Non-Linear Finite Element Methods in Solid Mechanics

Attilio Frangi, attilio.frangi@polimi.it
Outline

Lesson 1-2: introduction, linear problems in statics
Lesson 3: dynamics
Lesson 4: locking problems
Lesson 5: geometrical non-linearities
Lesson 6-7: small strain plasticity

Final exam: project?

Slides on web-site (search page on www.dica.polimi.it)

Send e-mail with photo and name

Strong links with the course “Non-Linear Solid Mechanics”
Planning

Basic notions in linear elasticity

pre-requisite of the course a “fully operative” knowledge is mandatory
Planning

Time dependent problems
Lesson 3: Dynamics (and diffusion)

Temperature chart in an engine

Impact of a tyre on a surface. Rolling of a tyre on an inclined surface
Planning

Element Engineering
Lesson 4: Pathologies and cures of isoparametric finite elements

incompressibility.. rubber (tyres), beams, shells, plasticity..
Planning

Non linear quasi-static problems
Lesson 5: Introduction. Application to geometrical non-linearities

buckling of structures
Planning

Elastoplasticity

Lesson 6: “Local” issues
Lesson 7: “Global” issues

plastic deformation in an exaust-pipe
Textbook

originated from a course on FEM taught at the Ecole Polytechnique for 5 years

9 modules
(1h30 theory + 2h hands on sessions)

codes free for download from http://www.ateneonline.it

AIM: couple a rigorous theoretical treatment with an the introduction to FEM coding

Not only toy codes, but “pilot” codes prior to “serious” implementations
Governing equations in strong form

Field equations:

\[ \varepsilon(x) = \frac{1}{2}(\nabla u + \nabla^T u)(x) \quad (x \in \Omega), \]
\[ \text{div } \sigma(x) + \rho f(x) = 0 \quad (x \in \Omega), \]
\[ \sigma(x) = \mathcal{A} : \varepsilon(x) \quad (x \in \Omega) \]

If material isotropic:

\[ \sigma = \lambda \text{Tr} \varepsilon I + 2 \mu \varepsilon = \kappa \text{Tr} \varepsilon I + 2 \mu \varepsilon \]

Boundary conditions:

\[ u(x) = u^D(x) \quad (x \in S_u), \]
\[ \sigma(x) \cdot n(x) = T^D(x) \quad (x \in S_T). \]
Admissible spaces

- Space of regular displacements (associated to a bounded energy)
  \[ C = \left\{ \mathbf{v} \mid \mathbf{v} \text{ continuous over } \Omega, \int_{\Omega} \varepsilon[\mathbf{v}] : \mathbf{A} : \varepsilon[\mathbf{v}] \ d\mathbf{V} < +\infty \right\} \]

- Space of fields compatible with boundary data
  - **Kinematically admissible** displacements:
    \[ C(\mathbf{u}^D) = \left\{ \mathbf{v} \mid \mathbf{v} \in C \text{ and } \mathbf{v} = \mathbf{u}^D \text{ on } S_u \right\} \]
  - **Statically admissible** stresses
    \[ S(\mathbf{T}^D, f) = \left\{ \mathbf{\tau} \mid \text{div} \mathbf{\tau} + \rho \mathbf{f} = 0 \text{ in } \Omega, \mathbf{\tau} \cdot \mathbf{n} = \mathbf{T}^D \text{ on } S_T \right\} \]

- Space of displacements compatible with zero boundary displacements
  \[ C(0) = \left\{ \mathbf{v} \mid \mathbf{v} \in C \text{ and } \mathbf{v} = 0 \text{ on } S_u \right\} \]

**Formulation of the equilibrium problem in linear elasticity (strong form):**

find \( \mathbf{u} \in C(\mathbf{u}^D) \) and \( \mathbf{\sigma} \in S(\mathbf{T}^D, f) \) such that \( \mathbf{\sigma}(\mathbf{x}) = \mathbf{A} : \varepsilon[\mathbf{u}](\mathbf{x}) \) (\( \mathbf{x} \in \Omega \))
“Weak” problem formulation

- Weak form of local equilibrium equations:

\[
\int_{\Omega} \sigma : \varepsilon[w] \, dV = \int_{\Omega} \rho f \cdot w \, dV + \int_{\partial\Omega} [\sigma \cdot n] \cdot w \, dS \quad \forall w \in C
\]

- stems from an integration by parts procedure of:

\[
\int_{\Omega} (\text{div} \, \sigma + \rho f) \cdot w \, dV = 0 \quad \forall w \in C
\]

- corresponds essentially to a form of the Principle of Virtual Power (PPV)

- Compatibility equation and constitutive law enforced pointwise

\[
\sigma = A : \varepsilon[u]
\]

\[
\int_{\Omega} \varepsilon[u] : A : \varepsilon[w] \, dV = \int_{\Omega} \rho f \cdot w \, dV + \int_{S_u} T \cdot w \, dS + \int_{S_T} T^D \cdot w \, dS \quad \forall w \in C
\]
Variant: eliminate unknown tractions

\[
\int_\Omega \varepsilon[u] : \mathbf{A} : \varepsilon[w] \, dV = \int_\Omega \rho f \cdot w \, dV + \int_{S_u} T \cdot w \, dS + \int_{S_T} T^D \cdot w \, dS \quad \forall w \in \mathcal{C}
\]

