Parallelisation and domain decomposition in FEM

dr. Borut Černe
University of Ljubljana, Faculty of Mechanical Engineering

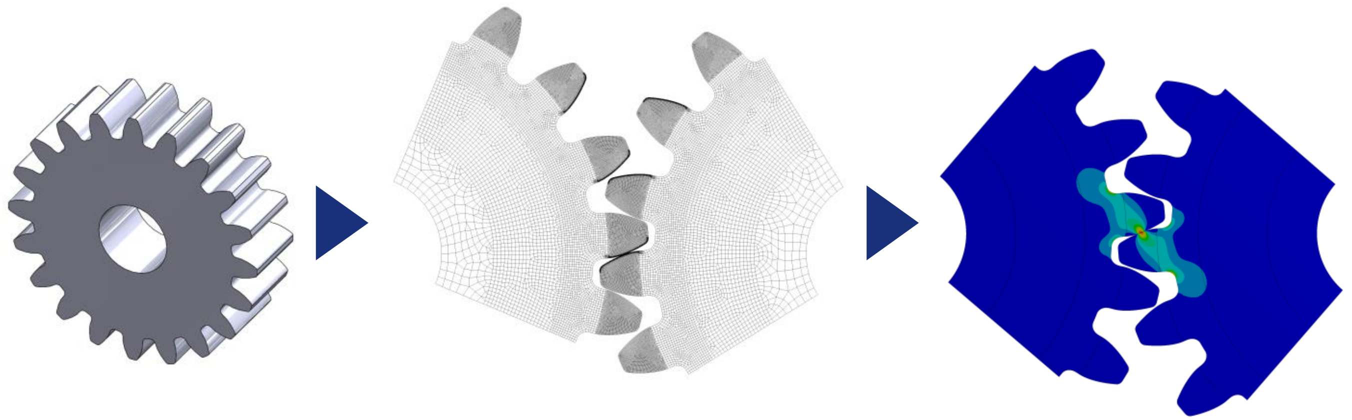
Typical FEM analysis workflow

1. **Preliminary problem dissection:**
   - Analysis type: Static, thermal, modal…
   - Constitutive/physical model
   - Geometry: single body/multibody
   - Geom. space: 2D/3D
   - Element types: Line/surface/solid elements
   - Symmetry conditions, etc.

2. **Preprocessing:**
   - Import/create geometry
   - Mesh geometry
   - Material properties
   - Loads and boundary conditions (BCs)

3. **Solving the model**

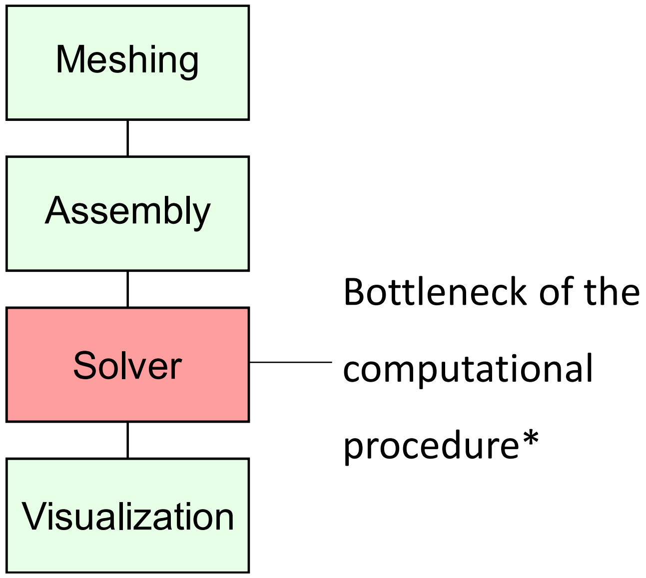
4. **Postprocessing:**
   - Review results
   - Data analysis
   - Verify/validate solution



