Numerical Methods in Biomechanics

ME-484

Alexandre Terrier

EPFL - Laboratory of Biomechanical Orthopedics

Content - schedule

W01: Organization, introduction and examples
W02: External lecturers
W03: Partial Differential Equations
W04: Solid mechanics in numerical biomechanics
W05: Fluid mechanics in numerical biomechanics
W06: Midterm project presentations
W07: The Finite Element Method (FEM), and its extensions
W08: Midterm evaluation
W09: Multiphysics, coupling, Comsol
W10: Multiphysics example 1
W11: Multiphysics example 2
W12: Multiphysics example 3
W13: Final project presentation

Numerical methods for PDE

Most PDEs can’t be solved algebraically

Numerical method should be robust and reliable:

- Stable (convergence, increasing/oscillating error)
- Accurate “enough” (constant error)

Numerical methods for PDE

- Approximate (discretized) solution of the PDE
- Subdivide the domain in small sub-domains
- 3 major discretization methods:
  - Finite difference method (FDM): simplest form
  - Finite volume method (FVM): mainly for fluids, heat
  - Finite element method (FEM): more general

Finite Element Method (FEM)

- PDE with dependent (solution) variable u(x)
- Strong form (PDE) → weak form (integral)
- Discretization: $u(x) = \sum u_i \, v_i(x)$ v_i: basis functions
- Discretized (weak) form → Matrix form: Ku = L
- Matrix solving
- Non-linearity (convergence criteria)
- Time integration

Example

- Strong form: $-u_{xx} = 1,\ u(0) = u(1) = 0$

- Exact solution: $(x - x^2)/2$

Integration by parts ($\int f' g = f g| - \int f g'$)

- Weak form: $$-\int u_{xx}\, v = \underbrace{- u_x\, v\Big|}_{=0} + \int u_x v_x = \int u_x v_x = \int v$$

