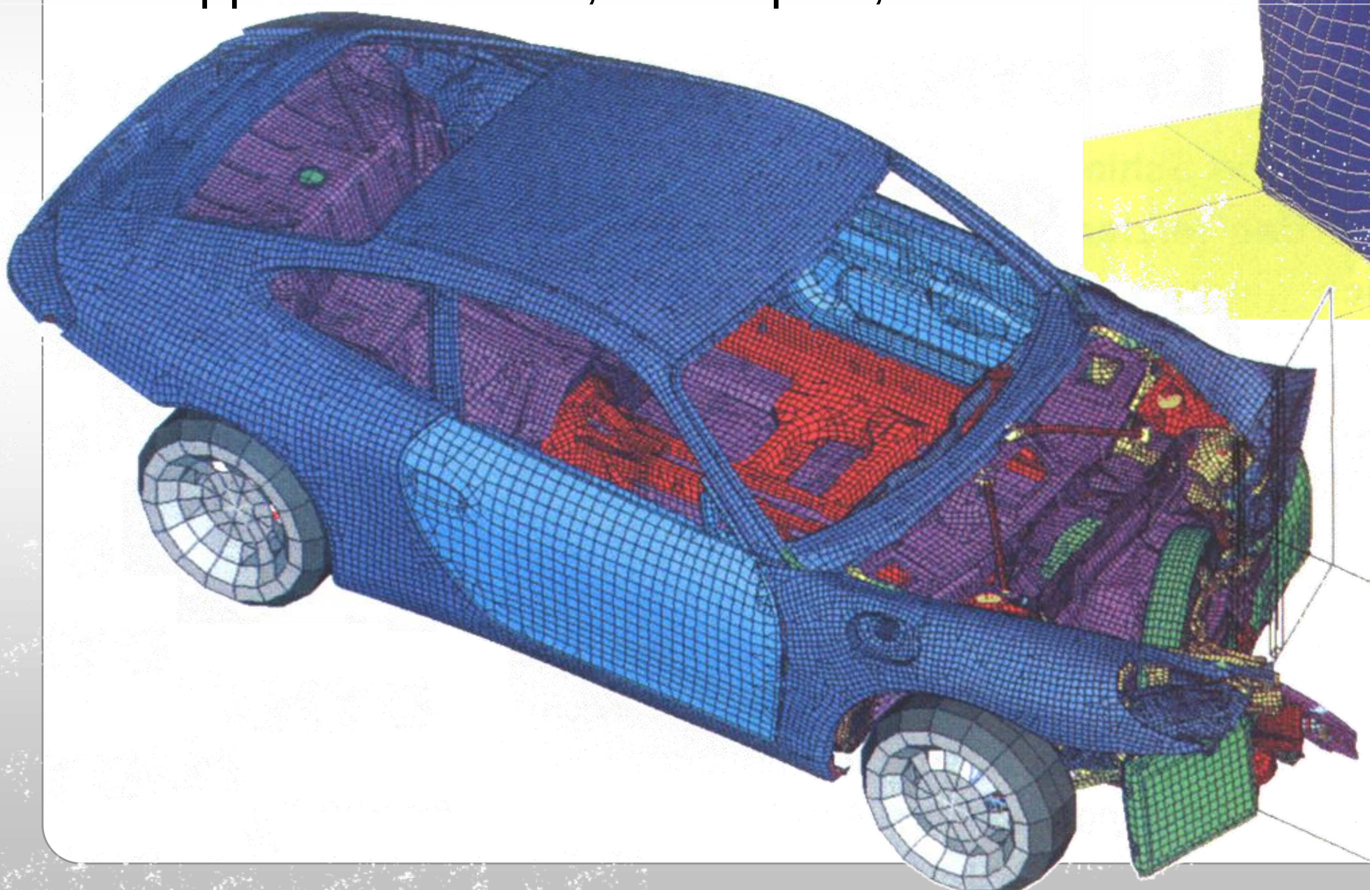


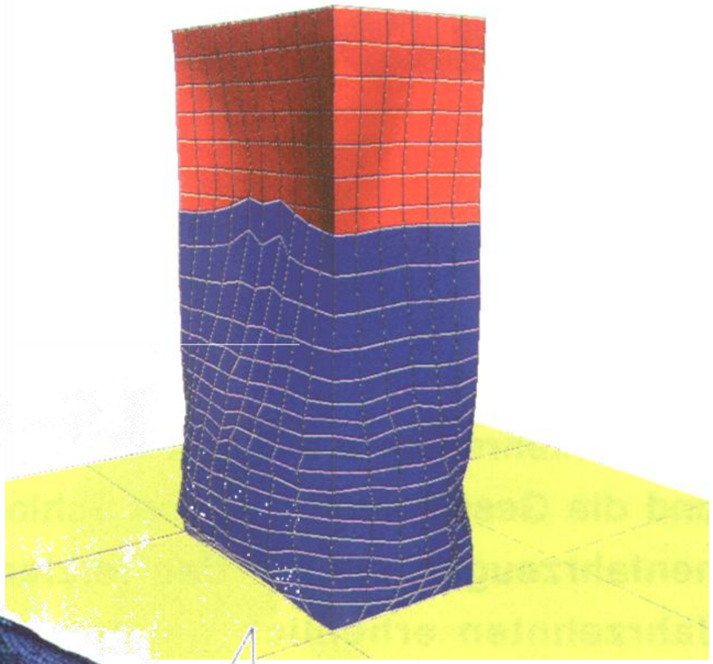
Initially: solid plate

Finally: Spokes

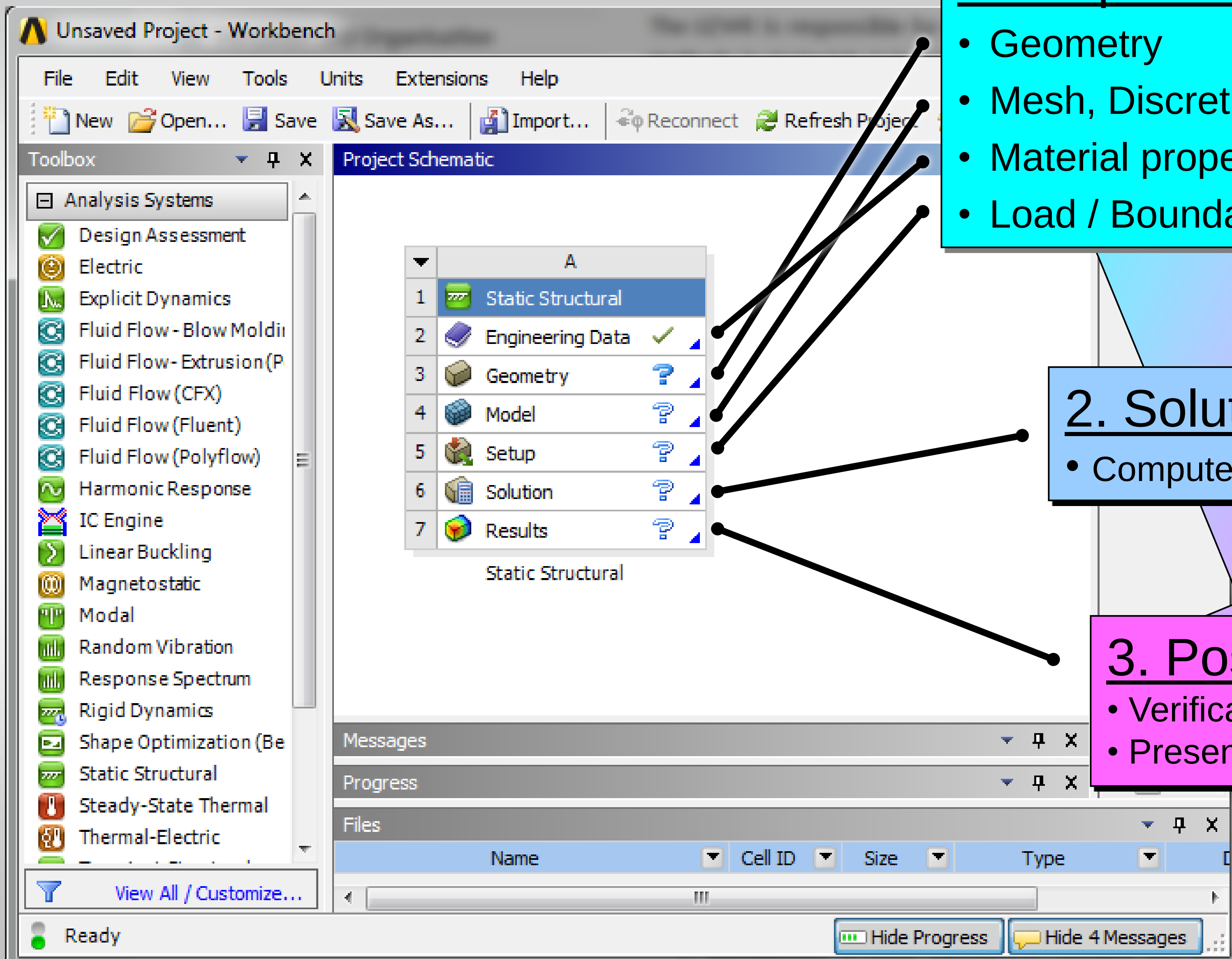
High Speed Dynamics

- An explicit FE solver is needed
- to solve initial value instead of boundary value problem
- Application: crash, fast impact, ...





Working Steps of a FEA



1. Preprocessor

- Geometry
- Mesh, Discretization
- Material properties
- Load / Boundary Conditions

