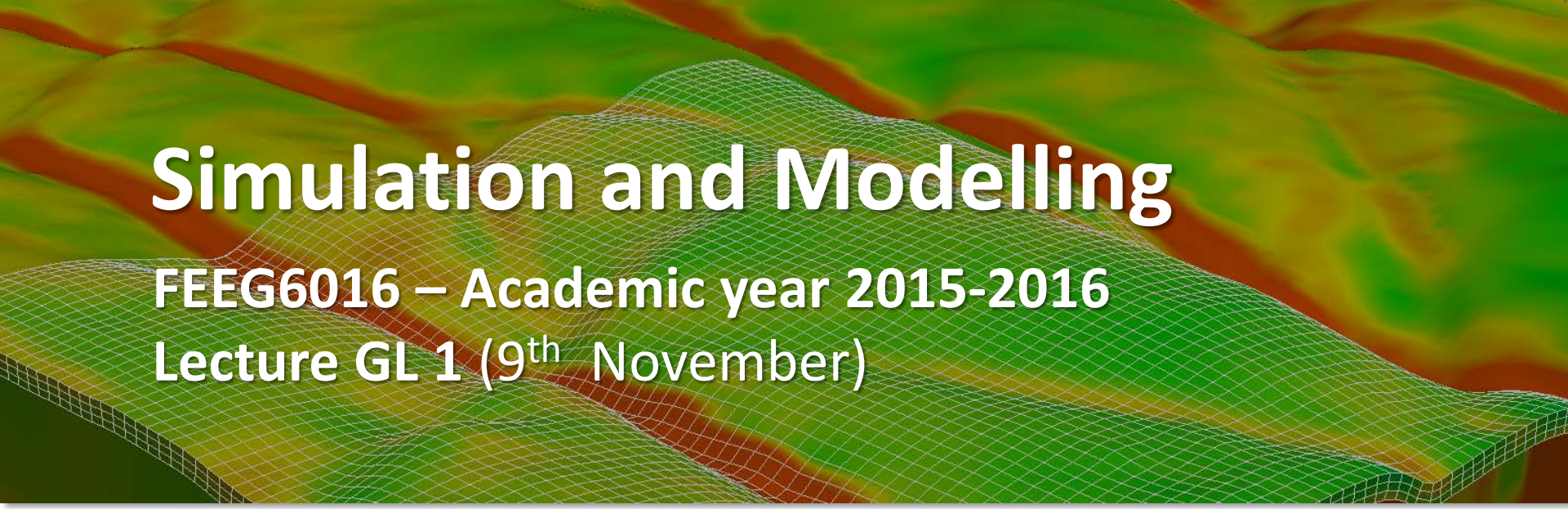


Simulation and Modelling

FEEG6016 – Academic year 2015-2016

Lecture GL 1 (9th November)

Georges Limbert [1, 2, 3]

g.limbert@soton.ac.uk | B7/4075

UNIVERSITY OF Southampton

[1] **national Centre for Advanced Tribology at Southampton**
www.southampton.ac.uk/ncats | www.biotribology.org

[2] **Bioengineering Science Research Group**
www.soton.ac.uk/ses/research/groups/bioeng.page?
Engineering Sciences, Faculty of Engineering and the Environment, University of Southampton, UK

[3] **Biomedical Engineering Association, IMechE**
Institution of Mechanical Engineers (IMechE), London, UK

Outline

- Schedule
- References
- Lecture 1

Lecture	Date	Topics
1	9 November 2014	**Introduction** Fundamental aspects of finite element methods Examples of applications of finite element methods The direct stiffness matrix method The weak form
2	16 November 2014	**Derivation of a finite element formulation** Discretisation | Shape functions Problem from elasticity

References

Finite element methods

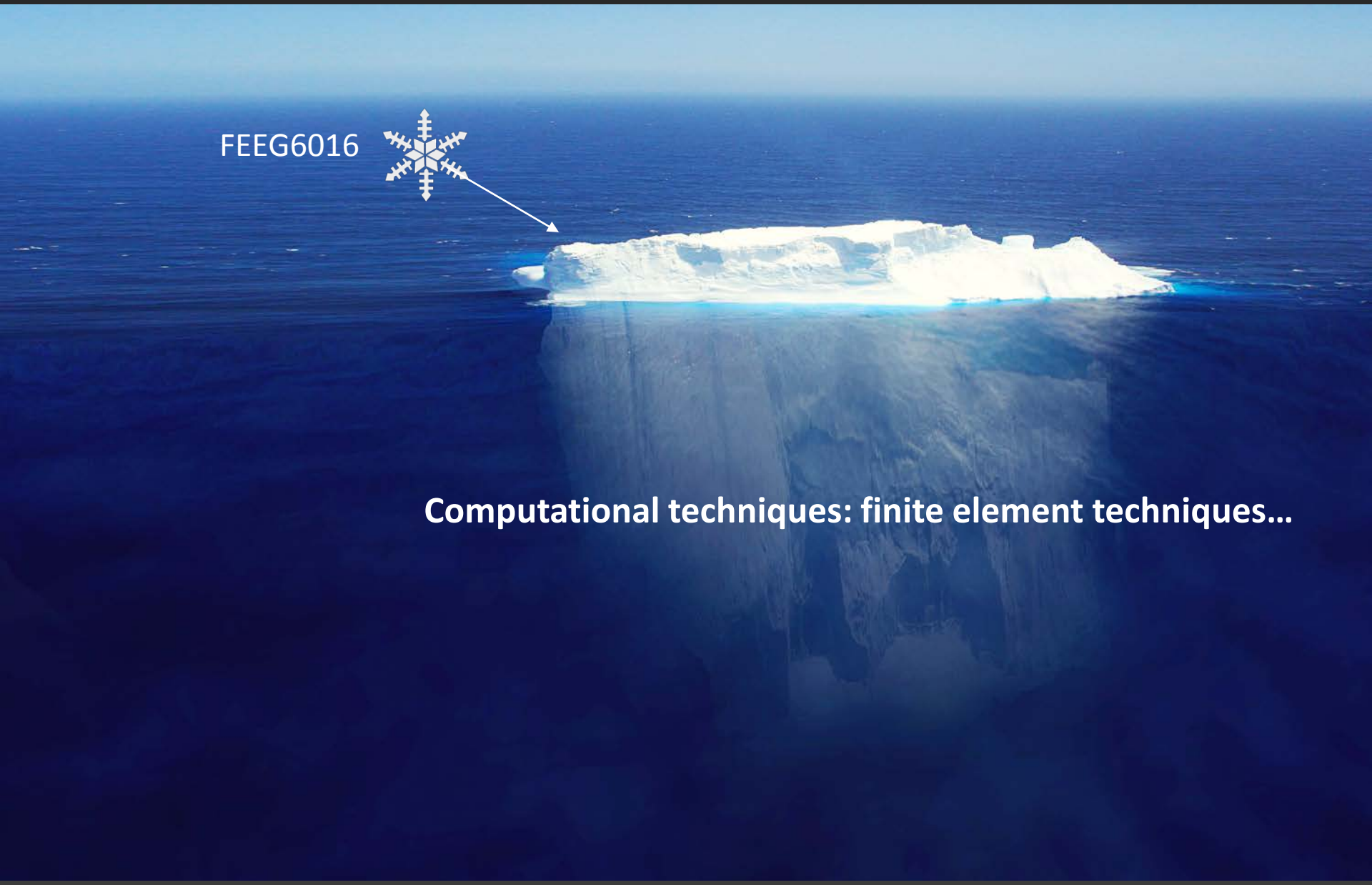
- The Finite Element Method - Linear Static and Dynamic Finite Element Analysis
  Thomas J. R. Hughes
  Dover Publications (2003)
  (a must have, highly recommended)

Non-linear finite element methods

- Nonlinear finite elements for continua and structures
  Ted Belytschko, Wing Kam Liu & Brian Moran
  John Wiley & Sons, Chichester (2000)
  (a must have, highly recommended)
- Computational Inelasticity
  Juan C. Simo & Thomas J.R. Hughes
  Springer-Verlag, New-York (1998)
  (a must have if interested in computational algorithms, highly recommended)
- Non-linear finite element analysis of solids and structures
  Michael A. Crisfield

  "Volume 1: Essentials" (1991)
  "Volume 2: Advanced topics" (1997)
  John Wiley & Sons, Chichester
  (a must have, highly recommended)

A word of caution…



Lecture 1

General introduction – the Finite Element Method

Aims

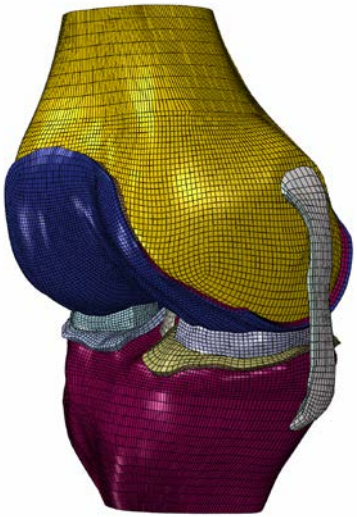
- Introduce the general “philosophy” behind the Finite Element Method (FEM)
- Rise awareness about the applications of the FEM
- Highlight the concept of finite elements through simple examples
- Introduce variational formulations of partial differential equations (PDEs)

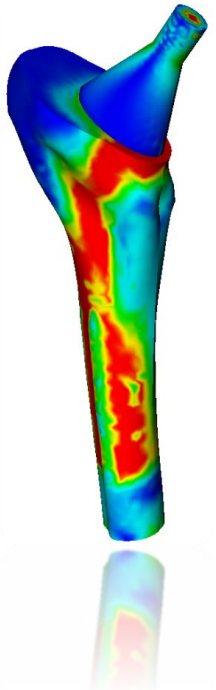
Outcomes

- Ability to understand, discuss and apply the general principles behind the FEMs
- Ability to apply a direct stiffness approach to simple spring systems to calculate displacements
- Ability to derive the weak form of any PDE

Computational engineering

- Modelling complex engineering systems
- Assessment of performance and safety before physical prototypes are built and tested
- Explore many design alternatives
- Optimise performance and safety
- Certification requirements





The Finite Element Method (FEM)

- Powerful numerical method to solve systems of partial differential equations (PDEs) over complex geometrical domains
- Rooted in variational methods
- Principles developed in the 1940's (Courant, Hrennikoff), 1950's (Feng)
- Mainly driven by the need to solve elasticity and structural analysis problems encountered in civil and aeronautical engineering
- Pioneering of applications of the FEM at Boeing in the 60's
- Terrific advances in the 60's-70's