Test function v

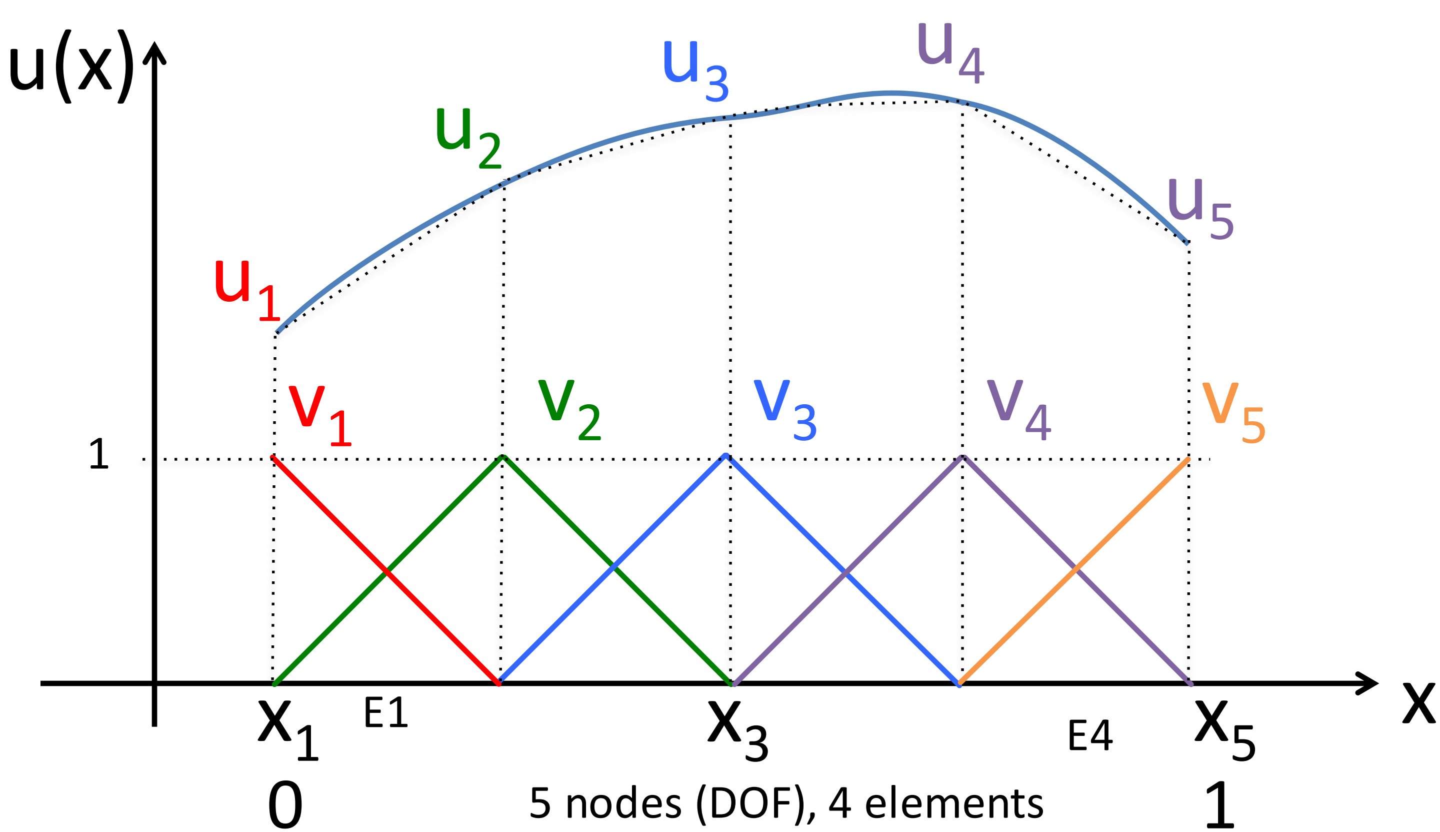
v(0)=v(1)=0

Test function v

Strong form → week form: Multiply by a test function and integrate over the domain

Discretization: basis functions v_i

$$u(x) \approx \sum u_i \, v_i(x)$$



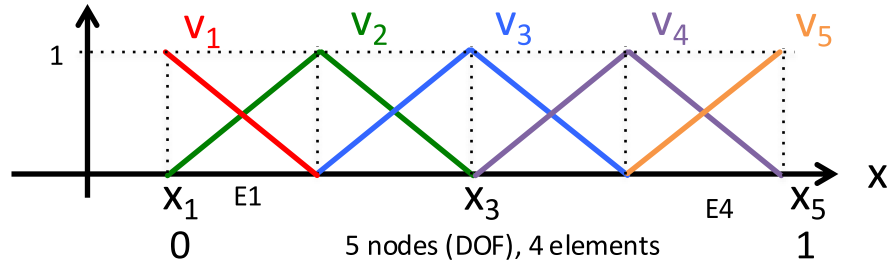
5 nodes (DOF), 4 elements

Matrix form

- $\int u_x \, v_x = \int v$ (weak form)
- For each element : $\sum u_i \int v_{xi} \, v_{xj} = \sum \int v_j$
- $\mathbf{K} = (K_{ij}) = (\int v_{xi} \, v_{xj})$, $v_{xi} = dv_i/dx$
- $\mathbf{u} = (u_i)$
- $\mathbf{L} = (L_j) = (\int v_j)$
- $\mathbf{K\,u = L}$
- Find **u** with **K** and **L** known