Assume that the two following conditions can be met (we will see later how)

- restrict \( w \) to be kinematically admissible with zero boundary data: choose \( w \) in
  \[
  \mathcal{C}(Q) = \{ \nu \mid \nu \in \mathcal{C} \text{ and } \nu = 0 \text{ on } S_u \}
  \]
- \( u \) satisfies a priori boundary conditions \textit{in a strong form}: \( u = u^D \)

Hence the problem formulation becomes:

\[
\text{find } u \in \mathcal{C}(u^D) \text{ such that } \int_\Omega \varepsilon[u] : \mathbf{A} : \varepsilon[w] \, dV = \int_\Omega \rho f \cdot w \, dV + \int_{S_T} T^D \cdot w \, dS \quad \forall w \in \mathcal{C}(Q)
\]

most FEM codes are based on this variant - displacement FEM
Galerkin approach for the weak formulation

Weak formulation of the linear elastic problem

\[ \int_{\Omega} \varepsilon[u] : A : \varepsilon[w] \, dV = \int_{\Omega} \rho f \cdot w \, dV + \int_{S_T} T^D \cdot w \, dS \quad \forall w \in C(0) \]

The choice of the unknown and of the virtual fields:

\[ u(x) = u^{(D)}(x) + \sum_{K=1}^{N} \alpha_K \varphi^K(x) \quad \Rightarrow u \in C(u^D) \]

\[ w(x) = \sum_{J=1}^{N} \alpha_J^* \varphi^J(x) \quad \Rightarrow w \in C(0) \]

leads to the linear system of equations:

\[ \forall \{ \alpha^* \} \in \mathbb{R}^N, \quad \{ \alpha^* \}^T [K] \{ \alpha \} - \{ \alpha^* \}^T \{ F \} = \{ 0 \} \]

\[ \Rightarrow \quad [K] \{ \alpha \} = \{ F \} \]
Galerkin approach for the weak formulation

\[ u_N(x) = u^{(D)}(x) + \sum_{K=1}^{N} \alpha_K \varphi^K(x) \]

\[ w_N(x) = \sum_{J=1}^{N} \alpha^*_J \varphi^J(x) \]

leads to the linear system of equations:

\[ \forall \{\alpha^*\} \in \mathbb{R}^N, \quad \{\alpha^*\}^T [K] \{\alpha\} - \{\alpha^*\}^T \{F\} = \{0\} \]

\[ \implies [K] \{\alpha\} = \{F\} \]

- List of generalized displacements: \( \{\alpha\} = \{\alpha_1, \ldots, \alpha_N\}^T \)

- Stiffness matrix \([K]\):

\[ K_{IJ} = \int_{\Omega} \varepsilon[\varphi^I] : A : \varepsilon[\varphi^J] \, dV \quad (1 \leq I, J \leq N) \]

- List of generalized forces \( \{F\} \):

\[ F_I = -\int_{\Omega} \varepsilon[u^{(D)}] : A : \varepsilon[\varphi^I] \, dV + \int_{\Omega} \rho f \cdot \varphi^I \, dV + \int_{S_T} T^D \cdot \varphi^I \, dS \quad (1 \leq I \leq N) \]
Galerkin approach: general properties (1/3)

Let us express the solution $u$ as $u = u_N + \Delta u$
($\Delta u$ is the “error” of the numerical solution $u_N$ with respect to the exact solution $u$)

- Virtual field
  
  $w_N(x) = \sum_{J=1}^{N} \alpha^*_J \varphi^J(x)$

- Weak continuum formulation (written for the exact solution $u$)
  
  $\int_{\Omega} \varepsilon[u] : A : \varepsilon[w_N] \, dV = \int_{S_T} T^D \cdot w_N \, dS$

- Weak discrete formulation (written for the approximate solution $u_N$)
  
  $\int_{\Omega} \varepsilon[u_N] : A : \varepsilon[w_N] \, dV = \int_{S_T} T^D \cdot w_N \, dS$

The error $\Delta u$ is orthogonal to every virtual field belonging to the space where the solution is sought (in the sense of the “energy norm”)

$\int_{\Omega} \varepsilon[\Delta u] : A : \varepsilon[w_N] \, dV = 0$
Galerkin approach: general properties (2/3)

Deformation energy of $u - v_N = \Delta u + (u_N - v_N)$

with arbitrary kinematically admissible $v_N = u^{(D)} + \sum_{K=1}^{N} \alpha_K \phi^K$

$$\int_{\Omega} \varepsilon[u - v_N] : A : \varepsilon[u - v_N] \, dV$$

$$= \int_{\Omega} \varepsilon[\Delta u] : A : \varepsilon[\Delta u] \, dV + \int_{\Omega} \varepsilon[u_N - v_N] : A : \varepsilon[u_N - v_N] \, dV$$

$$+ 2 \int_{\Omega} \varepsilon[\Delta u] : A : \varepsilon[u_N - v_N] \, dV$$

$$= 0 \text{ (orthogonality)}$$

Property of best approximation: $u_N$ is the best approximation of the exact solution $u$ in the selected space of approximation, in the sense of the energy norm:

$$\int_{\Omega} \varepsilon[\Delta u] : A : \varepsilon[\Delta u] \, dV \leq \int_{\Omega} \varepsilon[u - v_N] : A : \varepsilon[u - v_N] \, dV$$
Galerkin approach: general properties (3/3)