  J. H. Argyris (University of Stuttgart, Germany)

  R. W. Clough (University of Berkeley, USA)

  O. C. Zienkiewicz | **E. Hinton** | **B. Irons** (University of Swansea, UK)

  P. G. Ciarlet (University Pierre et Marie Curie, France)

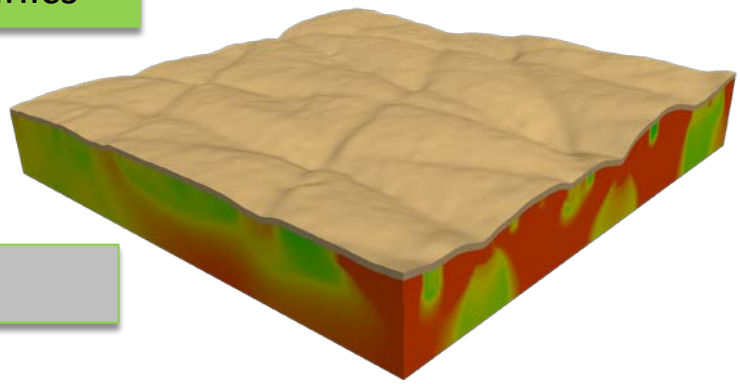
  G. Strang | **G. Fix** (MIT, USA)

  R. Gallagher (Cornell University, USA)

Domains of applications

Anywhere where time-dependent or time-independent systems of PDEs need to be solved over domain of "arbitrary" complexity

- Biophysics
- Mechanics
- Electromagnetism
- Aerodynamics
- Structural dynamics
- Heat transfer
- Porous media
- Biology
- Hydrodynamics
- Structural mechanics
- Nanomechanics
- Behavioural sciences
- Any domain of natural and physical sciences



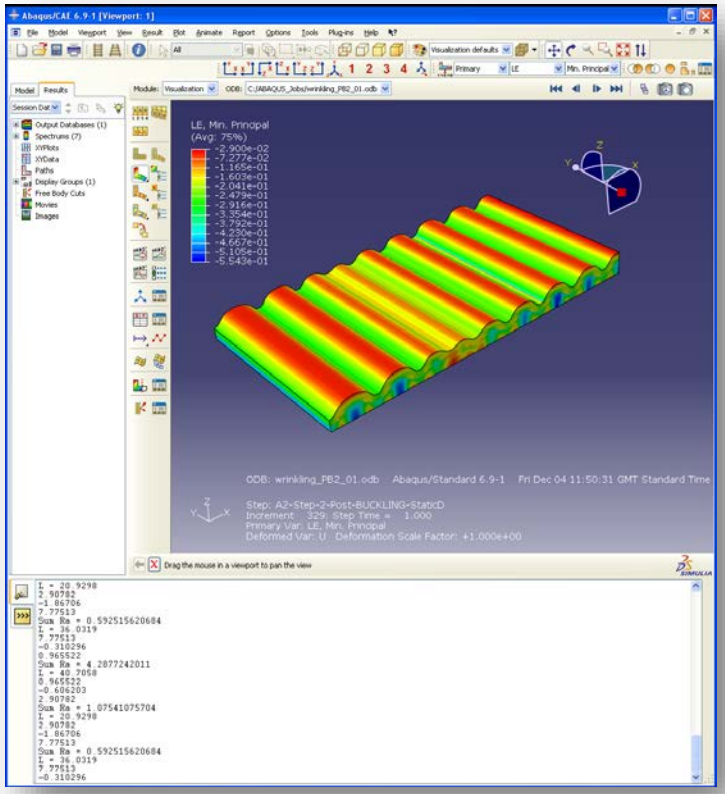
Finite element software applications: few examples

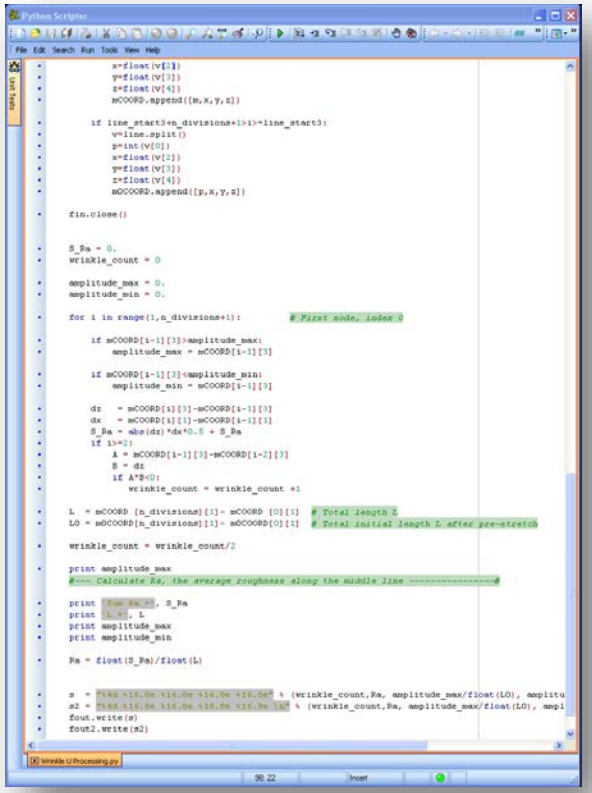
Commercial

- ABAQUS
- ANSYS
- COMSOL
- MARC
- NASTRAN
- LS-DYNA

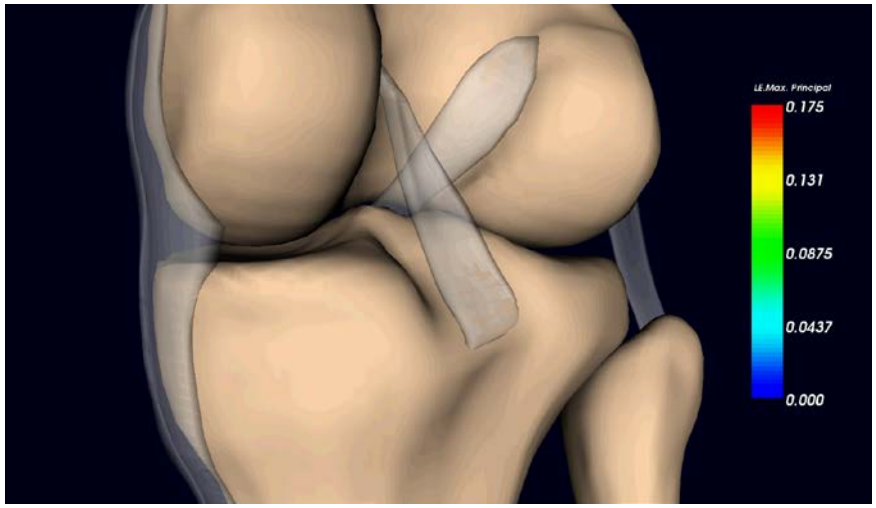
Open-source

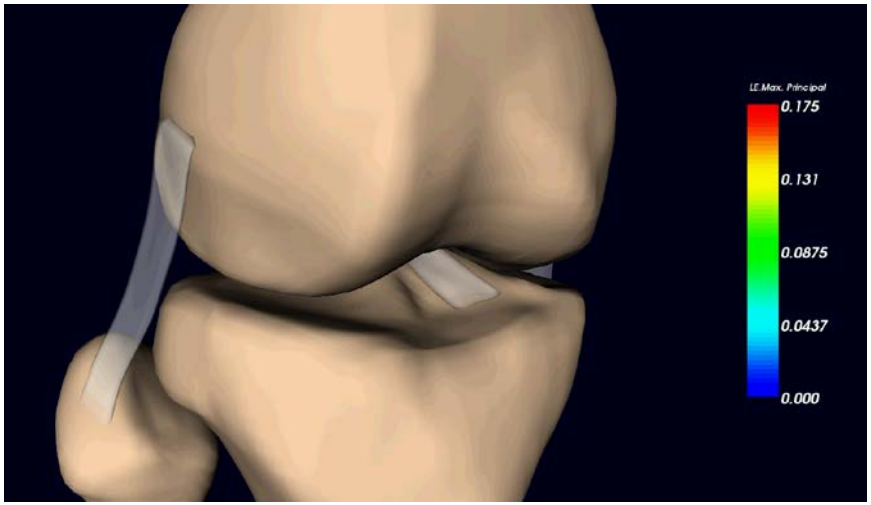
- **FEnics**
- CalculiX
- FEAPpv
- WARP3D
- Matlab toolboxes

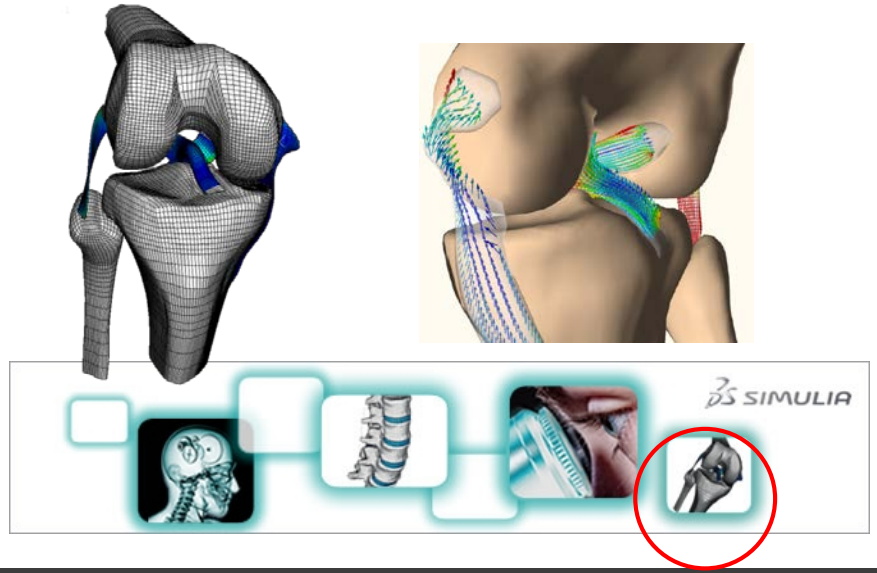


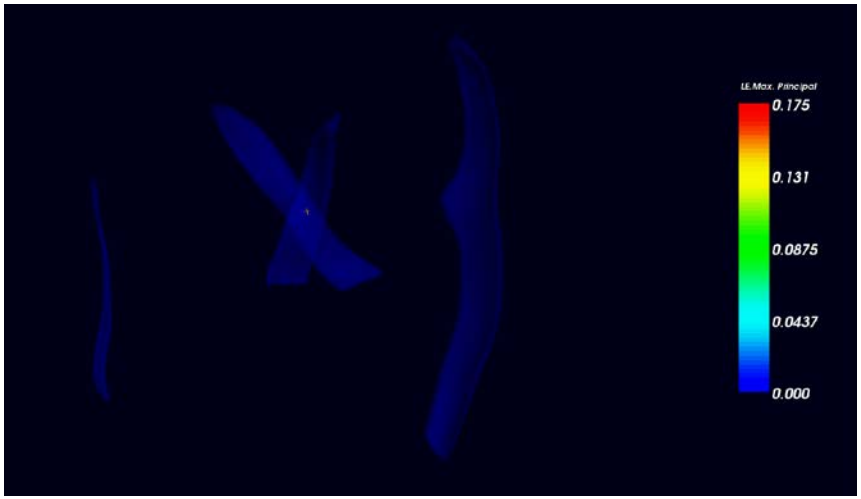


Simulated knee flexion (fibrous models of ligament)