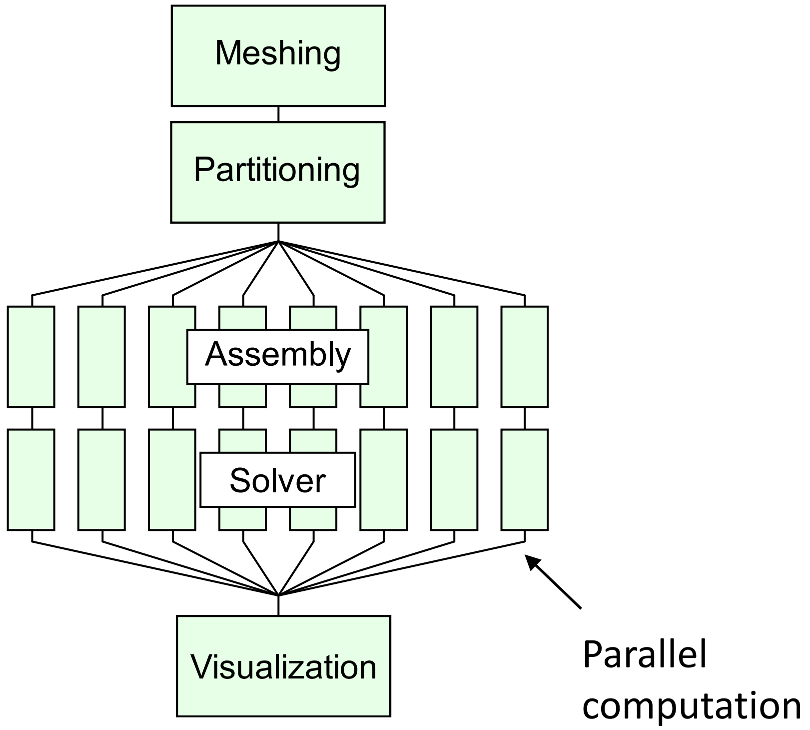
Parallelisation in FEM

- Computational steps in typical FEM analysis:

1. **Serial computation:**



2. **Parallelized computation**



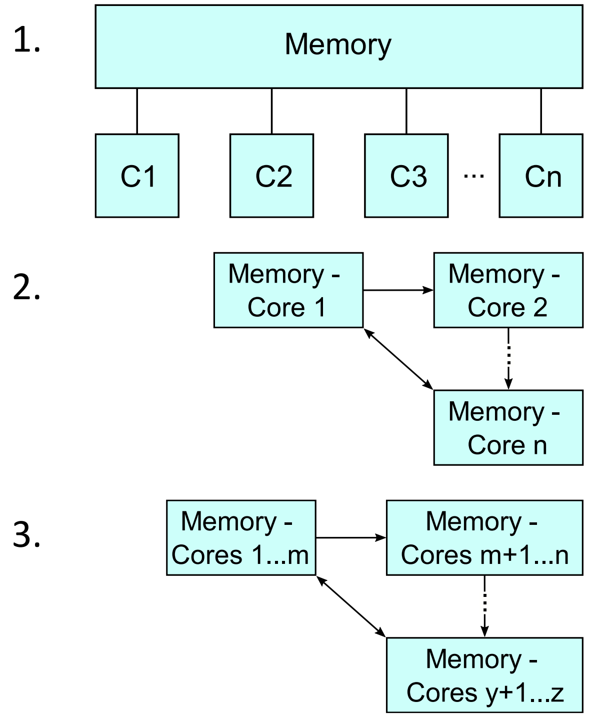
➢ * For large cases the other steps can also be highly time consuming – parallelization there also required

Parallelisation in FEM

- Data dependency – does the computation of one task require data from other tasks to proceed?
  - FEM is inherently data dependent – reflection of the physical reality of the problem

▸ Parallel computers:

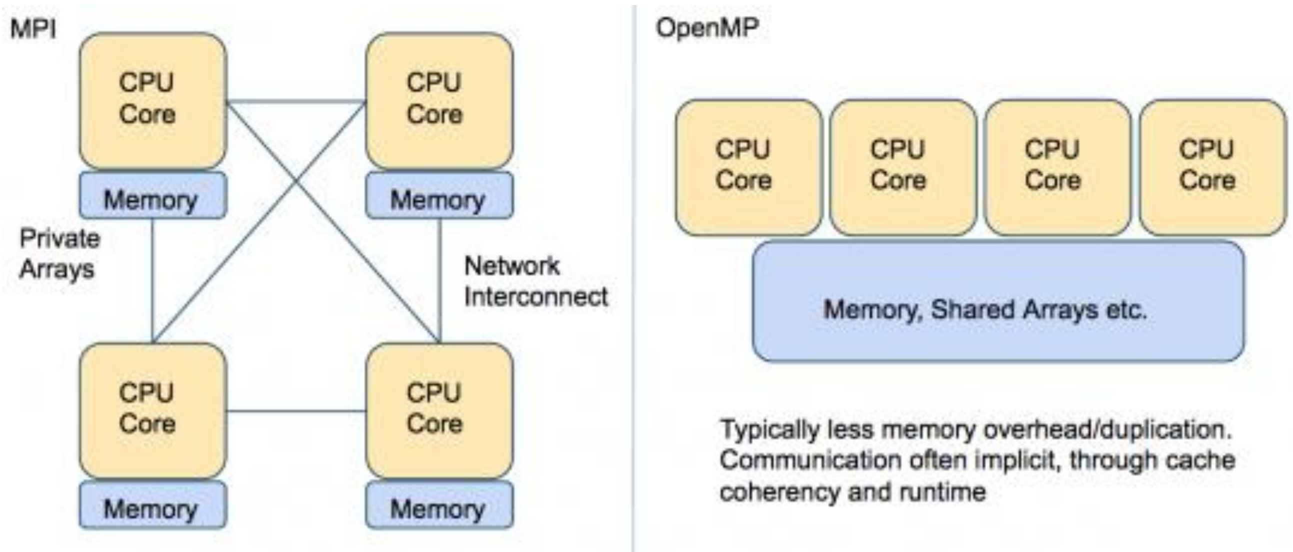
1. Shared memory – all cores access the whole memory
2. Distributed memory:
   - Each core has its own memory unit
   - Communication protocol for memory access between different cores
3. Typical HPC combines distributed and shared memory capabilities



Parallelisation in FEM

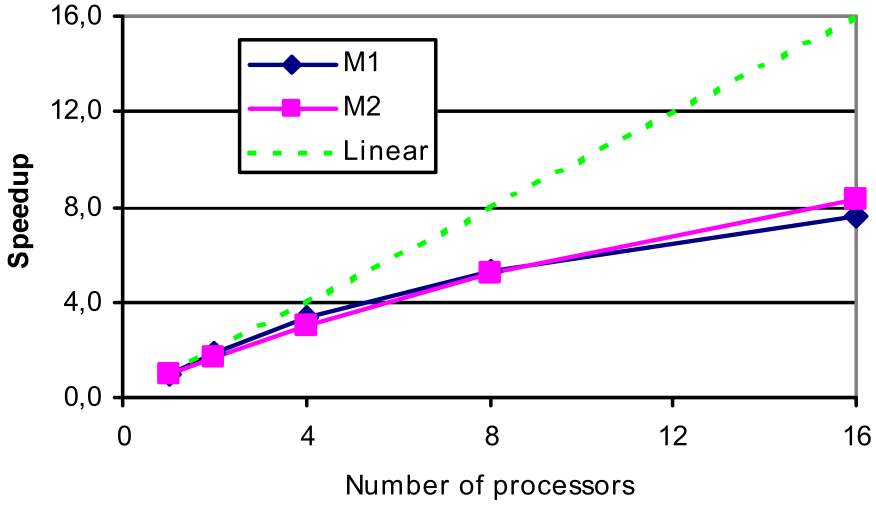
- **Message Passing Interface (MPI)**
  - ➢ Primarily used for distributed memory computing
  - ➢ Can be used both in distributed and shared memory computers
  - ➢ Programming model allows good parallel scalability
  - ➢ Programming is quite explicit

- **Threads (pthreads, OpenMP)**
  - ➢ Can be used only in shared memory computer
  - ➢ Limited parallel scalability
  - ➢ Simpler - less explicit programming