Deformation energy of exact solution $u = u_N + \Delta u$:

\[
\int_{\Omega} \varepsilon[u] : A : \varepsilon[u] \, dV = \int_{\Omega} \varepsilon[u_N] : A : \varepsilon[u_N] \, dV + \int_{\Omega} \varepsilon[\Delta u] : A : \varepsilon[\Delta u] \, dV \\
+ 2 \int_{\Omega} \varepsilon[u_N] : A : \varepsilon[\Delta u] \, dV
\]

Assumption: $u^D = 0$. Hence $u \in C(0)$ and then:

\[
\int_{\Omega} \varepsilon[u_N] : A : \varepsilon[\Delta u] \, dV = 0 \quad \text{(orthogonality)}
\]

**Property 3:** if $u \in C(0)$, $u_N$ approximates $u$ from below in the energy norm sense:

\[
\int_{\Omega} \varepsilon[u_N] : A : \varepsilon[u_N] \, dV < \int_{\Omega} \varepsilon[u] : A : \varepsilon[u] \, dV
\]
Introduction to FEM Galerkin approach

A partition of $\Omega$ into triangular elements sharing nodes. This introduces a discretized $\Omega_h$.

Neighbouring elements always share nodes.

The notion of conformity:
- either be well separated
- or share one node
- or share one edge

Typically, the mesh is created with dedicated codes.

In our simple 2D case, GMSH.

(i) $\Omega_h$ tends to $\Omega$ when $h \to 0$;
(ii) $\Omega_h = \Omega$ if $\Omega$ has piecewise straight boundaries.

$h = \max_{\text{elements}} \frac{d}{d}$
Let us consider one scalar field (e.g. one component of displ. or a temperature field)

We draw “nodal values” of the field...

The blue line denotes the discretization of an $S_u$ region (imposed displacements)
Linear interpolation at the local level

Let us now focus on a specific element

The three nodal values completely define the displacement field within the element.

• the assumed displacement field is **continuous**
• its restriction to each triangle is **linear** and depends only on **nodal values**

\[
\mathbf{v}_h(\mathbf{x}) = c_0 + c_1 x_1 + c_2 x_2 \quad \leftrightarrow \quad \mathbf{v}_h(\mathbf{x}) = N_k(x_1, x_2) v^{(k)} + N_\ell(x_1, x_2) v^{(\ell)} + N_m(x_1, x_2) v^{(m)}
\]

This is only a particular way to express a linear field!
Shape functions (local shape functions)

\[ v_h(x) = N_k(x_1, x_2) \nu^{(k)} + N_\ell(x_1, x_2) \nu^{(\ell)} + N_m(x_1, x_2) \nu^{(m)} \]

\( N_k, N_\ell \) and \( N_m \) are called shape functions and are:

(i) Linear in \((x_1, x_2)\);

(ii) Satisfy the property \( N_k(x^{(\ell)}) = \delta_{k\ell} \)

sometimes are called “local” or “elemental” shape functions as opposed to “global” shape functions (see later)
Galerkin interpolation

Global approximation: specific form of the Galerkin approach with:

\[ v_h(x) = \sum_{n=1}^{N_N} \tilde{N}_n(x) v^{(n)} = \sum_{n \mid x^{(n)} \in S_u} \tilde{N}_n(x) u^D(x^{(n)}) + \sum_{n \mid x^{(n)} \notin S_u} \tilde{N}_n(x) v^{(n)} \]

- Analysis domain: \( \Box_h \)
- \( v_h \in C_h(u^D) \): kinematically admissible in the sense of FEM approximation
Isoparametric elements: linear triangle revisited

Moreover area coordinates coincide with shape functions!!!

$$v_h = a_1 v^{(1)} + a_2 v^{(2)} + a_3 v^{(3)}$$

ISO-parametric
(geometry and displacement)
Shape functions are now imagined defined in the parametric space on the master element: \( N_k(\alpha) = \alpha_k \)

\[
N_k(\alpha^{(\ell)}) = \delta_{k\ell}
\]
Isoparametric elements: quadratic triangle

Isoparametric elements: quadratic triangle

\[ x_i = \sum_{k=1}^{6} N_k(a_1, a_2) x_i^{(k)} \]

\[ x_i = c_i^{(0)} + c_i^{(1)} a_1 + c_i^{(2)} a_2 + c_i^{(3)} a_1^2 + c_i^{(4)} a_1 a_2 + c_i^{(5)} a_2^2 \]

**physical triangle**

**master triangle**

A quadratic mapping from **master triangle** into **physical triangle** using area coordinates allows to better approximate curved boundaries!

The discretized and real domain still DO NOT coincide everywhere but node position is respected exactly!