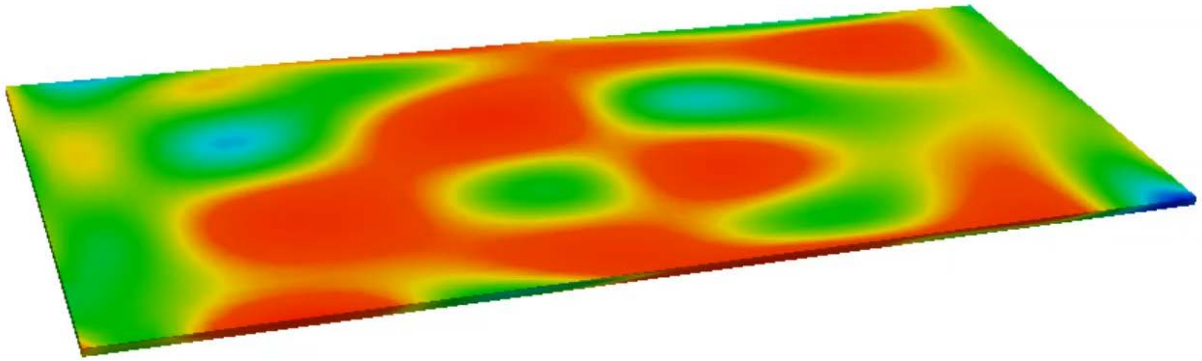




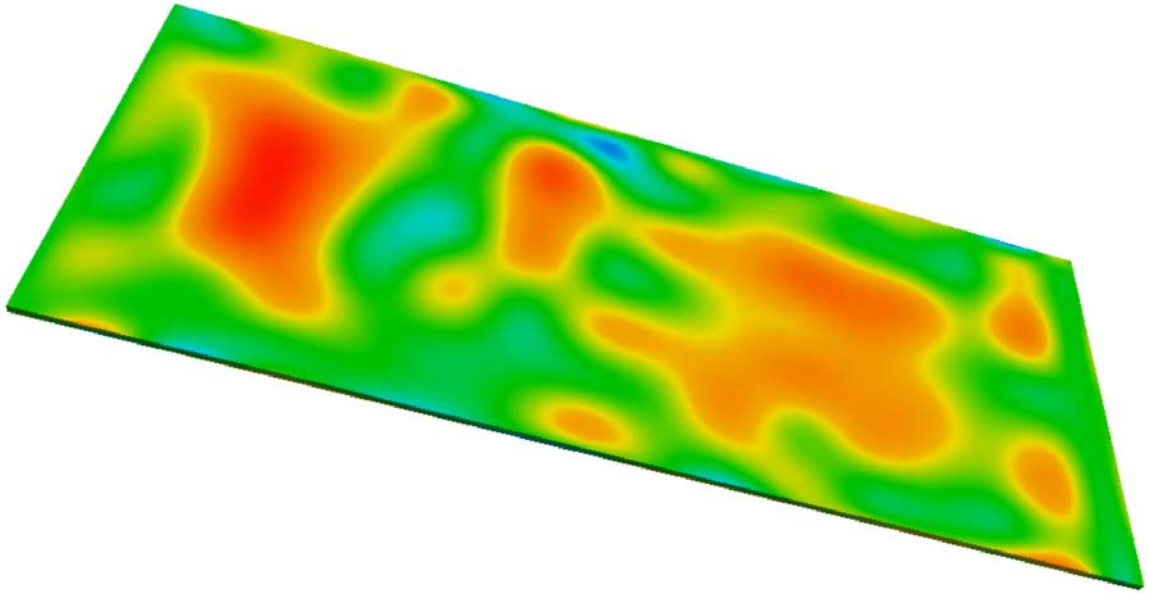




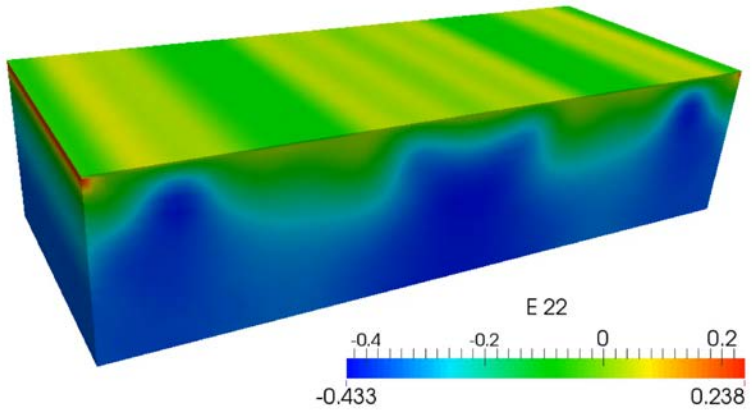
Wrinkling analyses (compression-induced)

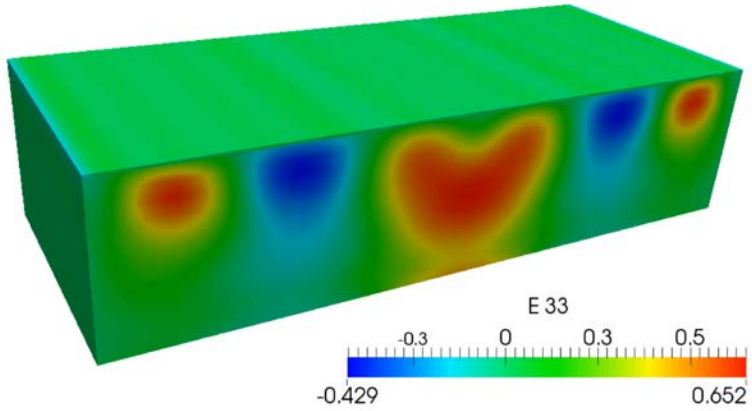


First principal strains

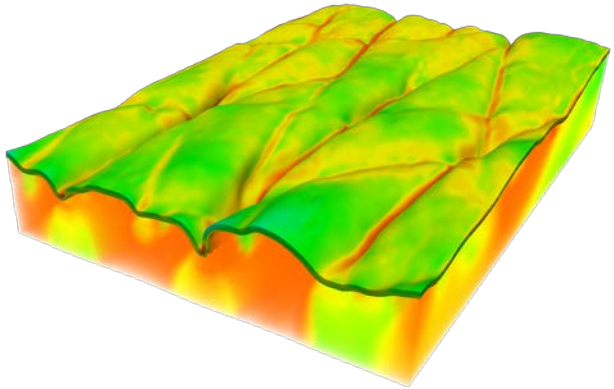


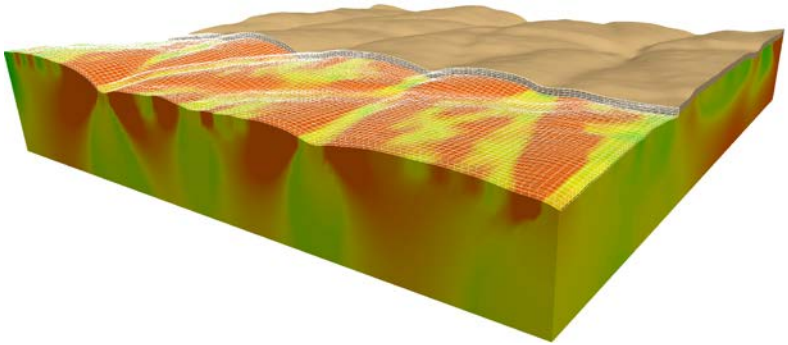
Bi-layer structure

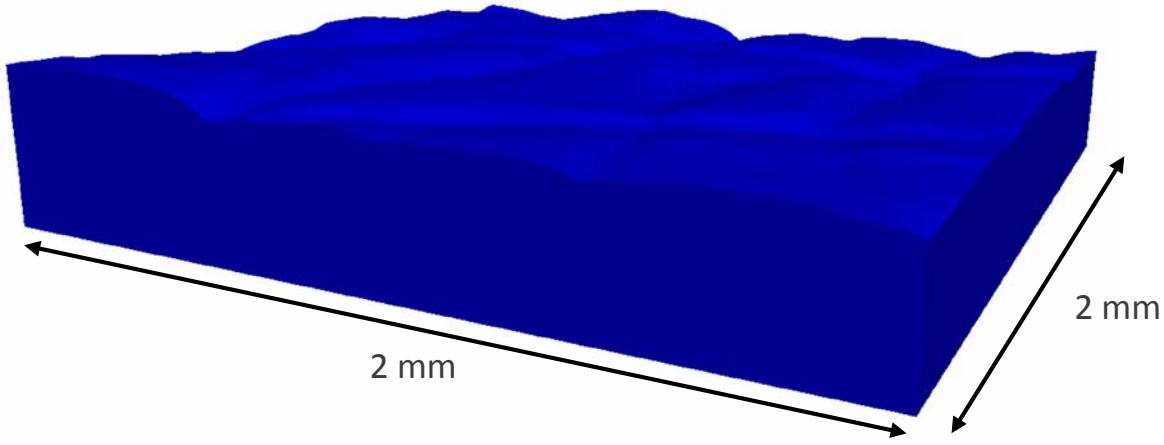




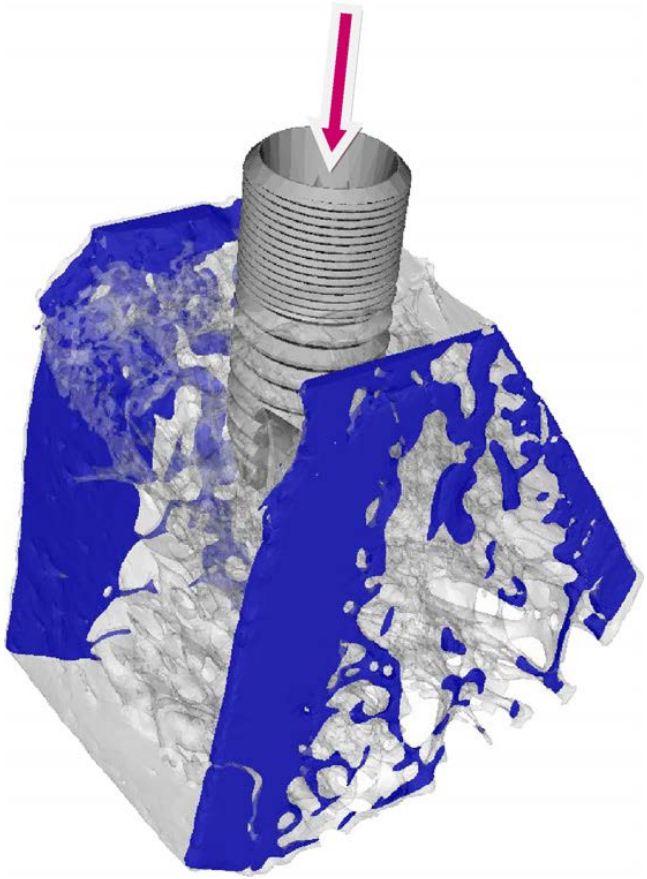
Simulation of in-plane compression of the epidermis