2. Solution

- Computer is working

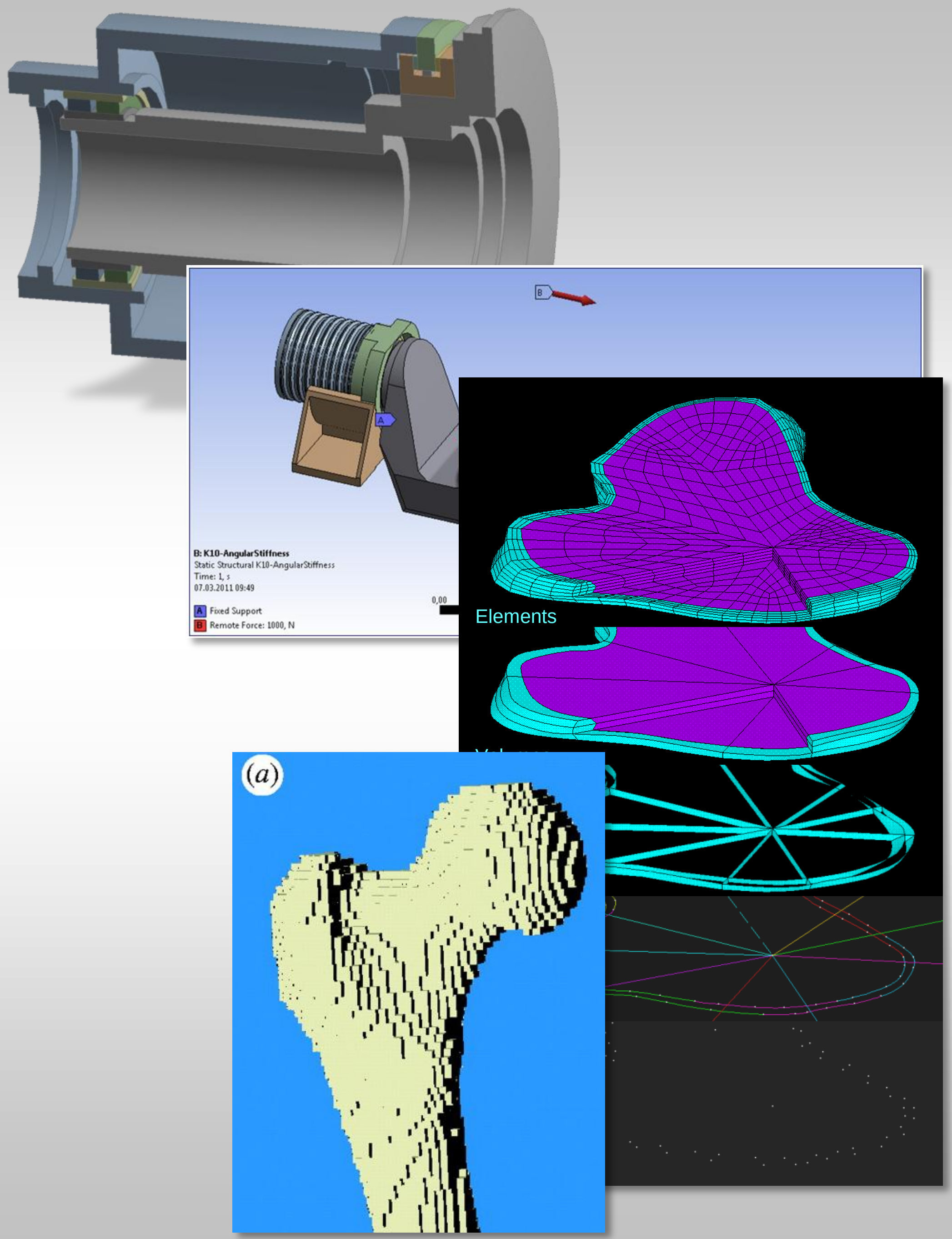
3. Postprocessor

- Verification,Validation
- Presenting results

Step 1: Preprocessor

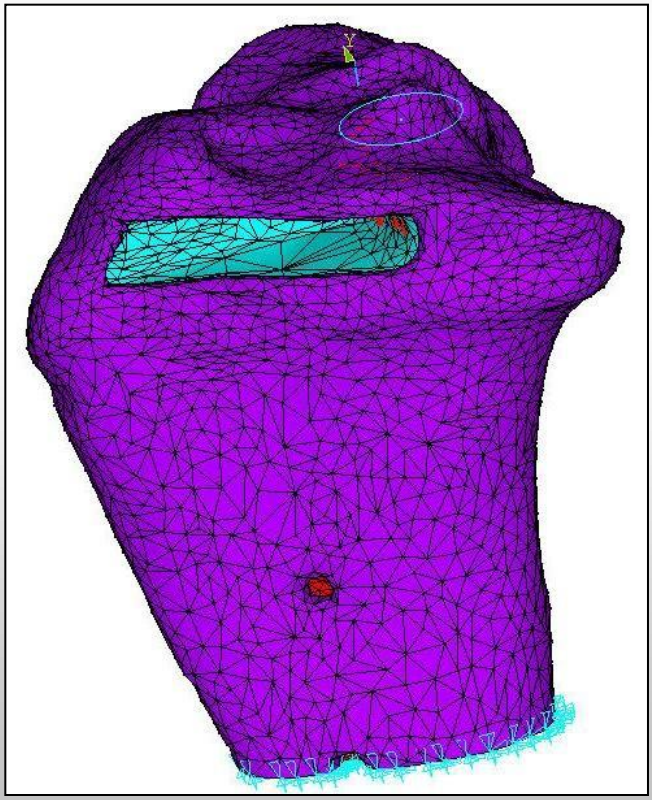
1.1 Geometry

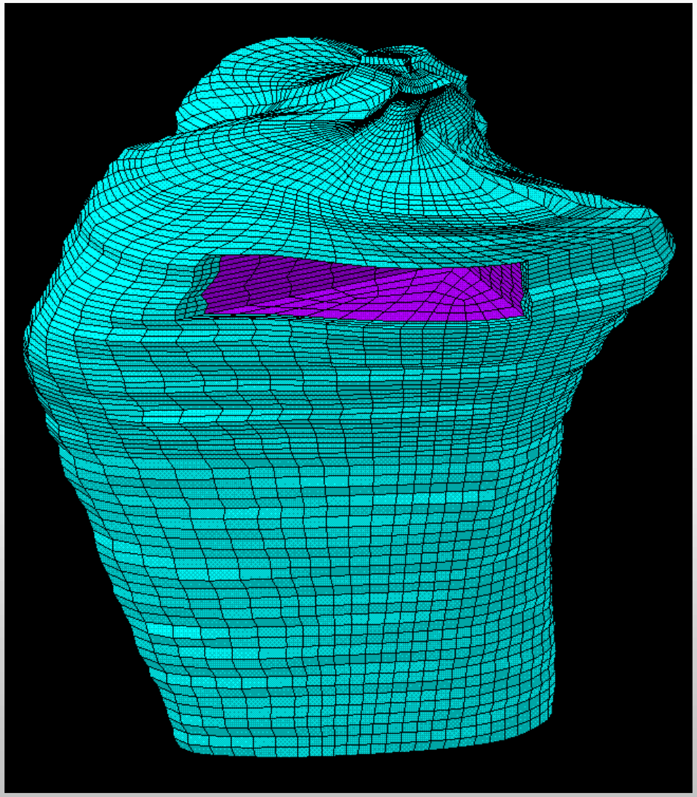
- Generate by CAD procedures
  - Primitives, 2D Sketches
  - Boolean operations
  - For simple Geometries
- Import Geometry from CAD
  - Primitives, 2D Sketches
  - Boolean operations
  - For simple Geometries
- Bottom-up Method
  - Points → Lines → Areas → Volumes
  - For simple geometries
  - Maximum control
  - Should be scripted (e.g. APDL)
- Direct generation of Elements:
  - E.g. Voxel Models out of CT data
  - Robust , automatic generation
  - For very complex geometries
  - No smooth surfaces → not good for contact