The same shape functions are then employed to generate the approximation space for displacements

\[ \nu_h = \sum_{k=1}^{6} N_k(a_1, a_2) \nu^{(k)} \]
Isoparametric elements: quadratic triangle

Quadratic shape functions which satisfy the property:

\[ N_k(a^{(\ell)}) = \delta_{k\ell} \]

Every shape function is associated to a node and vanishes in all the other nodes.

\[ N_1 = a_1(2a_1 - 1) \]
\[ N_2 = a_2(2a_2 - 1) \]
\[ N_3 = a_3(2a_3 - 1) \]
\[ N_4 = 4a_1a_2 \]
\[ N_5 = 4a_2a_3 \]
\[ N_6 = 4a_3a_1 \]
Examples of master and physical elements

- the unit segment \((D = 1)\):
  \[ \Delta = \{ a \mid -1 \leq a \leq 1 \} \]
- the unit square \((D = 2)\):
  \[ \Delta = \{ (a_1, a_2) \mid -1 \leq a_1, a_2 \leq 1 \} \]
- the unit triangle \((D = 2)\):
  \[ \Delta = \{ (a_1, a_2) \mid (a_1, a_2) \geq (0, 0), 1 - a_1 - a_2 \geq 0 \} \]
- the unit cube \((D = 3)\):
  \[ \Delta = \{ (a_1, a_2, a_3) \mid -1 \leq a_1, a_2, a_3 \leq 1 \} \]
- the unit tetrahedron \((D = 3)\):
  \[ \Delta = \{ (a_1, a_2, a_3) \mid (a_1, a_2, a_3) \geq (0, 0, 0), 1 - a_1 - a_2 - a_3 \geq 0 \} \]

\[ x = \sum_{k=1}^{n_e} N_k(a) x^{(k)} \quad x \in E, a \in \Delta \]

\[ \psi_h(x) = \sum_{k=1}^{n_e} N_k(a) \psi^{(k)} \]

### Master and physical elements

<table>
<thead>
<tr>
<th>Master and physical elements</th>
<th>Shape functions</th>
</tr>
</thead>
</table>
| ![Diagram of master element](image) | \( N_1 = \frac{1}{2} (1 - a) \)  
\( N_2 = \frac{1}{2} (1 + a) \) |
| ![Diagram of physical element](image) | ![Diagram of master element](image) | \( N_1 = \frac{1}{2} a (1 - a) \)  
\( N_2 = \frac{1}{2} a (1 + a) \)  
\( N_3 = 1 - a^2 \) |

Politecnico di Milano, February 3, 2017, Lesson 1
Examples of master and physical elements

\begin{align*}
N_1 &= a_1 \\
N_2 &= a_2 \\
N_3 &= 1 - a_1 - a_2 \\
N_1 &= a_1(2a_1 - 1) \\
N_2 &= a_2(2a_2 - 1) \\
N_3 &= (1 - a_1 - a_2)(1 - 2a_1 - 2a_2) \\
N_4 &= 4a_1a_2 \\
N_5 &= 4a_2(1 - a_1 - a_2) \\
N_6 &= 4a_1(1 - a_1 - a_2) \\
N_1 &= \frac{1}{4}(1 - a_1)(1 - a_2) \\
N_3 &= \frac{1}{4}(1 + a_1)(1 + a_2) \\
N_2 &= \frac{1}{4}(1 + a_1)(1 - a_2) \\
N_4 &= \frac{1}{4}(1 - a_1)(1 + a_2) \\
N_1 &= \frac{1}{4}(1 - a_1)(1 - a_2)(-1 - a_1 - a_2) \\
N_2 &= \frac{1}{4}(1 + a_1)(1 - a_2)(-1 + a_1 - a_2) \\
N_3 &= \frac{1}{4}(1 + a_1)(1 + a_2)(-1 + a_1 + a_2) \\
N_4 &= \frac{1}{4}(1 - a_1)(1 + a_2)(-1 - a_1 + a_2) \\
N_5 &= \frac{1}{2}(1 - a_1^2)(1 - a_2) \\
N_6 &= \frac{1}{2}(1 - a_2^2)(1 + a_1) \\
N_7 &= \frac{1}{2}(1 - a_1^2)(1 + a_2) \\
N_8 &= \frac{1}{2}(1 - a_2^2)(1 - a_1)
\end{align*}
Examples of master and physical elements

\[ N_1(a_1, a_2, a_3) = a_1 \]
\[ N_2(a_1, a_2, a_3) = a_2 \]
\[ N_3(a_1, a_2, a_3) = a_3 \]
\[ N_4(a_1, a_2, a_3) = 1 - a_1 - a_2 - a_3 \]

\[ N_1(a_1, a_2, a_3) = \frac{1}{8}(a_1 - 1)(a_2 - 1)(a_3 - 1) \]
\[ N_2(a_1, a_2, a_3) = \frac{1}{8}(a_1 + 1)(a_2 - 1)(a_3 - 1) \]
\[ N_3(a_1, a_2, a_3) = \frac{1}{8}(a_1 + 1)(a_2 + 1)(a_3 - 1) \]
\[ N_4(a_1, a_2, a_3) = \frac{1}{8}(a_1 - 1)(a_2 + 1)(a_3 - 1) \]
\[ N_5(a_1, a_2, a_3) = \frac{1}{8}(a_1 - 1)(a_2 - 1)(a_3 + 1) \]
\[ N_6(a_1, a_2, a_3) = \frac{1}{8}(a_1 - 1)(a_2 + 1)(a_3 + 1) \]
\[ N_7(a_1, a_2, a_3) = \frac{1}{8}(a_1 + 1)(a_2 + 1)(a_3 + 1) \]
\[ N_8(a_1, a_2, a_3) = \frac{1}{8}(a_1 + 1)(a_2 - 1)(a_3 + 1) \]
Shape functions of isoparametric elements: properties