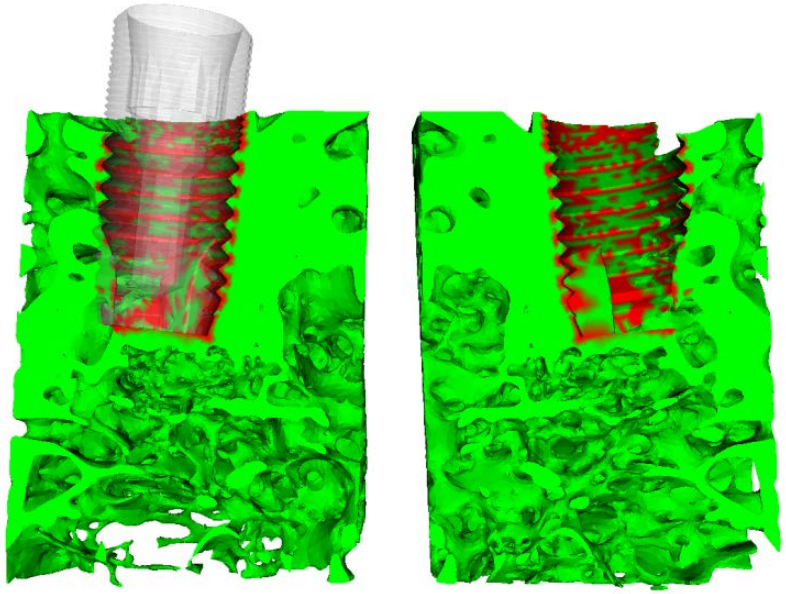




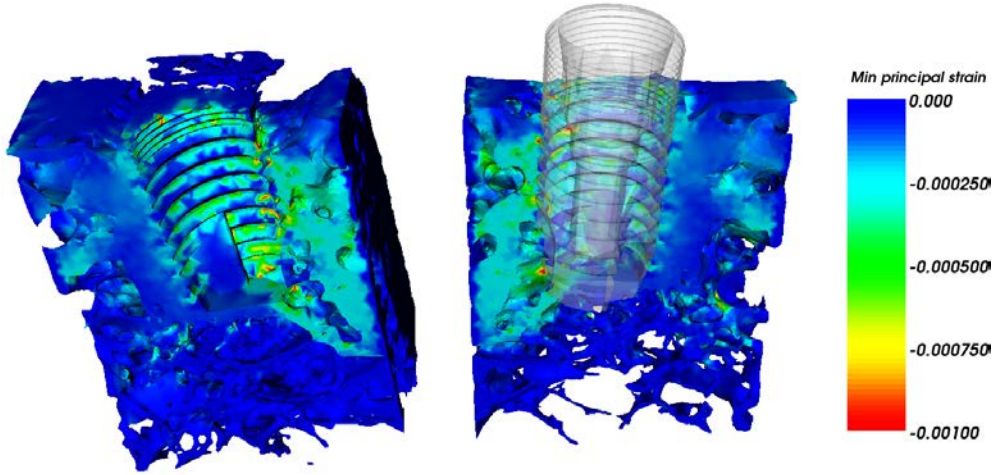


Oral implants





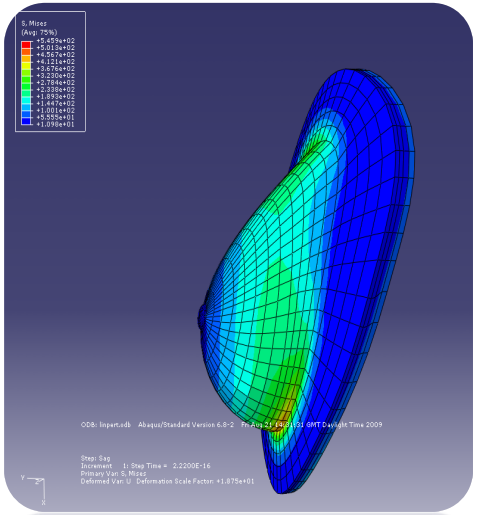
Implant micromotions



Strain magnitude in bone following implantation

Artificial organs and implants

- **Mechanics**
- **Materials**
- **Electronics**
- **Chemistry**



Biomedical: designing the ideal stent

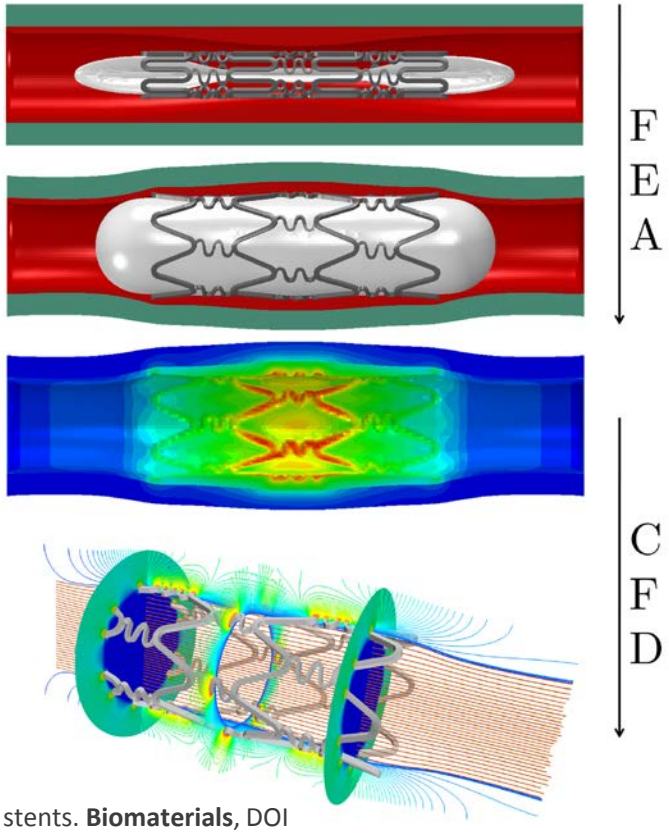
General requirements

- Good scaffolding
- Prevent arterial recoil
- High resistance to restenosis and thrombosis

Engineering aspects

- High radial strength
- Flexible
- Comfortable
- Minimum alteration of haemodynamics
- Uniform drug distribution

Sanjay Pant, University of Southampton



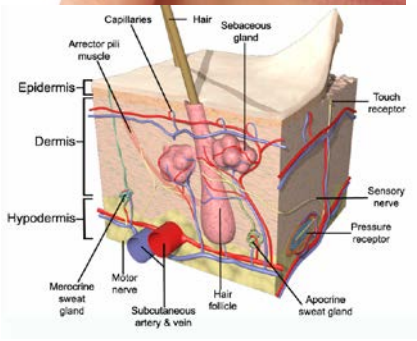
Pant, S., Bressloff, N. W., Limbert, G. and Kurzen, N. (2011) design optimisation of coronary stents. **Biomaterials**, DOI 10.1016/j.biomaterials.2011.07.059 (Opinion Leader paper)

Pant, S., Bressloff, N. W. and Limbert, G. (2011) Geometry parameterisation of multidisciplinary constrained optimisation of coronary stents, **Biomechanics and Modeling in Mechanobiology**, DOI 10.1007/s10237-011-0293-3

Pant, S., Bressloff, N. W., Forrester, A. and Kurzen, N. (2010) The influence of strut connectors in stented vessels: a comparson of pulsatile flow through five coronary stents, **Annals of Biomedical Engineering**, 38(5):1893–1907

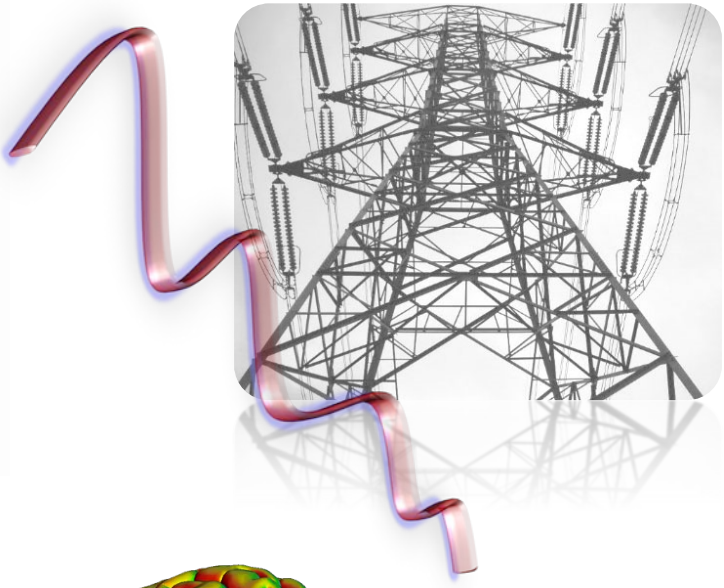
Consumer goods / cosmetics (to make you happy!)