Stiffness matrix

- $\mathbf{K} = (K_{ij}) = (\int v_{xi}\, v_{xj})$

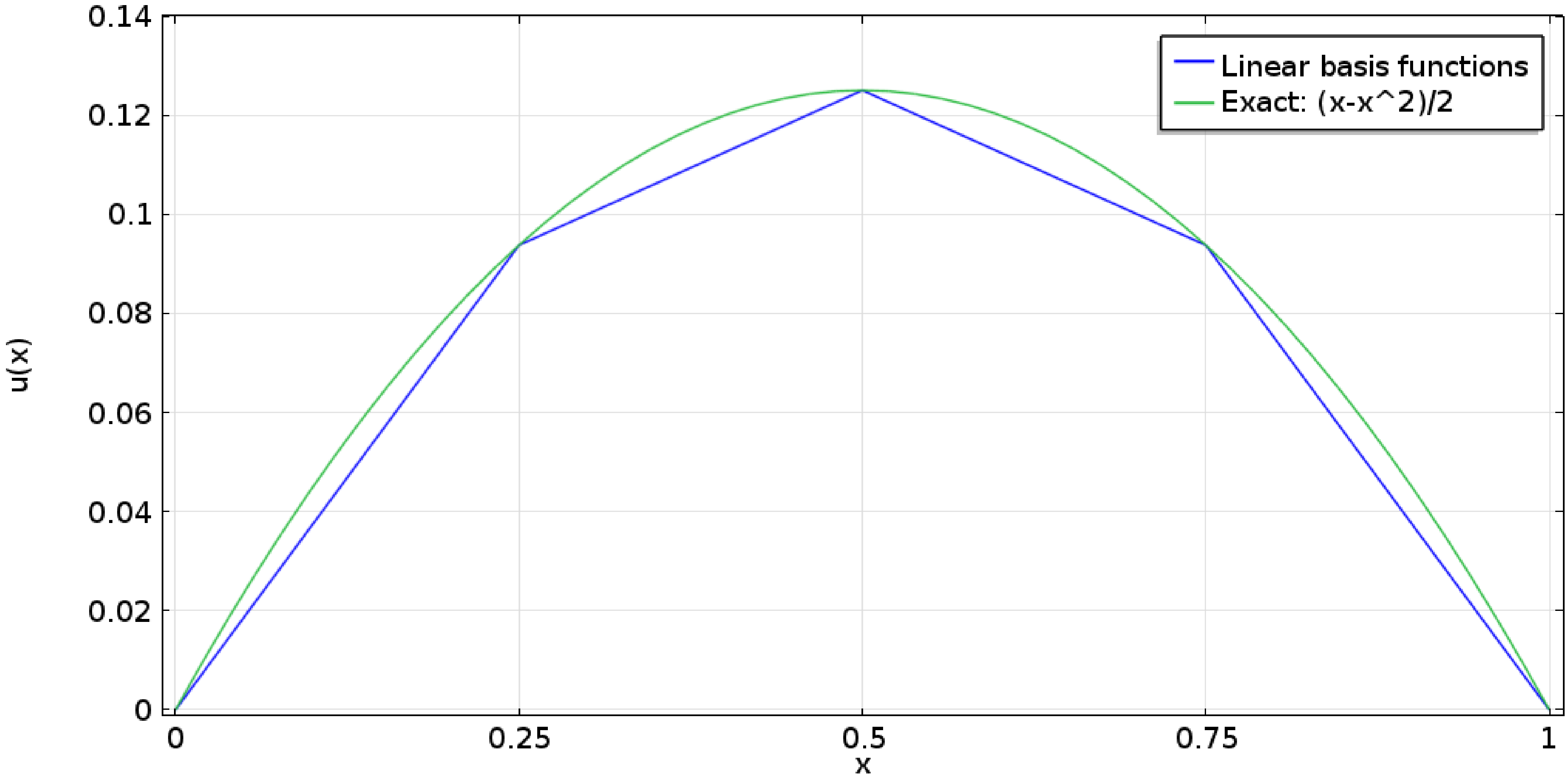


- $v_{x1} = dv_1/dx = -1/(1/4) = -4 \rightarrow K_{11} = (-4)(-4)\underbrace{(1/4)} = 4$
- $v_{x2} = dv_2/dx = \pm 4 \rightarrow K_{22} = (4)(4)\underbrace{(1/4)\ 2} = 8$
- $$K = \begin{bmatrix} 4 & -4 & 0 & 0 & 0 \\ -4 & 8 & -4 & 0 & 0 \\ 0 & -4 & 8 & -4 & 0 \\ 0 & 0 & -4 & 8 & -4 \\ 0 & 0 & 0 & -4 & 4 \end{bmatrix}$$

$\boxed{\text{Elements length}}$

- Stiffness matrix of each element → assembly

Solution



0.14
0.12
0.1
0.08
0.06
0.04
0.02
0
u(x)
0
0.25
0.5
0.75
1
x
Linear basis functions
Exact: (x-x^2)/2

Matrix form

$$\mathbf{E\,u'' + D\,u' + Ku = L}$$

- **E**: mass matrix
- **D**: damping matrix
- **K**: stiffness matrix
- **L** : load vector

$$\mathbf{u' = du/dt}$$

$$\mathbf{u'' = d^2u/dt^2}$$

FEM

- Discretization of the domain
- Evaluate the local displacement
- Evaluate the locale force
- Sum local forces
- Assembly of the matrices
- Solve the matrix system