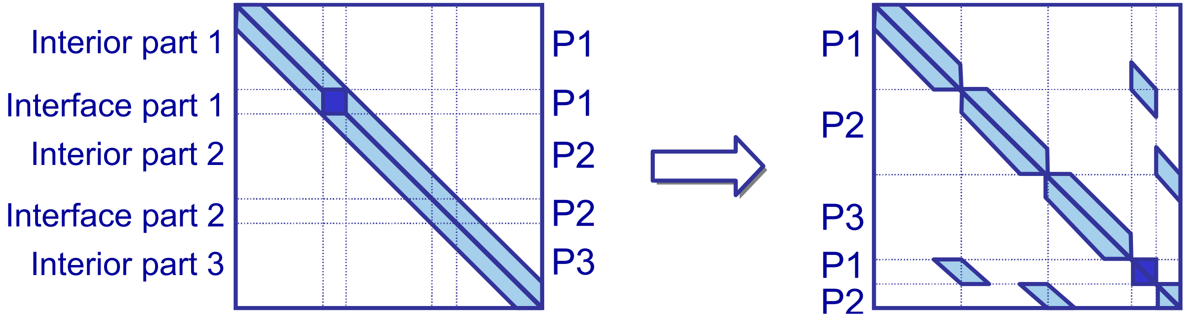
Parallelisation in FEM

- Scalability of parallelized FEM computations:
  - *Strong scaling* – how the solution time decreases with an increased number of processors for a fixed total problem size
    - ➢ Best case scenario: $N_p \cdot T$ = const.
  - *Weak scaling* – how the solution time varies with the number of processors for a fixed problem size per processor
    - ➢ Best case scenario: T = const.
  - Typically >10^4 FEs needed for suitable scaling



FEM Solvers

- **Iterative methods:**
  - Parallelized Conjugate Gradient (CG) with
    - Incomplete Cholesky *preconditioner* (ICCG) – block (BICCG) or renumbering process (PICCG-RP) methods
    - Diagonal preconditioning (DPCG) method
  - BI-Conjugate Gradient Stabilized method (BICGSTAB)
  - Quasi-Minimal Residual (QMR) method
  - QCR, GMRes, TFQMR,…



Example: PICCG-RP algorithm

- **Direct methods:**
  - Sparse LU decomposition solvers
  - Suitable for ill-conditioned cases (very stiff bodies, large displacements, etc.)
  - MUMPS parallel sparse solver

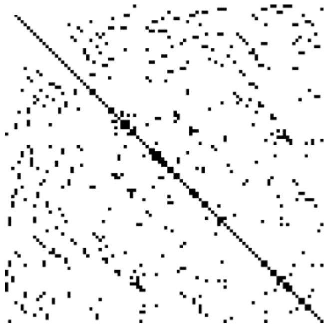
Sparse FE system:

$$\mathbf{K}\mathbf{u} = \mathbf{f}$$

$\rightarrow$ **K** - Stiffness matrix

$\rightarrow$ **u** - Displacement vector

$\rightarrow$ **f** - External loads vector

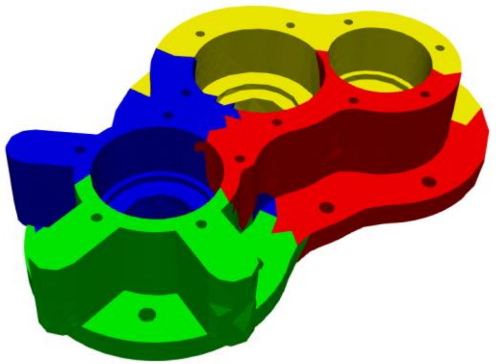


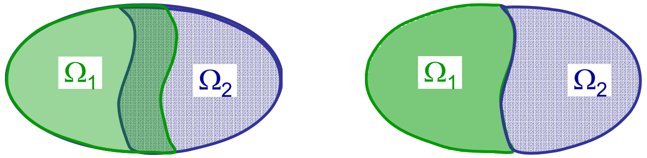
Black dots = non-zero values

FEM Domani decomposition

- **Domain decomposition (I):**
  - Decomposition of the spatial domain into subdomains
  - Iterative algorithms preferred to direct solvers (better efficiency)
  - Overlapping or non-overlapping decomposition methods

1. *Overlapping decomposition method:*
   - Schwarz iterative methods
   - Approximation of BCs on interface
   - On each subdomain – iterative or direct solver can be used