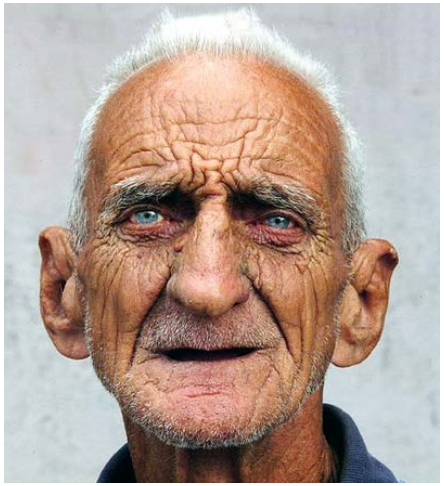
Because it's worth it

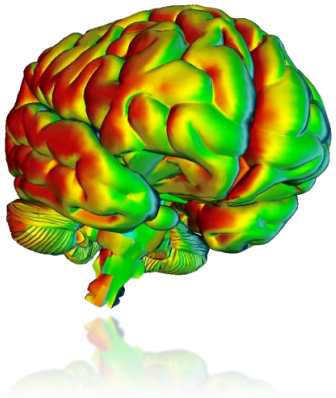


Electromagnetic radiations

- Microwave effects on biological tissues
- UV-induced damage of the skin



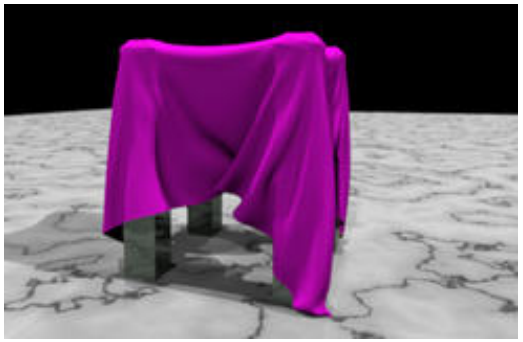




Movie / animations / games

Physics-based computer graphics is a very hot sector

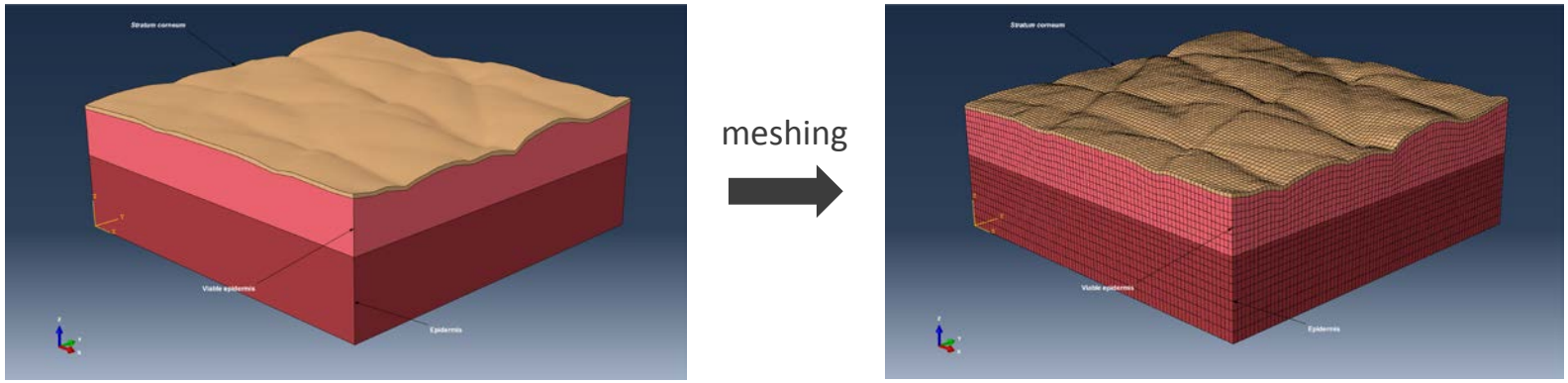
andyselle.com



Ron Fedkiw, Stanford University

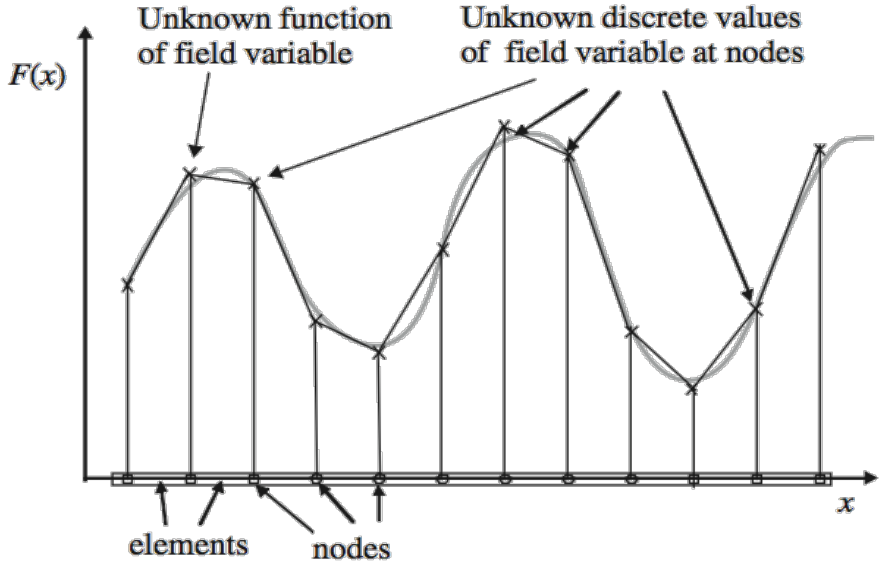
The Finite Element Method in a nutshell

- ① Transform a PDE or a system of PDEs associated with time and boundary conditions into a variational problem (focus of this lecture)
- ② Introduce a piece-wise approximation to the field variables (e.g. displacement, temperature, electric charge) in the governing equations
- ③ Discretise the physical domain into elements and write approximate equations for each element (**meshing** process). The local equations in each element can be expressed as matricial equations
- ④ Assemble the local equations of each element into a global matrix
- ⑤ Solve for the global response



FEM overview: step ②

- In each element, one need to choose **interpolation functions** (or **shape functions**) to approximate the unknown fields within each element in terms of **nodal values (degrees of freedom)**
- In each element, the value of each field variable anywhere in the element is a linear combination of the shape functions and the nodal values
- Interpolation functions are usually polynomials (Lagrange, Hermite, B-Spline, NURBS, T-Spline)



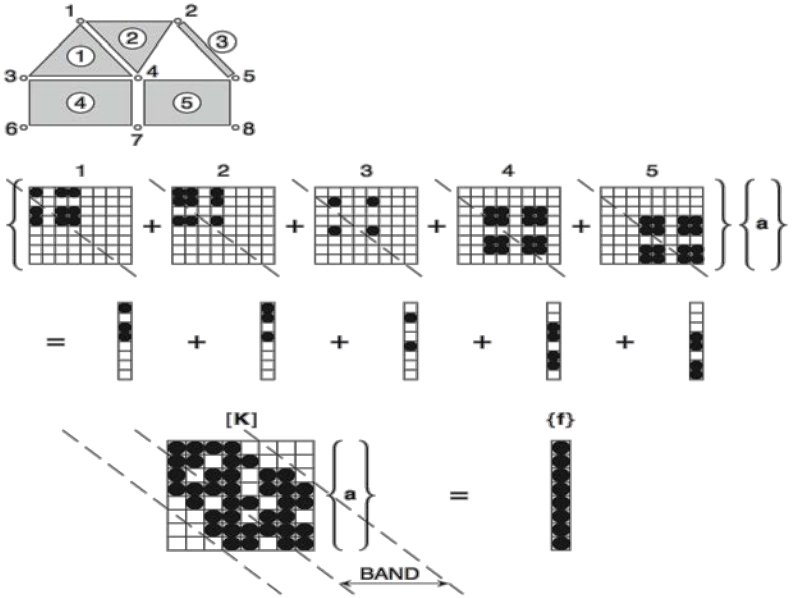
FEM overview: step ③ | step ④

Step ③

- The field approximations are injected into the variational form introduced in step ①. This is defined at the element level.
- After algebraic manipulations one obtains a general matrix equation valid for each element of the mesh

Step ④

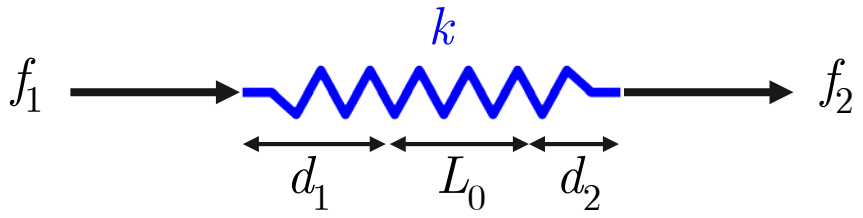
Assembly of local element-level equations into a global matrix system



Direct stiffness approach

A gentle introduction to the FEM

Equation for a single elastic spring (one element)



Equilibrium equation

$$f_1 + f_2 = 0$$

Hooke's law for the spring element

$$f_2 = k(d_2 - d_1)$$
$$f_1 = -k(d_2 - d_1) = -f_2$$

Equilibrium equations in matrix form

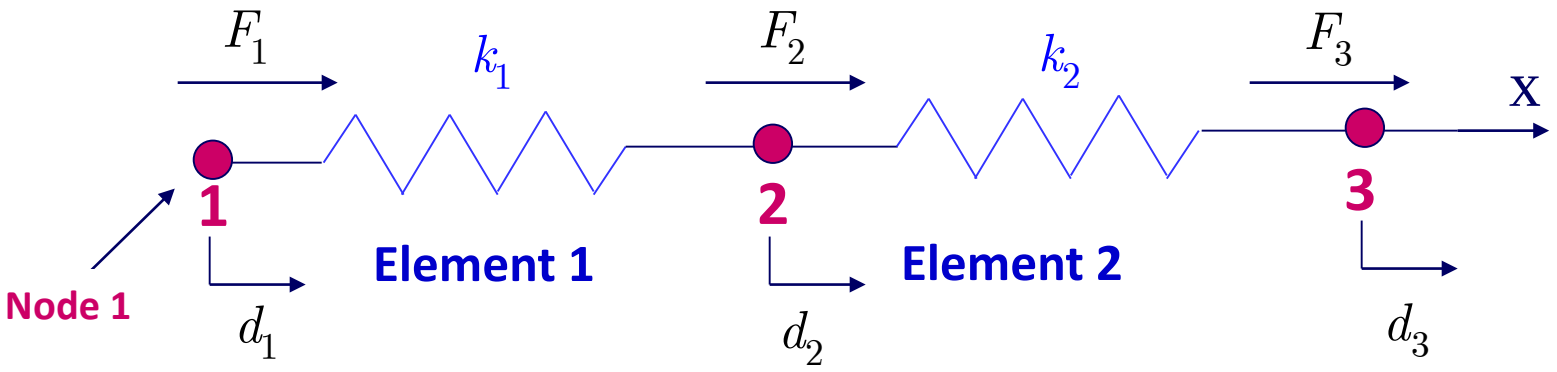
$$\underbrace{\begin{Bmatrix} f_1 \\ f_2 \end{Bmatrix}}_{\mathbf{f}} = \underbrace{\begin{bmatrix} k & -k \\ -k & k \end{bmatrix}}_{\mathbf{k}} \underbrace{\begin{Bmatrix} d_1 \\ d_2 \end{Bmatrix}}_{\mathbf{d}}$$

force stiffness displacement

The element stiffness matrix is symmetric but singular! Why?