**Exact representation of nodes (property P1).** For every type of element, the shape functions satisfy the condition that

\[ N_k(\alpha^{(\ell)}) = \delta_{k\ell} \quad (1 \leq k, \ell \leq n_e). \quad (2.14) \]

where \(\alpha^{(\ell)}\) denotes the parametric coordinates of the \(\ell\)-th master node. Thanks to this property, the representation (2.13) is guaranteed to be exact at the nodes, i.e.

\[ x^{(\ell)} = \sum_{k=1}^{n_e} N_k(\alpha^{(\ell)}) x^{(k)} \quad (1 \leq \ell \leq n_e), \quad (2.15) \]

**Representation of edges and faces (property P2).** A given shape function is associated with a node and vanishes on all the geometrical features (edges and faces) of the element which do not contain that node. Hence the geometrical representation of edges and faces only depends on the coordinates of the nodes lying on that specific feature.

**Unit sum (property P3).** For all the elements described so far, one has:

\[ \sum_{k=1}^{n_e} N_k(\alpha) = 1, \quad \forall \alpha \in \Delta \quad (2.16) \]

i.e. the shape functions always sum to unity (they are said to achieve a *partition of unity*).
Conformal meshes

Two neighbouring elements are required not to overlap, nor to make holes at common boundaries.

If the families of isoparametric elements described before are employed, this is guaranteed if two elements are:
• either be well separated
• or share one node
• or share one edge, and in this case they have the same number of nodes on the common edge with the same position

This is a fundamental benefit of isoparametric elements
Example

Important distinction between local and global numbering of nodes. E.g. the 4th local node in element 1 is the 7th global node.

For each element we select the physical (global) node that will correspond to local node number 1 (this choice is not unique since any corner node will do). The rest of the connectivity flows from this choice.

<table>
<thead>
<tr>
<th>$n_e$</th>
<th>nodes</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>6 3 9 1 7 6 2</td>
</tr>
<tr>
<td>(element 1)</td>
<td></td>
</tr>
<tr>
<td>(element 2)</td>
<td></td>
</tr>
</tbody>
</table>
List of degrees of freedom: DOF

\[ \text{nodes}(n).dof(j) > 0 \quad u_j^{(n)} \text{ free,} \]
\[ \text{nodes}(n).dof(j) = -1 \quad u_j^{(n)} \text{ prescribed.} \]

\[ \text{nodes}(:).dof(1) = \{1, 3, 5, 7, -1, 9, 11, -1, 13, 15, -1\} \]
\[ \text{nodes}(:).dof(2) = \{2, 4, 6, 8, -1, 10, 12, -1, 14, 16, -1\} \]

\[ i = \text{nodes}(n).dof(j) > 0 \text{ is the } \text{global number} \text{ of the unknown nodal value} \]

list \( \{\mathbb{V}\} \) of all the unknowns:

\[ \{\mathbb{V}\} = \{v_j^{(n)} \mid \text{nodes}(n).dof(j) > 0, \, (1 \leq n \leq N_N, \, 1 \leq j \leq D)\} \]
Galerkin interpolation

Global approximation: specific form of the Galerkin approach with:

\[ v_h(x) = \sum_{n=1}^{N_N} \tilde{N}_n(x) v^{(n)} = \sum_{n \mid x^{(n)} \in S_u} \tilde{N}_n(x) u^D(x^{(n)}) + \sum_{n \mid x^{(n)} \notin S_u} \tilde{N}_n(x) v^{(n)} \]

- \( v_h \in C_h(u^D) \): kinematically admissible in the sense of FEM approximation
Galerkin interpolation

Field **dof** associates a number to every nodal component of displacement

\[ \nu_h(x) = u_h^D(x) + \nu_h^0(x) \quad (x \in \Omega_h) \]  
\[ \text{with} \]  
\[ u_h^D(x) = \sum_{(n,j) \mid \text{nodes}(n).dof(j) \leq 0} \tilde{N}_n(x) u_j^{(n)}(x) e_j \in \mathcal{C}_h(u^D), \]  
\[ \nu_h^0 = \sum_{(n,j) \mid \text{nodes}(n).dof(j) > 0} \tilde{N}_n(x) \nu_j^{(n)} e_j \in \mathcal{C}_h(0), \]  

Specific version for Galerkin approach with:

\[ \varphi^K(x) = \tilde{N}_n(x) e_j \quad \text{and} \quad \alpha_K = u_j^{(n)} \quad \text{with} \ K = \text{nodes}(n).dof(j). \]  

Structure of **dof** to be memorised!!