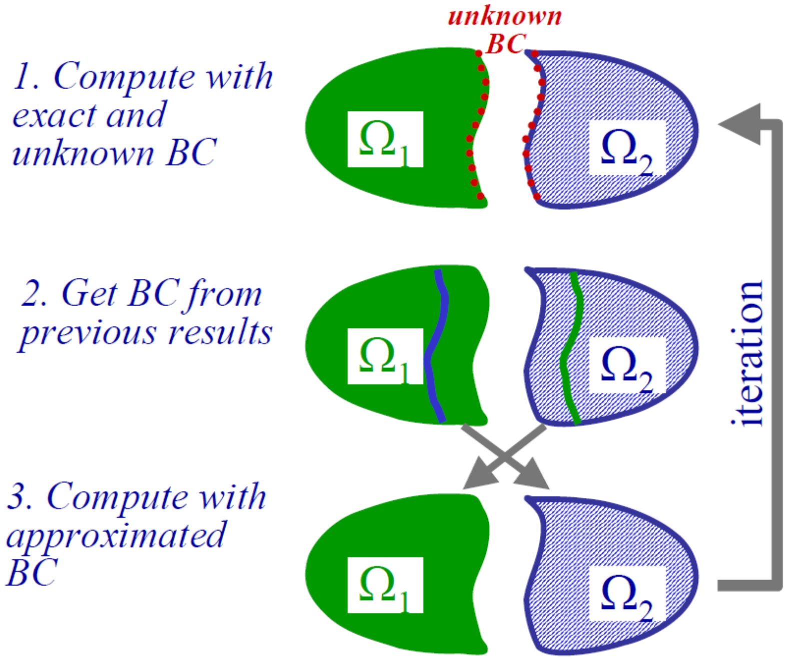




Overlapping

Non-overlapping

Overlapping decomposition:



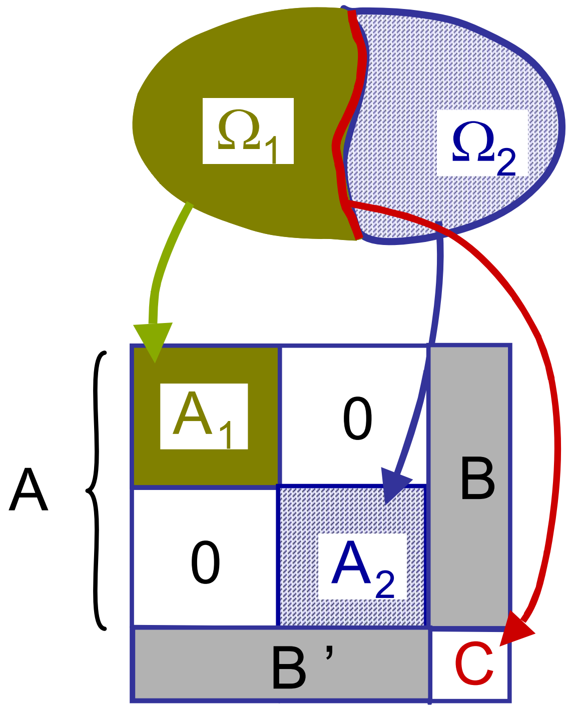
Parallelisation in FEM

- **Domain decomposition (II):**
  - *Non-overlapping decomposition method:*
    - Schur complement system method

$$\begin{bmatrix} A & B \\ B' & C \end{bmatrix} \begin{bmatrix} x \\ y \end{bmatrix} = \begin{bmatrix} b_1 \\ b_2 \end{bmatrix} \Rightarrow \begin{cases} x = A^{-1}b_1 - A^{-1}By & \rightarrow \text{Base system} \\ (C - B'A^{-1}B)y = b_2 - B'A^{-1}b_1 & \rightarrow \text{Schur comp. s.} \end{cases}$$

    - Continuity of BCs on subdomain interfaces can also be obtained using Lagrange multipliers
      - Similar approach used in contact mechanics
      - FE Tearing and Interconnecting (FETI) method uses such approach