Nonlinear Partial Differential Equations

FEM

Nonlinear Ordinary Differential Equations

Newton method (nonlinearity)

Linear Ordinary Differential Equations

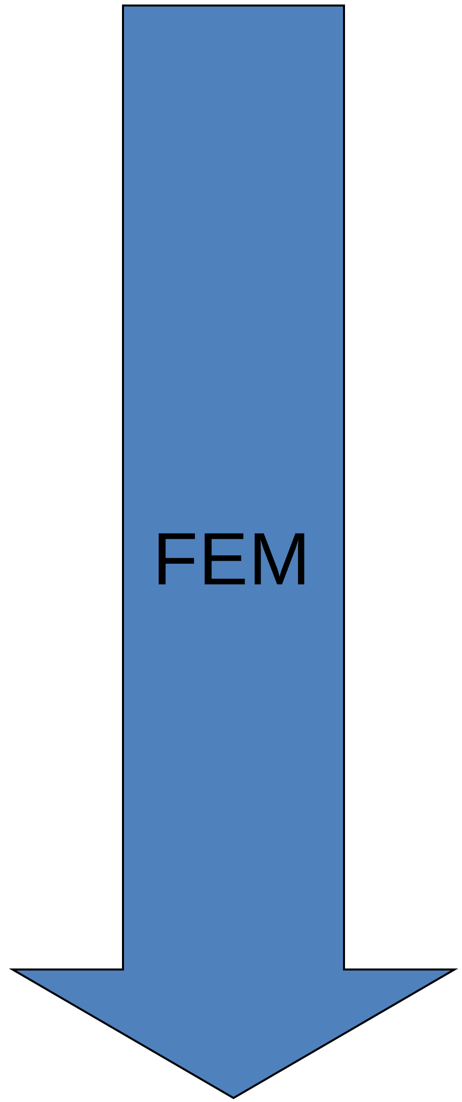
Euler method (time)

Linear Algebraic Equations

$Ku = L$ (direct/iterative, preconditioning)

Solution

Nonlinear Partial Differential Equations



Solution

Nonlinearity

- PDE — Navier-Stokes
- Geometrical: E = E(u) — Large displacement
- Material: S = S(E) — Soft tissues
- Contact: interfaces between device and tissue
- Plasticity, wear, etc.

→ Newton method

Nonlinearity

Newton method

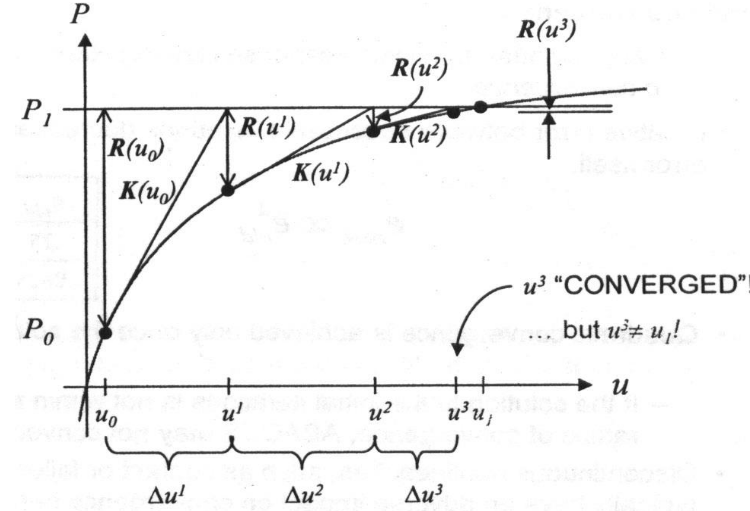
- Solve R(u) = Ku - L = 0 (R: Residue)
- Iterative method (linearization)