1.2 Meshing

- Tetraeheadrons: Better for complex geometry
- Hexaheadrons: Better mechanical properties
- Convergence: Better results with increasing number of elements.
  Finer mesh at higher solution gradients.
  Check dependency of your results on mesh size!
  (see Lab 2)





1.3 Material laws and properties

- Simplest: Linear elastic, isotropic: Young's modulus E and Poisson's ratio ν
- More complex: Non-linear, non-elastic (= plastic), hardening, fatigue, cracks
- Anisotropic: Transverse isotropic (wood), Orthotropic, ... → more than 2 parameters
- Biphasic: Porous media

1.4 Load and Boundary Conditions (BC)

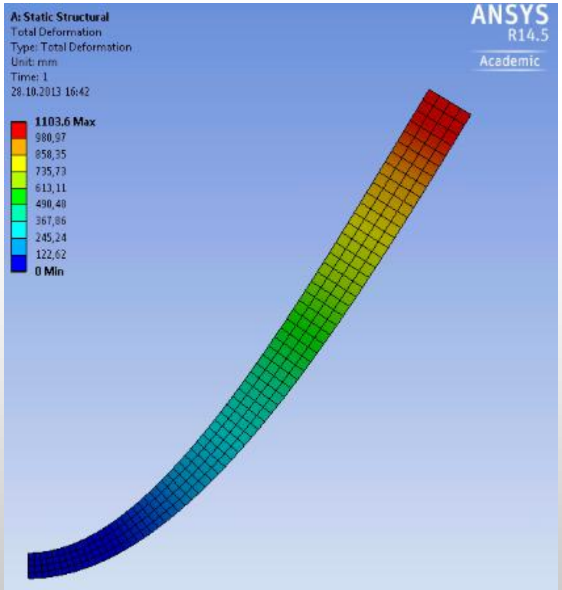
- Loads: Forces, pressures, displacements, accelerations, temperatures, …
- BC: fixations, supports, zero-displacements, symmetries, constraints, …
- Model should be supported by BC in order to prevent any rigid body movements!
- Ansys Workbench: Load & BC are applied to geometric items (areas, lines, …) and than transferred to the underlying nodes automatically.

Step 2: Solution

- The computer is doing the work
- Solver for linear systems: Direct solver or iterative solver
- Solver for non-linear systems: Iterativ, Newton-Raphson

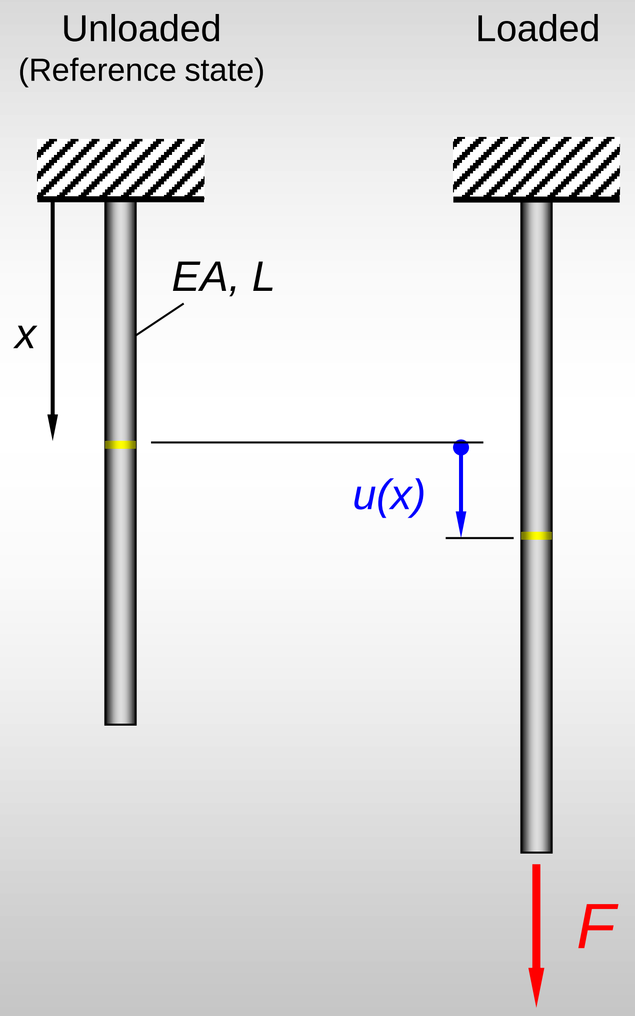
Step 3: Post-Processor

- Verification (check code, convergence, plausibility, ...)
- Validation (compare with experiments)
- Presenting the results (important message)
- Displacements (try always *true scale* and *high scale*)
- Strains, stresses
- Interpretation



Theory of the
Finite Element Method
using a ‘super simple’ example

Example: Tensile Rod



Given:

Rod with …

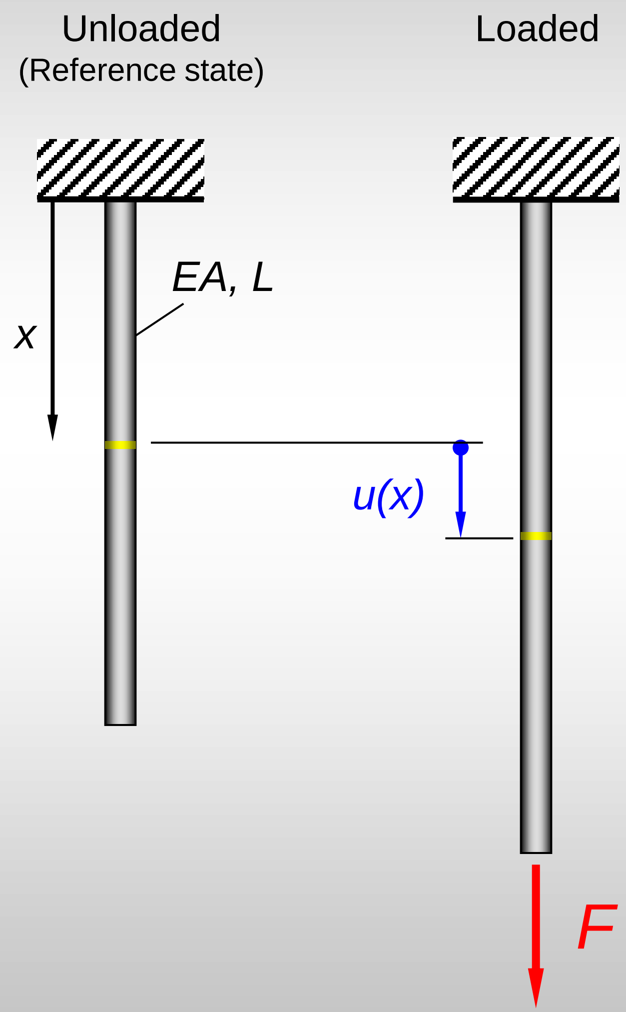
- Length L
- Cross-section A (constant)
- E-modulus E (constant)
- Force F (axial)
- Upper end fixed