Non-overlapping decomposition



Schur complement system method

How to choose FEM solver

Shared Memory Solver Guidelines

Solver	Typical Applications	Ideal Model Size	Memory Use	Disk (I/O) Use
Sparse Direct Solver (direct elimination)	When robustness and solution speed are required (nonlinear analysis); for linear analysis where iterative solvers are slow to converge (especially for ill-conditioned matrices, such as poorly shaped elements).	100,000 DOF (and beyond)	Out-of-core: 1 GB/MDOF In-core: 10 GB/MDOF	Out-of-core: 10 GB/MDOF In-core: 1 GB/MDOF
PCG Solver (iterative solver)	Reduces disk I/O requirement relative to sparse solver. Best for large models with solid elements and fine meshes. Most robust iterative solver in ANSYS.	500,000 DOF to 20 MDOF+	0.3 GB/MDOF w/MSAVE,ON; 1 GB/MDOF without MSAVE	0.5 GB/MDOF
JCG Solver (iterative solver)	Best for single field problems - (thermal, magnetics, acoustics, and multiphysics). Uses a fast but simple preconditioner with minimal memory requirement. Not as robust as PCG solver.	500,000 DOF to 20 MDOF+	0.5 GB/MDOF	0.5 GB/MDOF
ICCG Solver (iterative solver)	More sophisticated preconditioner than JCG. Best for more difficult problems where JCG fails, such as unsymmetric thermal analyses.	50,000 to 1,000,000+ DOF	1.5 GB/MDOF	0.5 GB/MDOF
QMR Solver (iterative solver)	Used for full harmonic analyses. This solver is appropriate for symmetric, complex, definite, and indefinite matrices.	50,000 to 1,000,000+ DOF	1.5 GB/MDOF	0.5 GB/MDOF

For more info: ANSYS Help: Mechanical APDL > 5.1. Selecting a Solver

How to choose FEM solver

Distributed Memory Solver Selection Guidelines

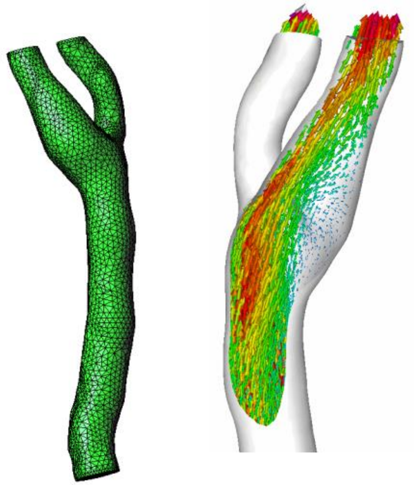
Solver	Typical Applications	Ideal Model Size	Memory Use	Disk (I/O) Use
Distributed Memory Sparse Direct Solver	Same as sparse solver but can also be run on distributed memory parallel hardware systems.	500,000 DOF to 10 MDOF (works well outside this range)	Out-of-core: 1.5 GB/MDOF on master machine, 1.0 GB/MDOF on slave machines	Out-of-core: 10 GB/MDOF
			In-core: 15 GB/MDOF on master machine, 10 GB/MDOF on slave machines	In-core: 1 GB/MDOF
Distributed Memory PCG Solver	Same as PCG solver but can also be run on distributed memory parallel hardware systems.	1 MDOF to 100 MDOF	1.5-2.0 GB/MDOF in total*	0.5 GB/MDOF
Distributed Memory JCG Solver	Same as JCG solver but can also be run on distributed memory parallel hardware systems. Not as robust as the distributed memory PCG or shared memory PCG solver.	1 MDOF to 100 MDOF	0.5 GB/MDOF in total*	0.5 GB/MDOF