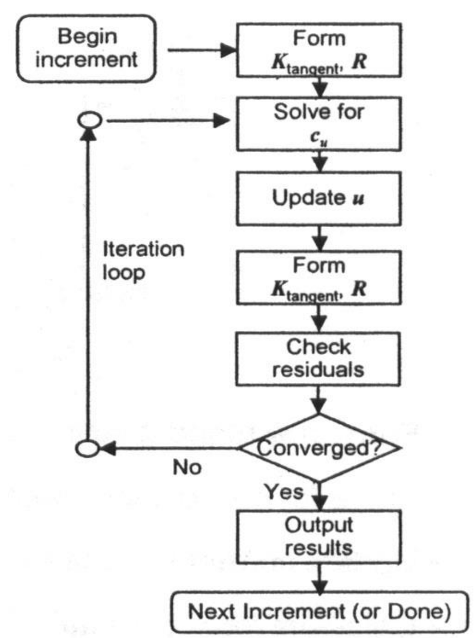
$$R(u^{i+1}) \approx R(u^i)+K^i(u^{i+1} - u^i) = 0$$

$$K^i = \text{estimate of } dR/du(u^i)$$

$$u^{i+1} = u^i + \Delta u \qquad (\Delta u : \text{correction})$$

- Convergence test: $R/F < \varepsilon$, $\Delta u/u < \varepsilon$ (F is an average force)
- Typical $\varepsilon = 0.001$





Nonlinearity

- Provide (good) initial values
- Use a parametric solver (auxillary sweep)
- Use a time-dependent instead of stationary solver
- Check for singularities
- Refine mesh in regions of high gradient
- Solve physics sequentially, and then coupled
- Stabilization techniques (artificial diffusion, viscosity, damping, etc.)

Time dependency

- Solve time dependency of PDE
- Linear Ordinary Differential Equations
- Time dependency in coefficient u_i
- $u(x,t) = \sum u_i(t)\, v_i(x)$
- Euler method

Time dependency

Euler method → Jupyter notebook: https://go.epfl.ch/course-nmb-euler

- $u'(t) = f[u(t)]$, $u(0) = u_0$

- $u_{i+1} = u_i + (1\text{-}a)\ \Delta t\ v_i + a\ \Delta t\ v_{i+1}$
  $v_i \quad = f(u_i)$
  $v_{i+1} = f(u_{i+1})$

  $a = 0$: explicit (Euler)
  $a = 1$: implicit (backward Euler) unconditionally stable
  $a = 1/2$: Crank-Nicolson

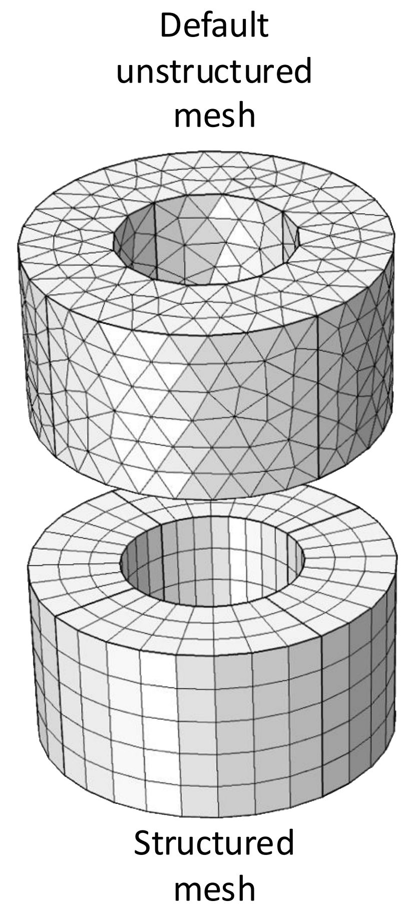
- Generalization with Runge-Kutta method

Contact

- Source-destination (master-slave) approach
  - Destination (slave) can’t penetrate source (master)
  - Source on stiffer boundary
  - Source on concave boundary
  - Destination mesh 2 times finer than source
- Iterative algorithm
  - Find contact region (penetration)
  - Apply forces to push back penetration
- Normal direction discontinuities
  → oscillations (contact chatter)

Elements

- Element type:
  - (1D) Edge,
  - (2D) Triangle, Quadrilateral,
  - (3D) Tetrahedron, Hexahedron, Prism, Pyramid
- Element (shape function) shape and order
  - 1 (linear) to 5-order polynomial (Lagrange, …)
- Meshing techniques, optimal element type
  - Structured/unstructured
  - Seeding (size)
  - Avoid sharp corners
  - Tools: sweep, boundary layers (fluid)
  - Mesh quality (Jacobian: ideal (1), bad (0), inverted (<0)



Error control

- Error = $O(h^{p+1})$
- h: size of the mesh (element)
- p: order of the polynomial basis functions
- Adaptive meshing based on error indicator