Since the spring is not constrained in space it can attain multiple positions in space for the same nodal forces

System with 2 elastic springs



Equilibrium equation

$$F_1 + F_2 + F_3 = 0$$

Constitutive equations

$$F_1 = k_1(d_1 - d_2)$$
$$F_3 = k_2(d_3 - d_2)$$

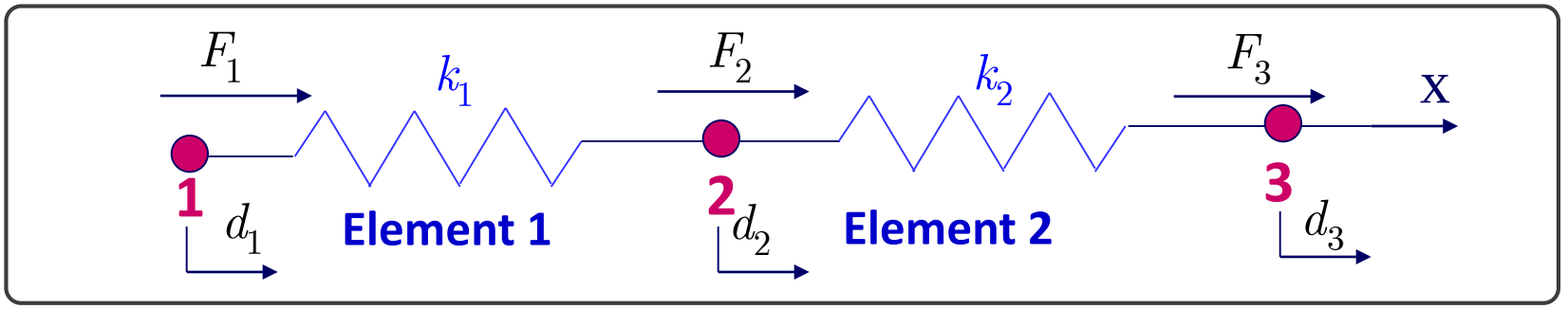
Hence

$$F_2 = -k_1 d_1 + (k_1 + k_2)d_2 - k_2 d_3$$

Equilibrium equations in matrix form

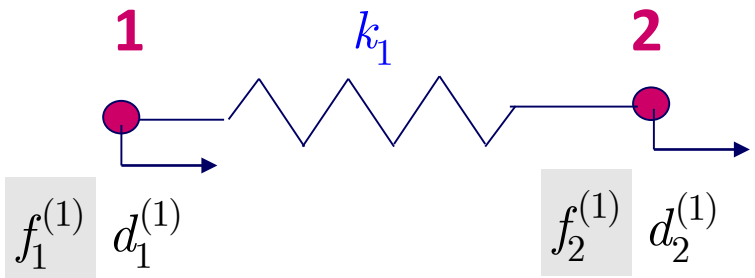
$$\underbrace{\begin{Bmatrix} F_1 \\ F_2 \\ F_3 \end{Bmatrix}}_{\mathbf{F}} = \underbrace{\begin{bmatrix} k_1 & -k_1 & 0 \\ -k_1 & k_1 + k_2 & -k_2 \\ 0 & -k_2 & k_2 \end{bmatrix}}_{\mathbf{K}} \underbrace{\begin{Bmatrix} d_1 \\ d_2 \\ d_3 \end{Bmatrix}}_{\mathbf{d}}$$

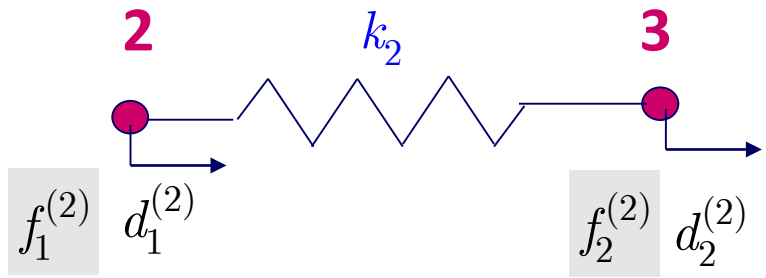
System with 2 elastic springs



Split the original structure into elemental components

$\bullet^{(n)}$:quantity • at node n



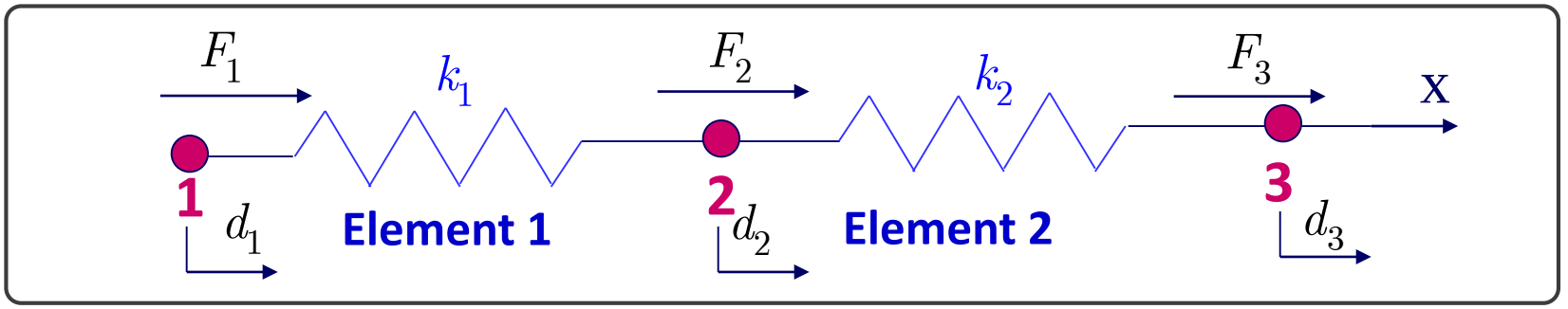


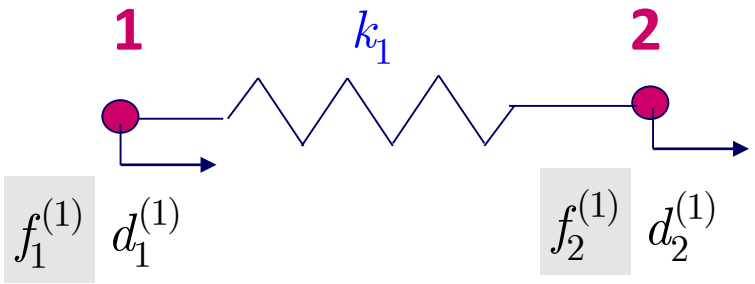
$$\underbrace{\begin{Bmatrix} f_1^{(1)} \\ f_2^{(1)} \end{Bmatrix}}_{\mathbf{f}^{(1)}} = \underbrace{\begin{bmatrix} k_1 & -k_1 \\ -k_1 & k_1 \end{bmatrix}}_{\mathbf{k}^{(1)}} \underbrace{\begin{Bmatrix} d_1^{(1)} \\ d_2^{(1)} \end{Bmatrix}}_{\mathbf{d}^{(1)}}$$

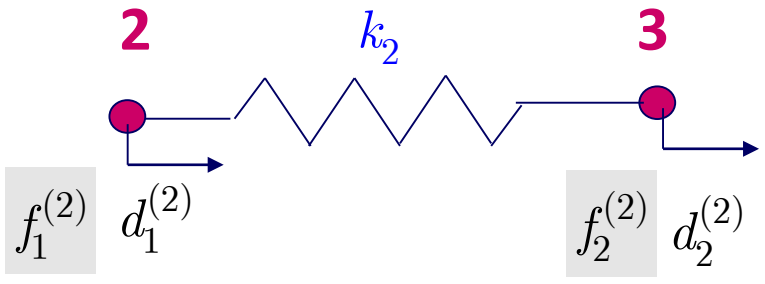
$$\underbrace{\begin{Bmatrix} f_1^{(2)} \\ f_2^{(2)} \end{Bmatrix}}_{\mathbf{f}^{(2)}} = \underbrace{\begin{bmatrix} k_2 & -k_2 \\ -k_2 & k_2 \end{bmatrix}}_{\mathbf{k}^{(2)}} \underbrace{\begin{Bmatrix} d_1^{(2)} \\ d_2^{(2)} \end{Bmatrix}}_{\mathbf{d}^{(2)}}$$

To assemble these two results into a single description of the response of the entire structure we need to link the **local** to the **global** variables

Q1: How to relate local to global displacements?





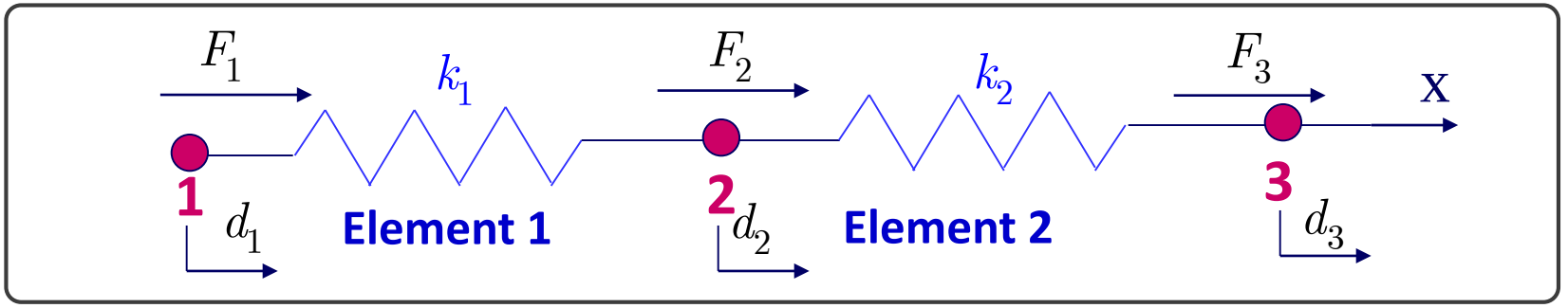


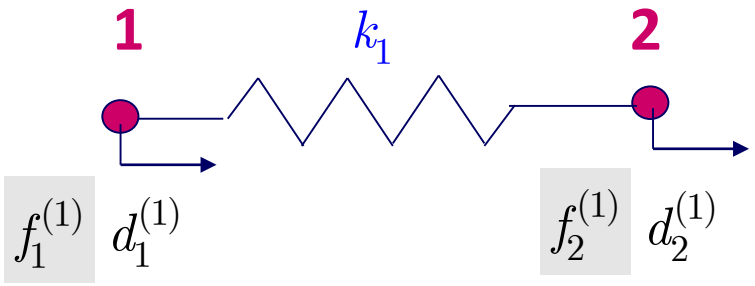
$$
\begin{aligned}
d_1^{(1)} &= d_1 \\
d_2^{(1)} &= d_1^{(2)} = d_2 \\
d_2^{(3)} &= d_3
\end{aligned}
$$

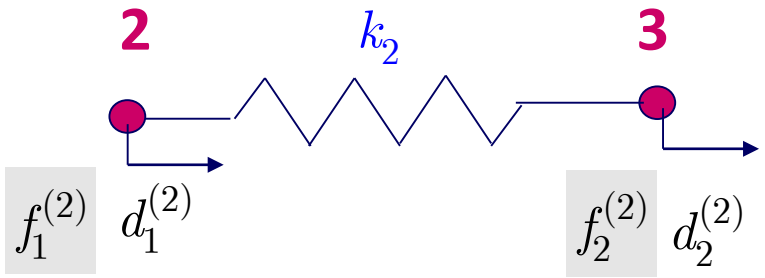
Element 1 is made of nodes 1 and 2
Element 2 is made of nodes 2 and 3