For more info: ANSYS Help: Mechanical APDL > 5.1. Selecting a Solver

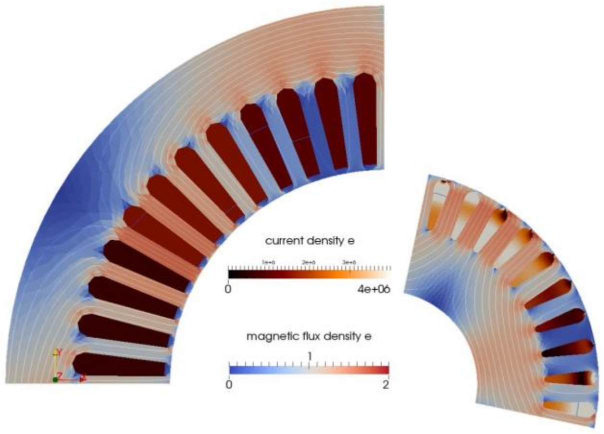
Linear v. nonlinear FEM problems

Physical models in FEM

- Heat transfer
  - Steady state heat equation
  - Transient heat equation
- Solid mechanics
  - Linear elasticity
  - Finite elasticity
  - Shell equations
- Fluid mechanics and transport phenomena
  - Navier-Stokes equation
  - Advection-diffusion equation
  - Reynolds equation – thin film flow
- Acoustics
  - Helmholtz model
  - Linearized Navier-Stokes in the frequency domain
- Electromagnetism
  - Electrostatics
  - Circuits and dynamics solver
  - Magnetic induction equation
- Other
- User defined elements and models



Raback, PRACE 2014

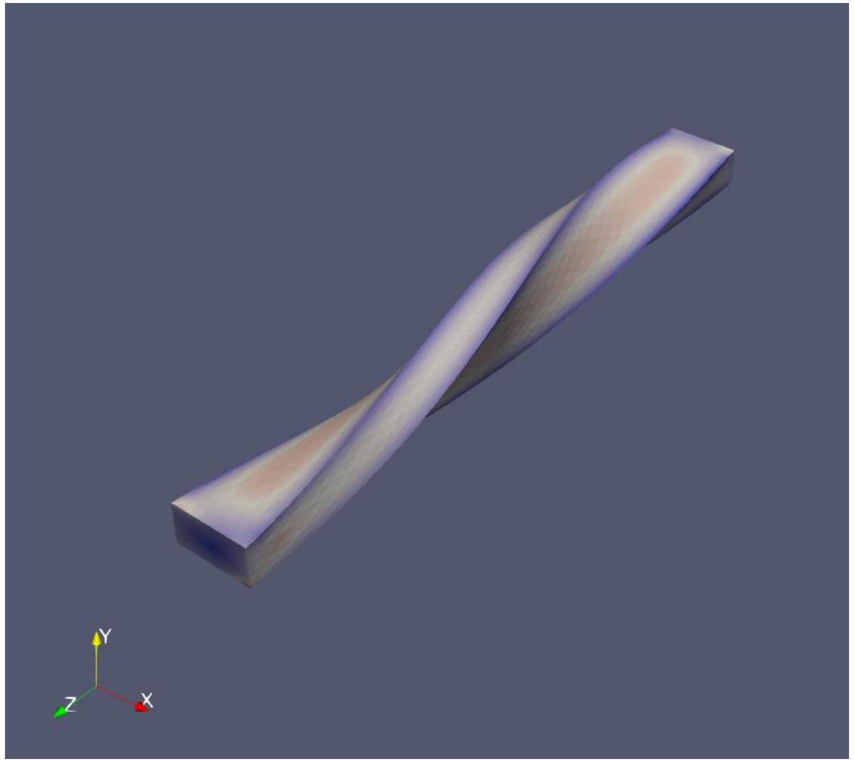


Ponomarev, SEMTEC 2017

Linear v. nonlinear problems in FEM

Nonlinear model examples

- Heat transfer
  - Transient heat equation with **heat radiation effect**
- Solid mechanics
  - **Geometric nonlinearities** (e.g. finite elasticity)
  - **Mechanical nonlinearities** (e.g. hyperelasticity, viscoplasticity, nonlinear viscoelasticity)
  - **Contact nolinearities** (e.g. sliding frictional contacts)
- Fluid mechanics and transport phenomena
  - **Navier-Stokes** equation with **inertial fluid force** or **convective effect**
- Acoustics
  - Large-amplitude **wave propagation**
- Electromagnetism
  - **Poisson-Boltzmann** equation – steady state electric potential law