- \( \text{nodes}(n).dof(j) > 0 \) \( u_j^{(n)} \) free,
- \( \text{nodes}(n).dof(j) = -1 \) \( u_j^{(n)} \) prescribed.
Global problem formulation

seek the unknown displacement field \( \mathbf{u} \) in the form

\[
\mathbf{u}_h(\mathbf{x}) = \mathbf{u}^{(D)}_h(\mathbf{x}) + \mathbf{u}^{(0)}_h(\mathbf{x}) \in \mathcal{C}_h(\mathbf{u}^D) \quad (\mathbf{x} \in \Omega_h),
\]

with

\[
\mathbf{u}^{(D)}_h(\mathbf{x}) = \sum_{(n,j) \mid \text{nodes}(n).\text{dof}(j) \leq 0} \tilde{N}_n(\mathbf{x}) \mathbf{u}^{(D)}_j(\mathbf{x}^{(n)}) e_j \in \mathcal{C}_h(\mathbf{u}^D)
\]

and

\[
\mathbf{u}^{(0)} = \sum_{(n,j) \mid \text{nodes}(n).\text{dof}(j) > 0} \tilde{N}_n(\mathbf{x}) \mathbf{u}^{(n)}_j e_j \in \mathcal{C}_h(\mathbf{0});
\]

kinematically admissible virtual fields \( \mathbf{w} \)

\[
\mathbf{w}(\mathbf{x}) = \sum_{(n,j) \mid \text{nodes}(n).\text{dof}(j) > 0} \tilde{N}_n(\mathbf{x}) \mathbf{w}^{(n)}_j e_j \in \mathcal{C}_h(\mathbf{0}) \quad (\mathbf{x} \in \Omega_h).
\]

Find \( \mathbf{u}^{(0)}_h \in \mathcal{C}_h(\mathbf{0}) \) such that: \( \forall \mathbf{w} \in \mathcal{C}_h(\mathbf{0}), \)

\[
\int_{\Omega_h} \varepsilon[\mathbf{u}^{(0)}_h] : \mathbf{A} : \varepsilon[\mathbf{w}] \, dV = - \int_{\Omega_h} \varepsilon[\mathbf{u}^{(D)}_h] : \mathbf{A} : \varepsilon[\mathbf{w}] \, dV
\]

\[
+ \int_{\Omega_h} \mathbf{f} \cdot \mathbf{w} \, dV + \int_{S_{T,h}} \mathbf{T}^D \cdot \mathbf{w} \, dS. \quad (3.3)
\]
Global problem formulation

\[ \{W\}^{T}[K]\{U\} = \sum_{e=1}^{N_E} \int_{E_e} \varepsilon[u_h^{(0)}] : \mathcal{A} : \varepsilon[w] \, dV, \]

\[ \{W\}^{T}\{F^{u}\} = \sum_{e=1}^{N_E} \left\{ -\int_{E_e} \varepsilon[u_h^{(D)}] : \mathcal{A} : \varepsilon[w] \, dV \right\} \]

\[ \{W\}^{T}\{F^{ext}\} = \sum_{e=1}^{N_E} \left\{ \int_{E_e} f \cdot w \, dV + \int_{\Gamma^e_T} T^D \cdot w \, dS \right\} \]

Find \( u_h^{(0)} \in C_h(0) \) such that: \( \forall w \in C_h(0), \)

\[ \int_{\Omega_h} \varepsilon[u_h^{(0)}] : \mathcal{A} : \varepsilon[w] \, dV = -\int_{\Omega_h} \varepsilon[u_h^{(D)}] : \mathcal{A} : \varepsilon[w] \, dV + \int_{\Omega_h} f \cdot w \, dV + \int_{S^e_{T,h}} T^D \cdot w \, dS. \quad (3.3) \]

\[ \text{find} \ \{U\} \in \mathbb{R}^N \text{ such that: } \forall \{W\} \in \mathbb{R}^N, \]

\[ \{W\}^{T}[K]\{U\} = \{W\}^{T}(\{F^{u}\} + \{F^{ext}\}) = \{W\}^{T}\{F\}, \quad (3.4) \]

linear system of unknown \( \{U\} \in \mathbb{R}^N: \quad [K]\{U\} = \{F\}. \)
Engineering notation

\[
\{v_h(x)\} = [N(a)]\{V_e\}
\]

\[
[N(a)] = \begin{bmatrix}
N_1(a) & 0 & 0 & \cdots & N_{n_e}(a) & 0 & 0 \\
0 & N_1(a) & 0 & \cdots & 0 & N_{n_e}(a) & 0 \\
0 & 0 & N_1(a) & \cdots & 0 & 0 & N_{n_e}(a)
\end{bmatrix}
\]