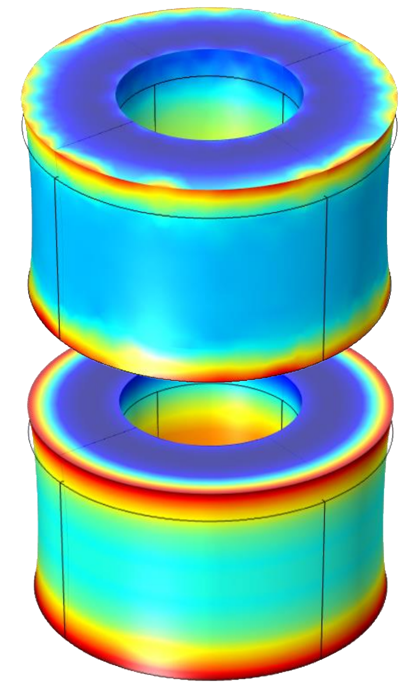
  h-adaptive refinement (mesh size)

  p-adaptive refinement (polynomial order)
- A posteriori methods (energy)

Example: Hollow cylinder

- Element size
- Linear vs. quadratic
- Degrees of freedom
- Structured mesh
- Mesh quality
- Linear vs. nonlinear solver
  - Study 1/Stationary solver/Linearity
- Direct vs. iterative solver (Ku = L)
  - Study 1/Stationary solver (right click)/Direct or Iterative
- Adaptive mesh
  - *Study 1/Step 1: Stationary/Study Extensions/Adaptation and error estimates*