To determine:

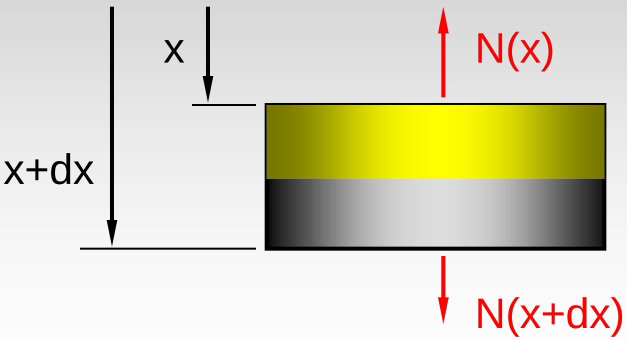
Deformation of the loaded rod:

Displacement function $u(x)$

A) Classical Solution (Method of „infinite“ Elements)



Differential Element
(infinitesimale heigth dx)



Generate the Differential Equation

1. Kinematics: $\varepsilon = u'$
2. Material: $\sigma = E\varepsilon \;\Rightarrow\; N = EAu'$
3. Equilibrium: $N' = 0$

$$\boxed{\Rightarrow \text{ODE:} \qquad (EA\,u')' = 0}$$

If EA = const then

$$\boxed{u'' = 0}$$

A) Classical Solution (Method of „infinite“ Elements)

Solve the Differential Equation

$$u''(x) = 0$$

Integrate 2 times: $u'(x) = C_1$

$u(x) = C_1 * x + C_2$ (General Solution)

Adjust to Boundary Conditions

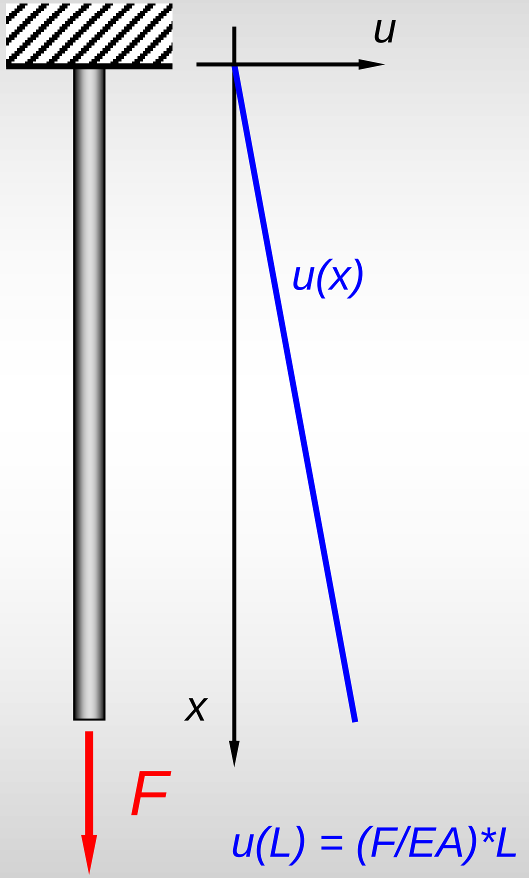
Top (fixation): $u(0) = 0 \Rightarrow C_2 = 0$

Bottom (open, force): $N(L) = F \Rightarrow u'(L) = F/(EA)$

$\Rightarrow C_1 = F/EA$

Adjusted Solution

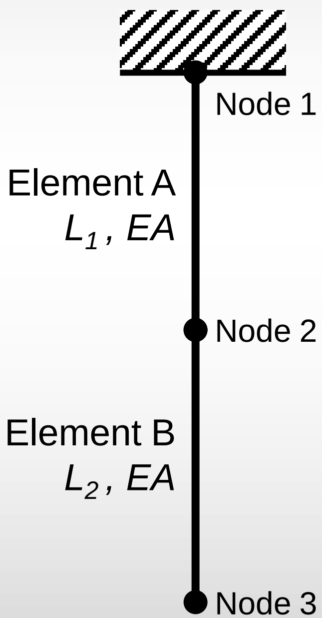
$$\mathbf{u(x) = (F/EA)*x}$$



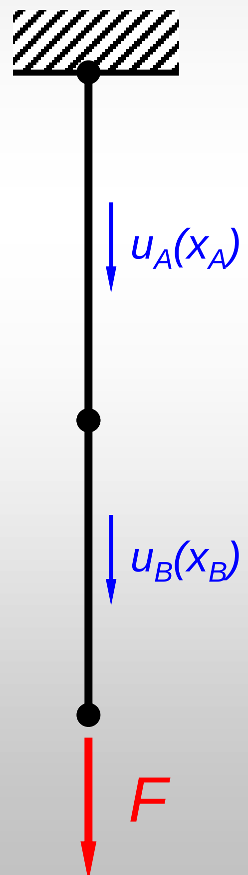
B) Solution with FEM

Discretization: We divide the rod into (only) two finite (= not infinitesimal small) **Elements.** The Elements are connected at their **nodes**.