\{V_e\} of length \(D \times n_e\) according to the convention

\[
\{V_e\} = \{v^{(1)}, \ldots, v^{(n_e)}\}^T = \{v_1^{(1)}, v_2^{(1)}, \ldots, v_D^{(n_e)}\}^T.
\] (2.22)

\[
\{\sigma\} = \{\sigma_{11}, \sigma_{22}, \sigma_{33}, \sigma_{12}, \sigma_{13}, \sigma_{23}\}^T
\]

\[
\{\varepsilon\} = \{\varepsilon_{11}, \varepsilon_{22}, \varepsilon_{33}, 2\varepsilon_{12}, 2\varepsilon_{13}, 2\varepsilon_{23}\}^T
\]

\[
\{\sigma\} = [A]\{\varepsilon\}
\]

with \( [A] = \)

\[
\begin{bmatrix}
\lambda + 2\mu & \lambda & \lambda & 0 & 0 & 0 \\
\lambda & \lambda + 2\mu & \lambda & 0 & 0 & 0 \\
\lambda & \lambda & \lambda + 2\mu & 0 & 0 & 0 \\
0 & 0 & 0 & \mu & 0 & 0 \\
0 & 0 & 0 & 0 & \mu & 0 \\
0 & 0 & 0 & 0 & 0 & \mu
\end{bmatrix}
\]
Application: quadratic triangle T6

Let us analyse the T6 triangular element of Table 2.2 assuming that the nodal coordinates are stored in the $6 \times 2$ matrix $[X]$, such that $x_i^{(k)} = [X]_{ki}$. The components $[J]_{ij}$ of the jacobian matrix $[J]$ are

$$[J]_{ij} = \frac{\partial x_i}{\partial a_j} \quad \text{with} \quad \frac{\partial x_i}{\partial a_j} = \sum_{k=1}^{6} \frac{\partial N_k}{\partial a_j} x_i^{(k)} = \sum_{k=1}^{6} [X]_{ki} [D]_{kj}$$

where $[D]_{kj} = \frac{\partial N_k}{\partial a_j}$ are the derivatives of the shape functions with respect to the parameters. Recalling that $a_3 = 1 - a_1 - a_2$, one has

$$[D] = \begin{bmatrix} 4a_1 - 1 & 0 & -4a_3 + 1 & 4a_2 & -4a_2 & 4(a_3 - a_1) \\ 0 & 4a_2 - 1 & -4a_3 + 1 & 4a_1 & 4(a_3 - a_2) & -4a_1 \end{bmatrix}^T$$

Finally, the jacobian matrix $[J]$ reads $[J] = [X]^T [D]$ and its determinant is easily computable. In MATLAB form these formulas become

```matlab
D=[4*a1-1 0 -4*a3+1 4*a2 -4*a2 4*(a3-a1); 0 4*a2-1 -4*a3+1 4*a1 4*(a3-a2) -4*a1]';
J=X'*D;
detJ=J(1,1)*J(2,2)-J(1,2)*J(2,1);
```
A similar procedure holds also for the gradient of $v_h$

$$
\frac{\partial v_{h,i}}{\partial x_j} = \sum_{k=1}^{6} \frac{\partial N_k}{\partial x_j} v_i^{(k)} = \frac{\partial N_k}{\partial a_m} \frac{\partial a_m}{\partial x_j} v_i^{(k)} = [D]_{km} [J]_{m,j}^{-1} v_i^{(k)}
$$

which is expressed in terms of the inverse of the jacobian matrix $[J]^{-1}$. If we introduce the matrix $[G]$ the coefficients $[G]_{kj}$ of which are the derivatives of the shape functions $N_k$ with respect to $x_j$, that is:

$$
[G] = [D] [J]^{-1}
$$

one has:

$$
\frac{\partial v_{h,i}}{\partial x_j} = \sum_{k=1}^{6} [G]_{kj} v_i^{(k)}
$$

The numerical computation of $[G]$ can be easily performed in the MATLAB environment as

```
invJ=1/detJ*[J(2,2) -J(1,2); -J(2,1) J(1,1)];
G=D*invJ;
```
\[ \{ \varepsilon[u_h](x) \} = [B_e(a)]\{U_e \} \]

\[
B = \begin{bmatrix}
G(1,1) & 0 & G(2,1) & 0 & G(3,1) & 0 & G(4,1) & 0 & G(5,1) & 0 & G(6,1) & 0; \\
0 & G(1,2) & 0 & G(2,2) & 0 & G(3,2) & 0 & G(4,2) & 0 & G(5,2) & 0 & G(6,2); \\
G(1,2) & G(1,1) & G(2,2) & G(2,1) & G(3,2) & G(3,1) & \ldots \\
G(4,2) & G(4,1) & G(5,2) & G(5,1) & G(6,2) & G(6,1) & \ldots 
\end{bmatrix};
\]
Element stiffness matrix

\[ \int_{E_e} \varepsilon[u_h] : \mathbf{A} : \varepsilon[w] \, dV \]

- Mapping on master element: \( x \in E_e \rightarrow a \in \Delta_e, \quad dV = J(a) \, dV(a) \)

\[ \{\varepsilon[u_h](x)\} = [B_e(a)]\{U_e\} \]
\[ \{\varepsilon[w](x)\} = [B_e(a)]\{W_e\} \]

- Density of power of internal stresses (amphi 2)

\[ \varepsilon[u_h] : \mathbf{A} : \varepsilon[w] = \{\varepsilon[w]\}^T [A] \{\varepsilon[u_h]\} = \{W_e\}^T [B(a)]^T [A][B(a)]\{U_e\} \]