Default unstructured mesh



Structured mesh

Verification and Validation

Verification: solving the equations right

Validation: solving the right equations

Check moodle documents

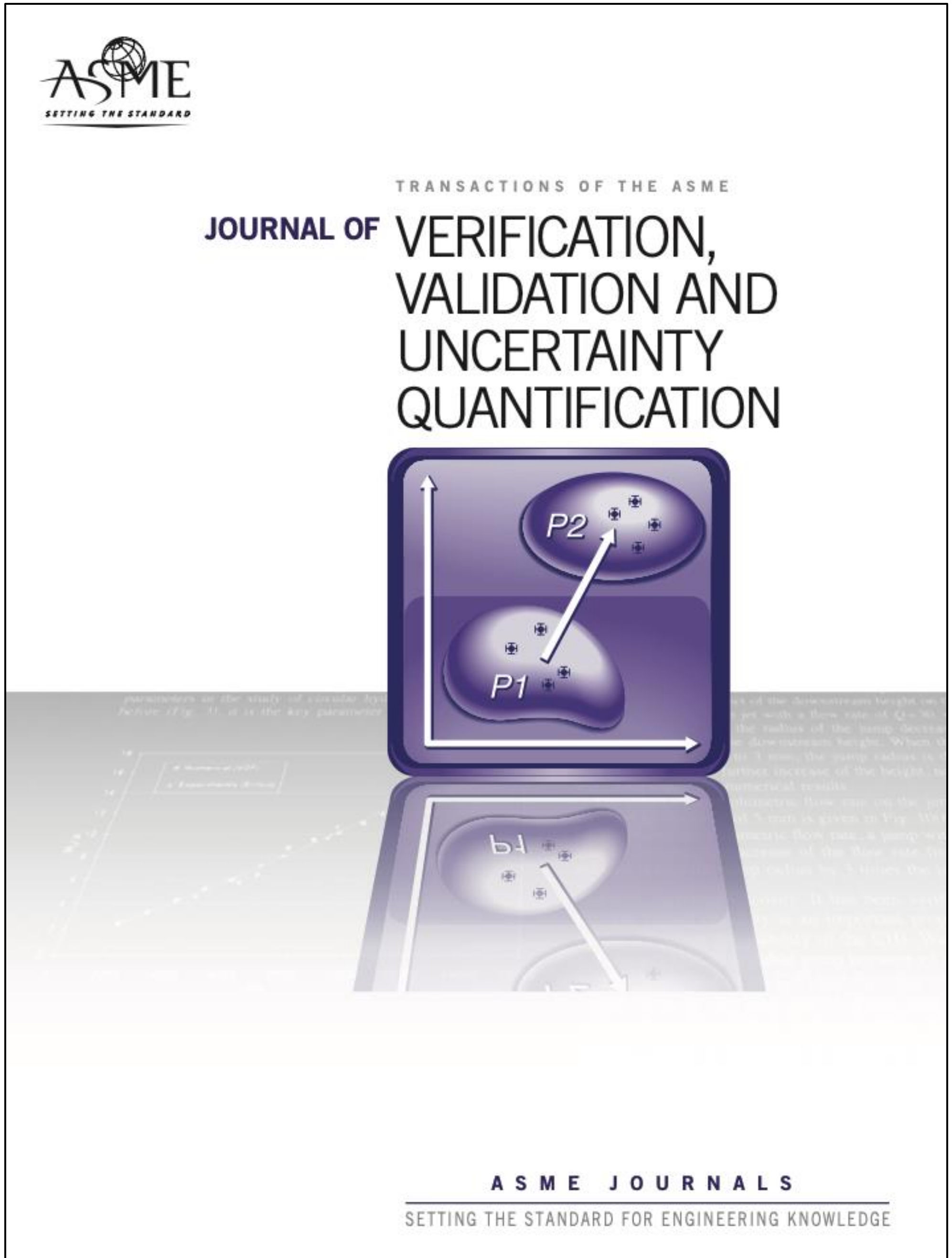
Verification

- Poor sister of validation
- Code/Calculation verification
- **Mesh convergence (discretization error)**
- Expected smoothness of solution
- Expected physical behavior
- Sensitivity analysis of (critical) model parameters
- Compare with (semi-) analytical solution (simplified model)

Validation

- Accuracy assessment (error between num. & exp.)
- Experimental error: random + systematic (value $\pm$ SD)
- Numerical error: solver type, input parameters (value $\pm$ SD)
- Statistical analysis: linear regression, slope, ordinate, correlation coefficient, Root Mean Square Error, Bland-Altman, Kolmogorov-Smirnov test
- Predictive capability of the model
- Limitations, field of application
- Validation ≠ (experimental) identification (calibration)
- Direct validation: comparison with specific experimental data
- **Indirect validation: comparison with literature**