Connectivity matrix $\begin{bmatrix} 1 & 2 \\ 2 & 3 \end{bmatrix}$

Q2: How to relate local to global forces?







$$\begin{aligned} &\text{node 1:} \quad F_1 - f_1^{(1)} = 0 \\ &\text{node 1:} \quad F_2 - f_2^{(1)} - f_1^{(2)} = 0 \\ &\text{node 1:} \quad F_3 - f_2^{(2)} = 0 \end{aligned} \quad \Rightarrow \quad \begin{Bmatrix} F_1 \\ F_2 \\ F_3 \end{Bmatrix} = \begin{Bmatrix} f_1^{(1)} \\ f_2^{(1)} + f_1^{(2)} \\ f_2^{(2)} \end{Bmatrix}$$

Expanded local equations

The location equations for each spring can be rewritten as:

$$\begin{Bmatrix} f_1^{(1)} \\ f_2^{(1)} \end{Bmatrix} = \begin{bmatrix} k_1 & -k_1 \\ -k_1 & k_1 \end{bmatrix} \begin{Bmatrix} d_1 \\ d_2 \end{Bmatrix} \qquad \begin{Bmatrix} f_1^{(2)} \\ f_2^{(2)} \end{Bmatrix} = \begin{bmatrix} k_2 & -k_2 \\ -k_2 & k_2 \end{bmatrix} \begin{Bmatrix} d_2 \\ d_3 \end{Bmatrix}$$

Or, we may expand the matrices and vectors to obtain:

$$\underbrace{\begin{Bmatrix} f_1^{(1)} \\ f_2^{(1)} \\ 0 \end{Bmatrix}}_{\mathbf{f}^{(1)\mathrm{e}}} = \underbrace{\begin{bmatrix} k_1 & -k_1 & 0 \\ -k_1 & k_1 & 0 \\ 0 & 0 & 0 \end{bmatrix}}_{\mathbf{K}^{(1)\mathrm{e}}} \underbrace{\begin{Bmatrix} d_1 \\ d_2 \\ d_3 \end{Bmatrix}}_{\mathbf{d}} \qquad \underbrace{\begin{Bmatrix} 0 \\ f_1^{(2)} \\ f_2^{(2)} \end{Bmatrix}}_{\mathbf{f}^{(2)\mathrm{e}}} = \underbrace{\begin{bmatrix} 0 & 0 & 0 \\ 0 & k_2 & -k_2 \\ 0 & -k_2 & k_2 \end{bmatrix}}_{\mathbf{K}^{(2)\mathrm{e}}} \underbrace{\begin{Bmatrix} d_1 \\ d_2 \\ d_3 \end{Bmatrix}}_{\mathbf{d}}$$

$\mathbf{K}^{(1)\mathrm{e}}$ Expanded element stiffness matrix of element 1 (local)

$\mathbf{f}^{(1)\mathrm{e}}$ Expanded nodal force vector of element 1 (local)

$\mathbf{d}$ Nodal displacement vector for the entire structure (global)

Expanded local equations

Adding up the expanded element level equations

$$\underbrace{\begin{bmatrix} k_1 & -k_1 & 0 \\ -k_1 & k_1 & 0 \\ 0 & 0 & 0 \end{bmatrix}}_{\mathbf{K}^{(1)\mathrm{e}}} \underbrace{\begin{Bmatrix} d_1 \\ d_2 \\ d_3 \end{Bmatrix}}_{\mathbf{d}} + \underbrace{\begin{bmatrix} 0 & 0 & 0 \\ 0 & k_2 & -k_2 \\ 0 & -k_2 & k_2 \end{bmatrix}}_{\mathbf{K}^{(2)\mathrm{e}}} \underbrace{\begin{Bmatrix} d_1 \\ d_2 \\ d_3 \end{Bmatrix}}_{\mathbf{d}} = \underbrace{\begin{Bmatrix} f_1^{(1)} \\ f_1^{(2)} + f_2^{(1)} \\ f_2^{(2)} \end{Bmatrix}}_{\mathbf{f}}$$

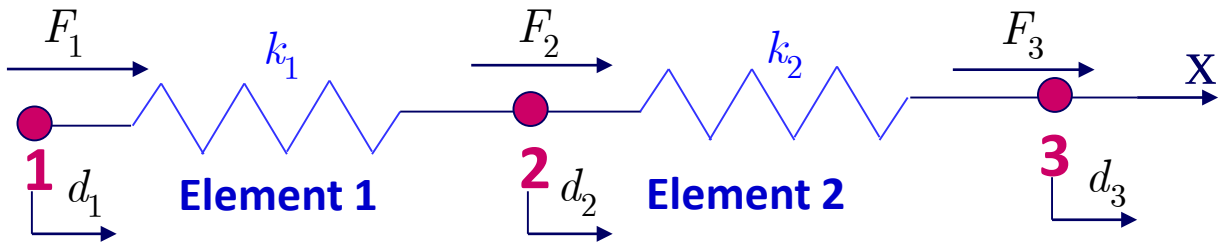
This leads to:

$$\underbrace{\begin{bmatrix} k_1 & -k_1 & 0 \\ -k_1 & k_1 + k_2 & -k_2 \\ 0 & -k_2 & k_2 \end{bmatrix}}_{\mathbf{K}} \underbrace{\begin{Bmatrix} d_1 \\ d_2 \\ d_3 \end{Bmatrix}}_{\mathbf{d}} = \underbrace{\begin{Bmatrix} f_1^{(1)} \\ f_1^{(2)} + f_2^{(1)} \\ f_2^{(2)} \end{Bmatrix}}_{\mathbf{f}}$$

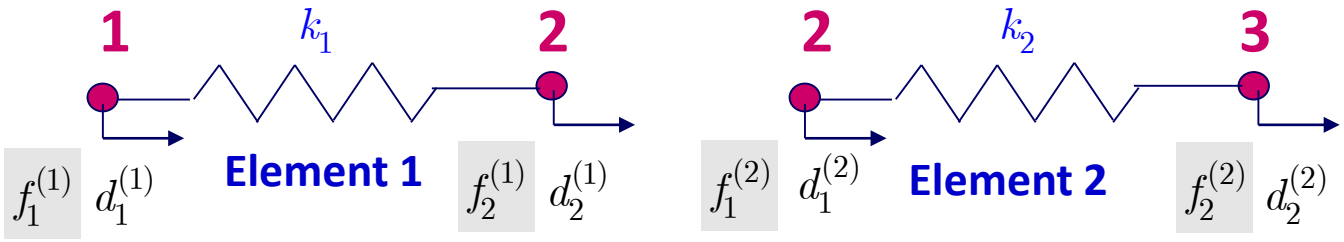
The matrix equation encodes 3 equilibrium equations

Direct assembly of the global stiffness matrix | 1

Global



Local



Node-element connectivity chart

Specifies the global node number corresponding to the local (element) node numbers

Element	Node 1 ID	Node 2 ID
1	1	2
2	2	3

Local node number

Global node number

Direct assembly of the global stiffness matrix | 2

Stifness matrix of element 1

$$\mathbf{K}^{(1)} = \begin{array}{c|cc} & \color{red}{1} & \color{red}{2} \\ \hline \color{red}{1} & k_1 & -k_1 \\ \color{red}{2} & -k_1 & k_1 \end{array}$$

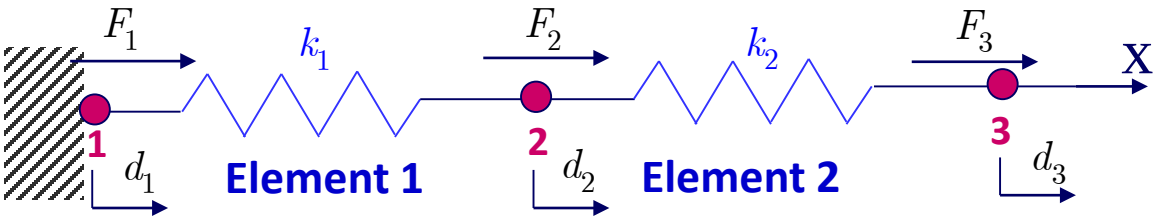
Stiffness matrix of element 2

$$\mathbf{K}^{(2)} = \begin{array}{c|cc} & \color{red}{2} & \color{red}{3} \\ \hline \color{red}{2} & k_2 & -k_2 \\ \color{red}{3} & -k_2 & k_2 \end{array}$$

Global stiffness matrix

$$\mathbf{K} = \begin{array}{c|ccc} & \color{red}{1} & \color{red}{2} & \color{red}{3} \\ \hline \color{red}{1} & k_1 & -k_1 & 0 \\ \color{red}{2} & -k_1 & k_1 + k_2 & -k_2 \\ \color{red}{3} & 0 & -k_2 & k_2 \end{array}$$

Special case: grounded spring



Partition and apply the boundary condition $d_1 = 0$

$$\left[\begin{array}{c|cc} k_1 & -k_1 & 0 \\ \hline -k_1 & k_1 + k_2 & -k_2 \\ 0 & -k_2 & k_2 \end{array}\right] \left\{\begin{array}{c} d_1 \\ \hline d_2 \\ d_3 \end{array}\right\} = \left\{\begin{array}{c} F_1 \\ \hline F_2 \\ F_3 \end{array}\right\}$$

Solve the partitioned system for d_1 and d_2

$$\begin{bmatrix} k_1 + k_2 & -k_2 \\ -k_2 & k_2 \end{bmatrix} \begin{Bmatrix} d_2 \\ d_3 \end{Bmatrix} = \begin{Bmatrix} F_2 \\ F_3 \end{Bmatrix}$$

After incorporating the boundary conditions the stiffness matrix is not singular any more

Global equilibrium equations

Q1: What is the physical meaning of K_{ij} ?

It denotes the force felt at node i due to a unit displacement at node j (keeping all other nodes fixed)

Q2: Which elements contribute to K_{ij} ?

Those between nodes i and j

Q3: Which elements contribute to K_{ii} ?