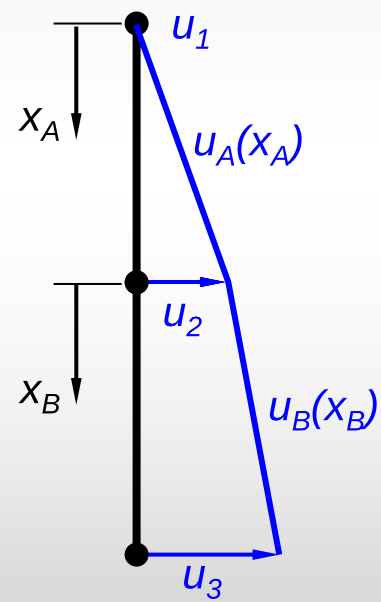
Unloaded:
(Reference condition)



Loaded:



Ansatz functions (linear) for the unknown displacements u



The unknown displacement function of the entire rod is described with a series of simple (linear) **ansatz functions** (see figure). This is the basic concept of FEM.

$$u_A(x_A) = \hat{u}_1 + (\hat{u}_2 - \hat{u}_1)\frac{x_A}{L_A} \quad = \hat{u}_1\left(1 - \frac{x_A}{L_A}\right) + \hat{u}_2\frac{x_A}{L_A}$$

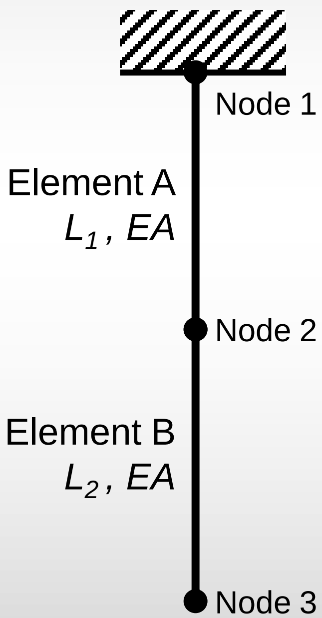
$$u_B(x_B) = \hat{u}_2 + (\hat{u}_3 - \hat{u}_2)\frac{x_B}{L_B} \quad = \hat{u}_2\left(1 - \frac{x_B}{L_B}\right) + \hat{u}_3\frac{x_B}{L_B}$$

The remaining unknowns are the three "nodal displacements" $\hat{u}_1, \hat{u}_2, \hat{u}_3$ and a no longer a whole function $u(x)$. Now we introduce the so-called "**virtual displacements (VD)". ** These are additional, virtual, small, arbitrarydisplacements $\delta\hat{u}_1, \delta\hat{u}_2, \delta\hat{u}_3$, consistent with BC. Basically: we "waggle" the nodes a bit.

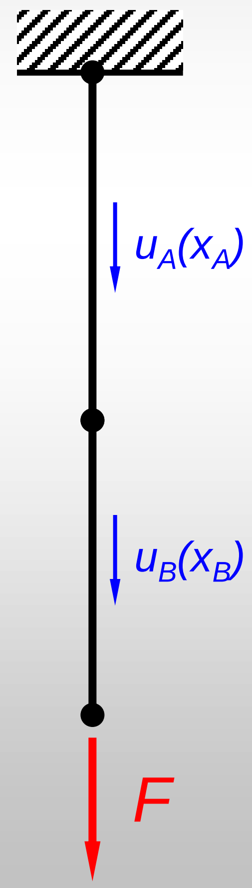
Now the **Principle of Virtual Displacements (PVD)** applies: A mechanical system is in equilibrium when the total work (i.e. elastic minus external work) due to the virtual displacements consequently disappears.

$$\delta W = 0 \quad \Rightarrow \quad \delta W_{el} - \delta W_a = 0$$

Unloaded:
(Reference condition)

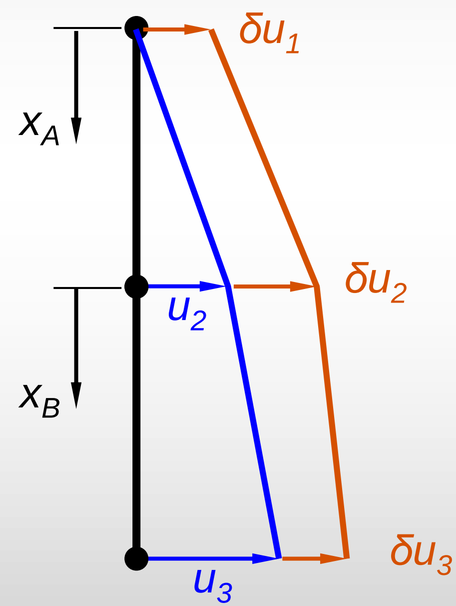


Loaded:



Ansatz functions (linear) for the unknown displacements *u*

Virtual displacements at all nodal displace-ments



For our simple example we can apply:

Virt. elastic work = normal force N times VD

Virt. external work = external force F times VD

The normal force N can be replaced by the expression EA/L times the element elongation. Element elongation again can be expressed by a difference of the nodal displacements:

$$\delta W = \int_{x_A=0}^{l_A} N_A(x_A)\cdot\delta\varepsilon(x_A)dx_A + \int_{x_B=0}^{l_B} N_B(x_B)\cdot\delta\varepsilon(x_B)dx_B - F\delta\hat{u}_3$$

$$\begin{aligned}\delta W &= N_A(\delta\hat{u}_2-\delta\hat{u}_1)+N_B(\delta\hat{u}_3-\delta\hat{u}_2)-F\delta\hat{u}_3\\ &= \frac{EA}{L_A}(\hat{u}_2-\hat{u}_1)(\delta\hat{u}_2-\delta\hat{u}_1)+\frac{EA}{L_B}(\hat{u}_3-\hat{u}_2)(\delta\hat{u}_3-\delta\hat{u}_2)-F\delta\hat{u}_3\end{aligned}$$

$$\begin{aligned}\delta W = {}& \delta\hat{u}_1\left(+\frac{EA}{L_A}\hat{u}_1-\frac{EA}{L_A}\hat{u}_2\right)\\ &+\delta\hat{u}_2\left(-\frac{EA}{L_A}\hat{u}_1+\frac{EA}{L_A}\hat{u}_2+\frac{EA}{L_B}\hat{u}_2-\frac{EA}{L_B}\hat{u}_3\right)\\ &+\delta\hat{u}_3\left(-\frac{EA}{L_B}\hat{u}_2+\frac{EA}{L_B}\hat{u}_3-F\right)=0\end{aligned}$$