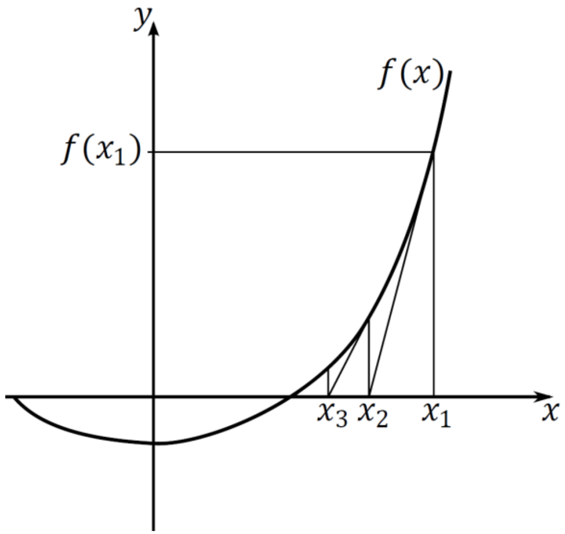
Linear v. nonlinear problems in FEM

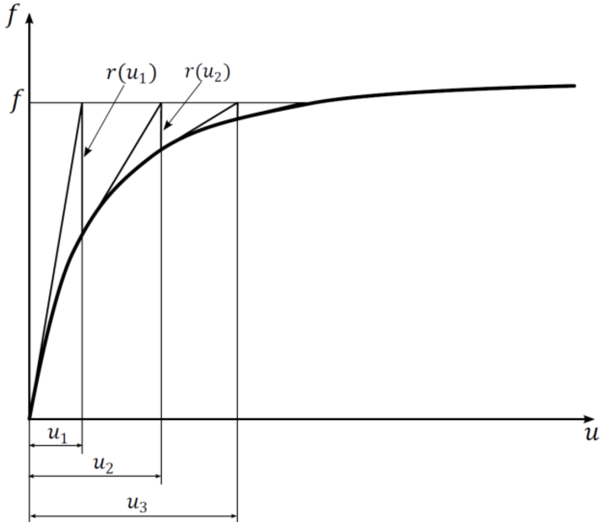
How FEM solvers deals with nonlinear problems?

- Solver always iterative
- Problem always first linearized to the form

$$A(u_{i-1})u_i = b(u_{i-1})$$

- Two iterative algorithms available: **Newton** (aka Newton-Raphson) and **Picard** schemes
- Choice of interative solution scheme depends on used model, e.g.:
  - Solid mechanics – Newton method
  - Navier-Stokes – Picard + Netwon method combination





Newton-Raphson method applied to a nonlinear solid material model

Cerne 2014 – Master thesis

See also: Wikiversity – Nonlinear finite elements

Linear v. nonlinear problems in FEM

Time-dependent physical problems

- Can be linear or nonlinear
- General first order differential eq. (DE) problem form:

$$M\frac{\partial \Phi}{\partial t} + K\Phi = F$$

  - Example – viscoelastic solid models

- Time-dependent term can be solved numerically using various schemes:
  1. Backward Different Formula (BDF)

$$\left(\frac{1}{\Delta t}M + K\right)\Phi^{i+1} = F^{i+1} + \frac{1}{\Delta t}M\Phi^{i}$$

  - First order implicit solution

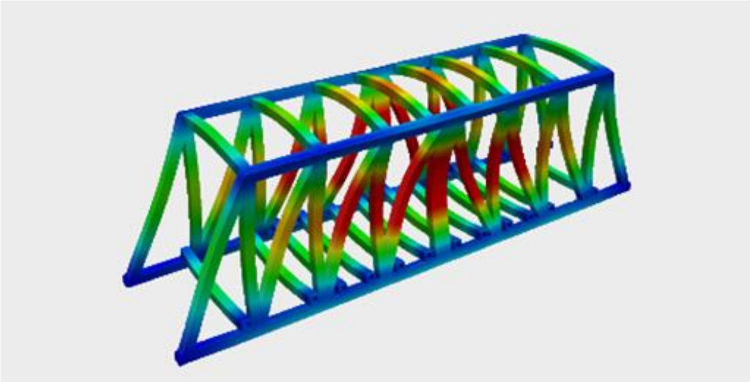
  2. Crank-Nicholson method

- Second order time derivatives:

$$M\frac{\partial^2 \Phi}{\partial t^2} + B\frac{\partial \Phi}{\partial t} + K\Phi = F$$

  - Example – structural dynamics

- the Bossak-Newmark method can be used to solve the DE numerically



Thank you for your attention!

dr. Borut Černe

University of Ljubljana, Faculty of Mechanical Engineering