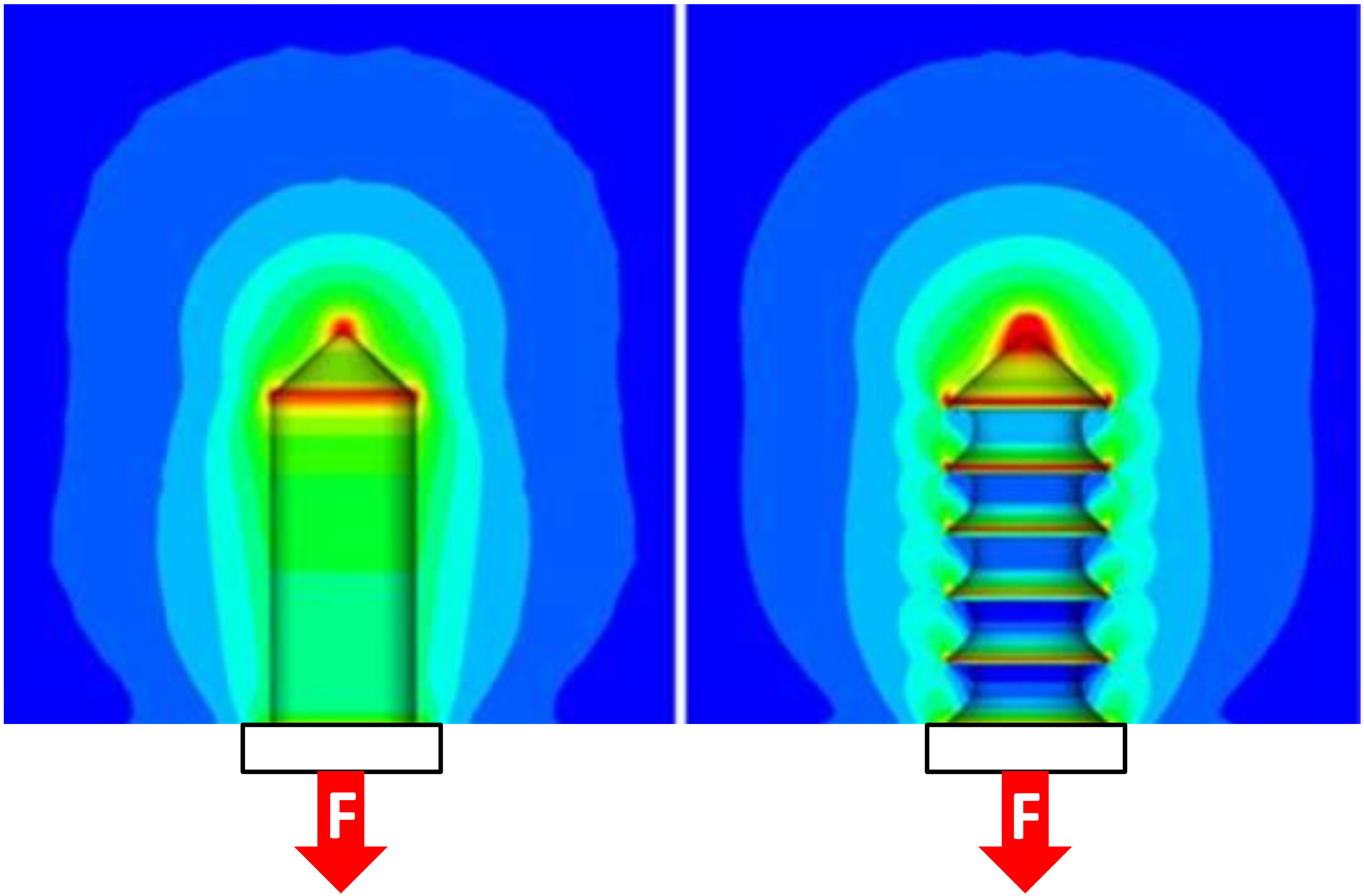
Since 2016…


ASME
SETTING THE STANDARD
TRANSACTIONS OF THE ASME
JOURNAL OF VERIFICATION, VALIDATION AND UNCERTAINTY QUANTIFICATION
P2
P1
ASME JOURNALS
SETTING THE STANDARD FOR ENGINEERING KNOWLEDGE

General advices

- Start VERY simple
- Add complexity gradually
- Verify while adding complexity
- Use symmetry (as much as possible)
- Avoid over-complex model (question-based)
- Evaluate the range of validity of the model

Screw pullout test



FEM software

- Comsol (www.comsol.com)
  - Multiphysics
- Ansys (www.ansys.com)
  - Solid, fluid (Fluent), heat, electromagnetics
- Abaqus (www.simulia.com)
  - Solid, heat, (fluid, electromagnetics)
- Elmer (www.csc.fi/elmer)
  - Multiphyisics, free
- FEniCS Project (fenicsproject.org)
  - Multiphyisics, free
- FreeFEM (www.freefem.org)
  - Multiphysics, free
- FEBiO (www.febio.org/)
  - Solid, free
- BETA CAE (www.beta-cae.com)
  - Solid, multiphysics
- ArtiSynth (artisynth.magic.ubc.ca/artisynth/)
  - Coupling Rigid body dynamics & Solid deformation, free

References

- Méthode des éléments finis en mécanique des structures, Thomas Gmür
- Introduction à l'analyse numérique, J. Rappaz and M. Picasso
- Cardiovascular Mathematics, Modeling and simulation of the circulatory system, Luca Formaggia, Alfio Quarteroni and Alessandro Veneziani
- Partial Differential Equations, Modeling and Numerical Simulation, Roland Glowinski and Pekka Neittaanmäki
- Numerical Treatment of Partial Differential Equations, Christian Grossmann, Hans-Gorg Roos and Martin Stynes
- Scientific Computing with MATLAB and Octave/ by Alfio Quarteroni, Fausto Saleri, Paola Gervasio
- Multiphysics modeling using COMSOL: a first principles approach. Roger W. Pryor
- Multiphysics modelling with finite element methods. William B. J. Zimmerman
- Verification, validation and sensitivity studies in computational biomechanics. Computer Methods in Biomechanics and Biomedical Engineering, 2007, 10:3, 171-184
- Extracting clinically relevant data from finite element simulations. Clinical Biomechanics 2005;20:451-4.
- www.asme.org/codes-standards/publications-information/verification-validation-uncertainty