With this principle we unfortunately have only **<u>one</u> equation for the <u>three</u> unknown** displacements $\hat{u}_1$, $\hat{u}_2$, $\hat{u}_3$. What a shame! However, there is a trick…

Abbreviated we write:

$$\delta\hat{u}_1(...)_1 + \delta\hat{u}_2(...)_2 + \delta\hat{u}_3(...)_3 = 0$$

The **virtual displacements can be chosen independently** of one another. For instance all except one can be zero. Then the term within the bracket next to this not zero VD has to be zero, in order to fulfill the equation. However, as we can chose the VD we want and also another VD could be chosen as the only non-zero value, consequently all three brackets must individually be zero. **We get three equations**. Juhu!

$$(...)_1 = 0; \quad (...)_2 = 0; \quad (...)_3 = 0$$

… which we can also write down in matrix form:

$$\begin{bmatrix} \frac{EA}{L_1} & -\frac{EA}{L_1} & 0 \\ -\frac{EA}{L_1} & \frac{EA}{L_1} + \frac{EA}{L_2} & -\frac{EA}{L_2} \\ 0 & -\frac{EA}{L_2} & \frac{EA}{L_2} \end{bmatrix} \begin{bmatrix} \hat{u}_1 \\ \hat{u}_2 \\ \hat{u}_3 \end{bmatrix} = \begin{bmatrix} 0 \\ 0 \\ F \end{bmatrix}$$

$$\begin{bmatrix} \frac{EA}{L_1} & -\frac{EA}{L_1} & 0 \\ -\frac{EA}{L_1} & \frac{EA}{L_1}+\frac{EA}{L_2} & -\frac{EA}{L_2} \\ 0 & -\frac{EA}{L_2} & \frac{EA}{L_2} \end{bmatrix} \begin{bmatrix} \hat{u}_1 \\ \hat{u}_2 \\ \hat{u}_3 \end{bmatrix} = \begin{bmatrix} 0 \\ 0 \\ F \end{bmatrix}$$

Or in short:

$$\underline{\underline{K}}\,\underline{\hat{u}} = \underline{\underline{F}}$$

$\underline{\underline{K}}$ - Stiffness matrix

$\underline{\hat{u}}$ - Vector of the unknown nodal displacement

$\underline{F}$ - Vector of the nodal forces

This is the classical fundamental equation of a structural mechanics, linear FE-analysis. A **linear system of equations** for the unknown nodal displacements

We still have to account for the **boundary conditions**. The rod is fixed at the top end. As a consequence node 1 cannot be displaced:

$$\hat{u}_1 = 0$$

Because the virtual displacements also have to fulfill the boundary conditions we have $\delta\hat{u}_1 = 0$. Therefore we need to eliminate the first line in the system of equations, as this equation does no longer need to be fulfilled. The first column of the matrix can also be removed, as these elements are in any case multiplied by zero. So it becomes …

$$\begin{bmatrix} \frac{EA}{L_1}+\frac{EA}{L_2} & -\frac{EA}{L_2} \\ -\frac{EA}{L_2} & \frac{EA}{L_2} \end{bmatrix} \begin{bmatrix} \hat{u}_2 \\ \hat{u}_3 \end{bmatrix} = \begin{bmatrix} 0 \\ F \end{bmatrix}$$

$u(x) = (F/EA)*x$

We **solve** the system of equations and obtain the nodal displacements

$$\hat{u}_2 = \frac{L_A}{EA} F \quad \text{und} \quad \hat{u}_3 = \frac{L_A + L_B}{EA} F$$

Here the **FE-solution** corresponds exactly with the (existing) analytical solution. In a more complex example this would not be the case.
Generally, it applies that the convergence of the numerical solution with the exact solution continually improves with an increasing number of finite elements. For extremely complicated problems there is no longer an analytical solution; for such cases one needs FEM!

From the nodal displacements one can also determine **strains and stresses** in a subsequent calculation. In our example strains and stresses stay constant within the elements.

$$\varepsilon_A(x_A) = \frac{\hat{u}_2 - \hat{u}_1}{L_A}$$

$$\varepsilon_B(x_B) = \frac{\hat{u}_3 - \hat{u}_2}{L_B}$$

Strains

$$\sigma_A(x_A) = E\varepsilon_A(x_A)$$

$$\sigma_B(x_B) = E\varepsilon_B(x_B)$$

Stresses

Finished!

Summary

The **essential steps** and ideas of FEM are thus:

- Discretization: Division of the spatial domain into finite elements
- Choose simple Ansatz functions (polynomials) for the unknown variables within the elements. This reduces the problem to a finite number of unknowns.
- Write up a mechanical principle (e.g. PVD, the mathematician says "weak formulation" of the PDE) and
- From this derive a system of equations for the unknown nodal variables
- Solve the system of equations

Many of these steps will no longer be apparent when using a commercial FE program. With the selection of an analysis and an element type the underlying PDE and the Ansatz functions are implicitly already chosen. The mechanical principle was only being used during the development of the program code in order to determine the template structure of the stiffness matrix. During the solution run the program first creates the(big) linear system of equations based on that known template structure and than solves the system in terms of nodal displacements.