All elements that connect to node i

The stiffness matrix is invertible only when appropriate boundary conditions are prescribed

Weak form

General principles
The principle of virtual work

Turning PDEs into variational problems

The core of the recipe for turning a PDE into a **variational problem** is to multiply the PDE by a function (called a **test function**), integrate the resulting equation over the domain where the PDE is defined, and perform integration by parts of terms with second-order derivatives

This a very powerful technique which can accommodate any type of physics and is not restricted to conservative (i.e. non-dissipative) systems

Particular conditions associated with a PDE or a system of PDEs are naturally accounted for in the variational formulation

Initial boundary-value problem (IBVP)| an example

Equilibrium equation of a continuum subject to mechanical forces

PDEs	$\mathrm{div}\boldsymbol{\sigma} + \mathbf{b} = \rho\dot{\mathbf{v}} = \rho\ddot{\mathbf{u}}$
Prescribed displacement	$\mathbf{u} = \bar{\mathbf{u}} \quad on \quad \partial\Omega_u$
Prescribed traction vector	$\mathbf{t} = \bar{\mathbf{t}} \quad on \quad \partial\Omega_\sigma$
Initial condition in displacement	$\mathbf{u}(\mathbf{x},t)\Big\|_{t=0} = \mathbf{u}_0(\mathbf{X})$
Initial condition in velocity	$\dot{\mathbf{u}}(\mathbf{x},t)\Big\|_{t=0} = \dot{\mathbf{u}}_0(\mathbf{X})$

$$\begin{cases} \partial\Omega = \partial\Omega_u \cup \partial\Omega_\sigma \\ \partial\Omega_u \cap \partial\Omega_\sigma = \varnothing \end{cases}$$

$\mathrm{div}\boldsymbol{\sigma} + \mathbf{b} = \rho\dot{\mathbf{v}} = \rho\ddot{\mathbf{u}}$ is the **strong form** of the initial boundary value problem

Casting an IBVP into its weak form

Let's consider an arbitrary vector-valued function $\boldsymbol{\eta} = \boldsymbol{\eta}(\mathbf{x}) = \boldsymbol{\eta}(\chi(\mathbf{X}, t))$

This is the **test** or **weighting** function

- Time is assumed to be fixed
- $\boldsymbol{\eta}$ vanishes on the boundary of where displacements $\mathbf{u}$ are prescribed: $\partial\Omega_u$

Let's write a functional obtained by multiplying the strong form by the test function and integrating over the domain:

$$f(\mathbf{u}, \boldsymbol{\eta}) = \int_{\Omega_t} \left(-\mathrm{div}\boldsymbol{\sigma} - \mathbf{b} + \rho\ddot{\mathbf{u}}\right).\boldsymbol{\eta} dv = 0$$

Because the test function is arbitrary the weak and strong forms are equivalent:

$$f(\mathbf{u}, \boldsymbol{\eta}) = \int_{\Omega_t} \left(-\mathrm{div}\boldsymbol{\sigma} - \mathbf{b} + \rho\ddot{\mathbf{u}}\right).\boldsymbol{\eta} dv = 0 \quad \Leftrightarrow \quad \mathrm{div}\boldsymbol{\sigma} + \mathbf{b} = \rho\dot{\mathbf{v}} = \rho\ddot{\mathbf{u}}$$

Derivation of the weak form

$$f(\mathbf{u},\boldsymbol{\eta}) = \int_{\Omega_t} \left(-\mathrm{div}\boldsymbol{\sigma} - \mathbf{b} + \rho\ddot{\mathbf{u}}\right).\boldsymbol{\eta}\, dv = 0$$

$$\mathrm{div}\boldsymbol{\sigma}.\boldsymbol{\eta} = \mathrm{div}\left(\boldsymbol{\sigma}\boldsymbol{\eta}\right) - \boldsymbol{\sigma} : \mathrm{grad}\boldsymbol{\eta} = \mathrm{div}\left(\boldsymbol{\sigma}\boldsymbol{\eta}\right) - \boldsymbol{\sigma} : \nabla_{\mathbf{x}}\boldsymbol{\eta}$$

$$\rightarrow \quad f\left(\mathbf{u},\boldsymbol{\eta}\right) = \int_{\Omega_t} \left[\boldsymbol{\sigma} : \mathrm{grad}\boldsymbol{\eta} - \left(\mathbf{b} - \rho\ddot{\mathbf{u}}\right).\boldsymbol{\eta}\right] dv - \int_{\Omega_t} \mathrm{div}\left(\boldsymbol{\sigma}\boldsymbol{\eta}\right) dv$$

$$\rightarrow \quad f\left(\mathbf{u},\boldsymbol{\eta}\right) = \int_{\Omega_t} \left[\boldsymbol{\sigma} : \mathrm{grad}\boldsymbol{\eta} - \left(\mathbf{b} - \rho\ddot{\mathbf{u}}\right).\boldsymbol{\eta}\right] dv - \int_{\partial\Omega_t} \boldsymbol{\sigma}\boldsymbol{\eta}.\mathbf{n}\, ds$$

$$\int_{\partial\Omega_t} \boldsymbol{\sigma}\boldsymbol{\eta}.\mathbf{n}\, ds = \int_{\partial\Omega_u} \boldsymbol{\sigma}\boldsymbol{\eta}.\mathbf{n}\, ds + \int_{\partial\Omega_\sigma} \boldsymbol{\sigma}\boldsymbol{\eta}.\mathbf{n}\, ds$$

$\boldsymbol{\eta} = \mathbf{0}$ on $\partial\Omega_u$ $\rightarrow$

$$\int_{\partial\Omega_t} \boldsymbol{\sigma}\boldsymbol{\eta}.\mathbf{n}\, ds = \int_{\partial\Omega_\sigma} \boldsymbol{\sigma}\boldsymbol{\eta}.\mathbf{n}\, ds$$

$$\int_{\partial\Omega_\sigma} \boldsymbol{\sigma}\boldsymbol{\eta}.\mathbf{n}\, ds = \int_{\partial\Omega_\sigma} \overline{\mathbf{t}}.\boldsymbol{\eta}\, ds$$

Variational problem

$$f\left(\mathbf{u},\boldsymbol{\eta}\right)=\int_{\Omega_t}\left[\boldsymbol{\sigma}:\mathrm{grad}\boldsymbol{\eta}-\left(\mathbf{b}-\rho\ddot{\mathbf{u}}\right).\boldsymbol{\eta}\right]dv-\int_{\boxed{\partial\Omega_\sigma}}\overline{\mathbf{t}}.\boldsymbol{\eta}\,\boxed{ds}=0$$

$$\int_{\Omega_t}\mathbf{u}(\mathbf{x},t)\Big|_{t=0}.\boldsymbol{\eta}\,dv=\int_{\Omega_t}\mathbf{u}_0(\mathbf{X}).\boldsymbol{\eta}\,dv$$

$$\int_{\Omega_t}\dot{\mathbf{u}}(\mathbf{x},t)\Big|_{t=0}.\boldsymbol{\eta}\,dv=\int_{\Omega_t}\dot{\mathbf{u}}_0(\mathbf{X}).\boldsymbol{\eta}\,dv$$

Special choice of the test function

The test function is arbitrary so we can choose it to be a **virtual** displacement

$$\boxed{\boldsymbol{\eta} \equiv \delta\mathbf{u}}$$

$$f\left(\mathbf{u},\boldsymbol{\eta}\right) = \int_{\Omega_t} \left[\boldsymbol{\sigma} : \mathrm{grad}\boldsymbol{\eta} - \left(\mathbf{b} - \rho\ddot{\mathbf{u}}\right).\boldsymbol{\eta}\right] dv - \int_{\boxed{\partial\Omega_\sigma}} \overline{\mathbf{t}}.\boldsymbol{\eta}\, \boxed{ds} = 0$$

$$\int_{\Omega_t} \mathbf{u}(\mathbf{x},t)\Big|_{t=0} .\boldsymbol{\eta}\, dv = \int_{\Omega_t} \mathbf{u}_0(\mathbf{X}).\boldsymbol{\eta}\, dv$$

$$\int_{\Omega_t} \dot{\mathbf{u}}(\mathbf{x},t)\Big|_{t=0} .\boldsymbol{\eta}\, dv = \int_{\Omega_t} \dot{\mathbf{u}}_0(\mathbf{X}).\boldsymbol{\eta}\, dv$$

Principle of virtual work

Internal mechanical virtual work:

$$\delta W_{\mathrm{int}}\left(\mathbf{u},\delta\mathbf{u}\right) = \int_{\Omega_t} \boldsymbol{\sigma} : \mathrm{grad}\delta\mathbf{u}\, dv$$

External mechanical virtual work

$$\delta W_{\mathrm{ext}}\left(\mathbf{u},\delta\mathbf{u}\right) = \int_{\Omega_t} \mathbf{b}\delta\mathbf{u}\, dv + \int_{\boxed{\partial\Omega_\sigma}} \overline{\mathbf{t}}.\delta\mathbf{u}\, \boxed{ds}$$

$$\delta W_{\mathrm{int}}\left(\mathbf{u},\delta\mathbf{u}\right) = \delta W_{\mathrm{ext}}\left(\mathbf{u},\delta\mathbf{u}\right) - \int_{\Omega_t} \rho\ddot{\mathbf{u}}\delta\mathbf{u}\, dv$$

Initial boundary-value problem

Strong form

$$\mathrm{div}\boldsymbol{\sigma} + \mathbf{b} = \rho\dot{\mathbf{v}} = \rho\ddot{\mathbf{u}}$$

$$\mathbf{u} = \bar{\mathbf{u}} \quad on \quad \partial\Omega_u$$

$$\mathbf{t} = \bar{\mathbf{t}} \quad on \quad \partial\Omega_\sigma$$

$$\mathbf{u}(\mathbf{x},t)\Big|_{t=0} = \mathbf{u}_0(\mathbf{X})$$

$$\dot{\mathbf{u}}(\mathbf{x},t)\Big|_{t=0} = \dot{\mathbf{u}}_0(\mathbf{X})$$

$$\begin{cases} \partial\Omega = \partial\Omega_u \cup \partial\Omega_\sigma \\ \partial\Omega_u \cap \partial\Omega_\sigma = \varnothing \end{cases}$$

Weak form

$$\int_{\Omega_t} \rho\ddot{\mathbf{u}}.\delta\mathbf{u}\, dv = \delta W_{ext}(\mathbf{u},\delta\mathbf{u}) - \delta W_{int}(\mathbf{u},\delta\mathbf{u})$$

$$\delta W_{\text{ext}}(\mathbf{u},\delta\mathbf{u}) = \int_{\Omega_t} \mathbf{b}.\delta\mathbf{u}\, dv + \int_{\partial\Omega_\sigma} \bar{\mathbf{t}}.\delta\mathbf{u}\, ds$$

$$\delta W_{\text{int}}(\mathbf{u},\delta\mathbf{u}) = \int_{\Omega_t} \boldsymbol{\sigma} : \mathrm{grad}\delta\mathbf{u}\, dv$$

$$\int_{\Omega_t} \mathbf{u}(\mathbf{x},t)\Big|_{t=0}.\delta\mathbf{u}\, dv = \int_{\Omega_t} \mathbf{u}_0(\mathbf{X}).\delta\mathbf{u}\, dv$$

$$\int_{\Omega_t} \dot{\mathbf{u}}(\mathbf{x},t)\Big|_{t=0}.\delta\mathbf{u}\, dv = \int_{\Omega_t} \dot{\mathbf{u}}_0(\mathbf{X}).\delta\mathbf{u}\, dv$$

Laboratory workshop 1

Poisson problem

- Derivation of the weak form
- Development of a Python programme to compute the solution over a 2D domain
- Comparison of results with exact solutions

Inflation of an elastic membrane

- Derivation of the weak form
- Development of a Python programme to compute the solution over a 2D domain
- Comparison of results with exact analytical solutions