General Hints and Warnings

- FEA is a tool, not an solution
- Take care about nice pictures („GiGo“)
- Parameter
- Verification

needs experiments

- FE models are case (question) specific

Literature and Links reg. FEM

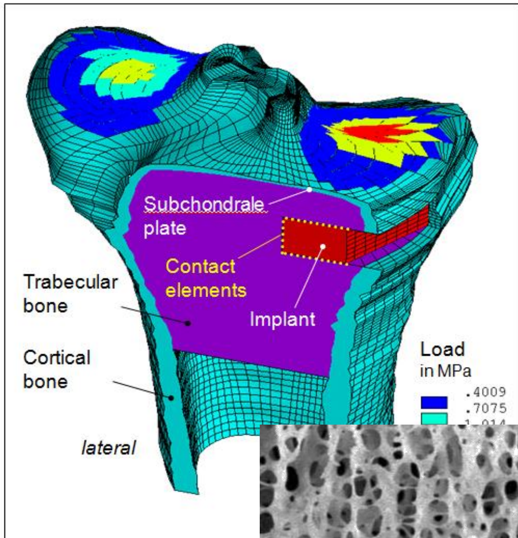
Books:

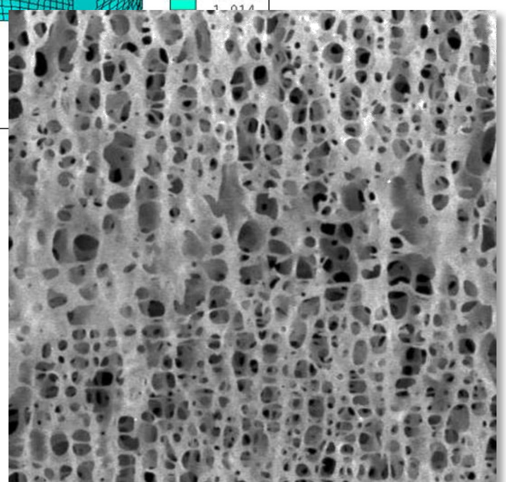
- <u>Zienkiewicz, O.C.</u>: *„Methode der finiten Elemente"; Hanser 1975 (engl. 2000).*
  The bible of FEM (German and English)
- <u>Bathe, K.-J.</u>: *„Finite-Elemente-Methoden"; erw.* 2. Aufl.; *Springer 2001*
  Textbook (theory)
- <u>Dankert, H. and Dankert, J.</u>: „*Technische Mechanik*"; Statik, Festigkeitslehre, Kinematik/Kinetik, mit Programmen; 2. Aufl.; Teubner, 1995.
  German mechanics textbook incl. FEM, with nice homepage
  http://www.dankertdankert.de/
- <u>Müller, G. and Groth, C.</u>: *„FEM für Praktiker, Band 1: Grundlagen"*, mit ANSYS/ED-Testversion (CD). (Band 2: Strukturdynamik; Band 3: Temperaturfelder)
  ANSYS Intro with examples (German)
- <u>Smith, I.M. and Griffiths, D.V.</u>: *„Programming the Finite Element Method"*
  From engineering introduction down to programming details (English)
- <u>Young, W.C. and Budynas, G.B</u>: „Roark's Formulas for Stress and Strain "
  Solutions for many simplified cases of structural mechanics (English)

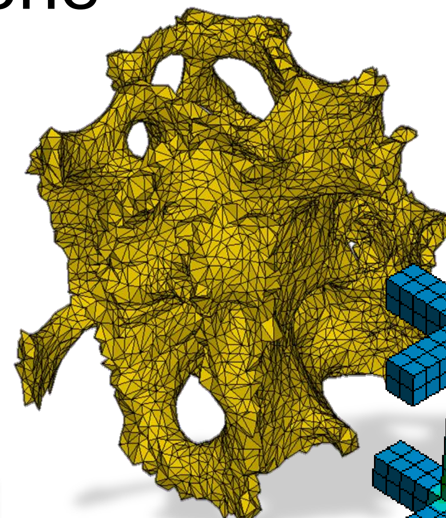
Links:

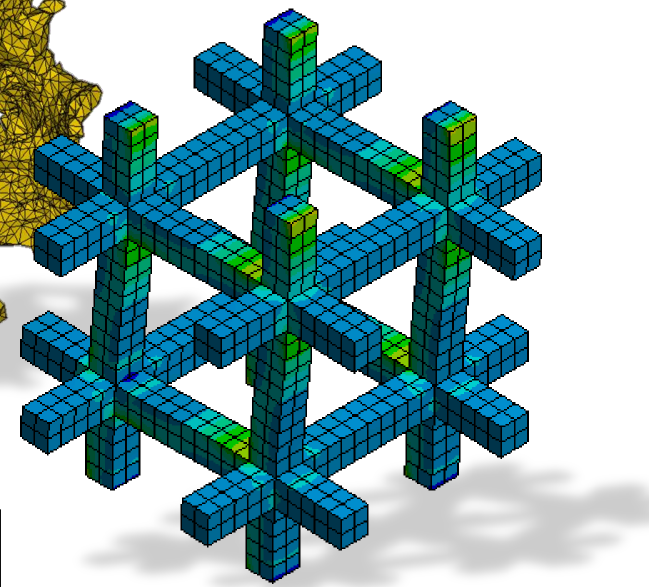
- <u>Z88 Free FE-Software:</u> http://z88.uni-bayreuth.de/

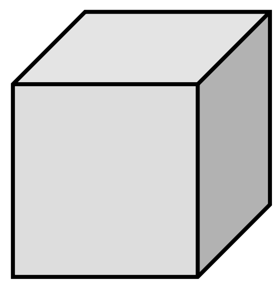
Modeling Trabecular Bone











Carter&Hayes, 1977:

$$E = 3790\,\dot{\epsilon}^{0.06}\,\rho^3.$$

$$S = 68\,\dot{\epsilon}^{0.06}\,\rho^2$$

https://www.researchgate.net/publication/22580090_The_Compressive_Behavior_of_Bone_as_Two_Phase_Porous_Structure