- Element stiffness matrix:

\[ \int_{E_e} \varepsilon[u_h] : \mathbf{A} : \varepsilon[w] \, dV = \{W_e\}^T \left\{ \int_{\Delta_e} [B_e(a)]^T [A][B_e(a)] \, J_e(a) \, dV(a) \right\} \{U_e\} \]

\[ \int_{E_e} \varepsilon[u_h] : \mathbf{A} : \varepsilon[w] \, dV = \{W_e\}^T [K_e] \{U_e\} \]

\[ [K_e] = \int_{\Delta_e} [B_e(a)]^T [A][B_e(a)] \, J_e(a) \, dV(a) \]

At this level no distinction is made between given and unknown displacements (see later)
Numerical integration with Gauss quadrature

(a) 1D integrals

\[ \int_{-1}^{1} f(a) \, da \approx \sum_{g=1}^{G} w_g f(a_g) \quad (a_g : \text{“Gauss points”}, \ w_g : \text{“weights”}) \]

**Formula with \( G \) Gauss points:** exact for \( f(a) \) any polynomial of degree \( \leq 2G - 1 \).

- \(-1 < a_g < 1\) (\(1 \leq g \leq G\)) for any formula with \( G \) points;
- Symmetry: if \((a_g, w_g)\) is a Gauss point, than \((-a_g, w_g)\) as well.

Example \((G = 2, \text{degree 3})\):

\[ \int_{-1}^{1} f(a) \, da \approx f\left(-\frac{1}{\sqrt{3}}\right) + f\left(\frac{1}{\sqrt{3}}\right) \]

Example \((G = 3, \text{degree 5})\):

\[ \int_{-1}^{1} f(a) \, da \approx \frac{5}{9} f\left(-\sqrt{\frac{3}{5}}\right) + \frac{8}{9} f(0) + \frac{5}{9} f\left(\sqrt{\frac{3}{5}}\right) \]
Numerical integration with Gauss quadrature

(a) 1D integrals
(b) 2D or 3D integrals over squares or cubes
Cartesian products of 1D formulas

\[
\int_{C^2} f(a_1, a_2) \, da_1 \, da_2 \approx \sum_{g_1=1}^{G} \sum_{g_2=1}^{G} w_{g_1} w_{g_2} f(a_{g_1}, a_{g_2})
\]

\[
\int_{C^3} f(a_1, a_2, a_3) \, da_1 \, da_2 \, da_3 \approx \sum_{g_1=1}^{G} \sum_{g_2=1}^{G} \sum_{g_3=1}^{G} w_{g_1} w_{g_2} w_{g_3} f(a_{g_1}, a_{g_2}, a_{g_3})
\]
Numerical integration with Gauss quadrature

(a) 1D integrals
(b) 2D or 3D integrals over squares or cubes
(c) 2D or 3D integrals over triangles or tetrahedra
Specific formulas not built from 1D formulas

\[ \int_{T^2} f(a) \, da_1 \, da_2 \approx \sum_{g=1}^{G} w_g \, f(a_g) \]

Example (triangle, Gauss-Hammer rules with G=3)

\[ \int_{T^2} f(a) \, da_1 \, da_2 \approx \frac{1}{6} \left[ f\left(\frac{1}{6}, \frac{1}{6}\right) + f\left(\frac{1}{6}, \frac{2}{3}\right) + f\left(\frac{2}{3}, \frac{1}{6}\right) \right] \]

Integrates exactly every second order polynomial
Numerical integration: bilinear quadrilateral

\[ [\mathbf{K}_e] = \sum_{g_1=1}^{G} \sum_{g_2=1}^{G} [\mathbf{B}_e(a_{g_1}, a_{g_2})]^T [\mathbf{A}] [\mathbf{B}_e(a_{g_1}, a_{g_2})] J_e(a_{g_1}, a_{g_2}) w_{g_1} w_{g_2} \]

Choice of the order of quadrature

The numerical integration is said to be **complete** if, assuming the **Jacobian matrix is constant** (non distorted element) the stiffness matrix is integrated exactly.

In our case?

G=2 is enough!

\[ N_1(a_1, a_2) = (1 - a_1)(1 - a_2)/4 \]

\[ N_2(a_1, a_2) = (1 + a_1)(1 - a_2)/4 \]

\[ N_3(a_1, a_2) = (1 + a_1)(1 + a_2)/4 \]

\[ N_4(a_1, a_2) = (1 - a_1)(1 + a_2)/4 \]
function Ke=T6_2A_solid_Ke(X,mate)

E=mate(1);
nu=mate(2);
A=E/((1+nu)*((1-2*nu))*(1-2*nu)/2);

a_gauss=1/6*[4 1 1; 1 4 1; 1 1 4]; % Gauss abscissae
w_gauss=[1/6 1/6 1/6]; % Gauss weights
Ke=zeros(12,12);
for g=1:3,
a=a_gauss(g,:);
D=[4*a(1)-1 0 -4*a(3)+1 4*a(2) ... 
   -4*a(2) 4*a(3)-a(1));
   0 4*a(2)-1 -4*a(3)+1 4*a(1) ... 
    4*(a(3)-a(2)) -4*a(1)];
J=X'*D;
detJ=J(1,1)*J(2,2)-J(1,2)*J(2,1);
invJ=1/detJ*[ J(2,2) -J(1,2); ... 
               -J(2,1)  J(1,1)];
G=D*invJ;
B=[G(1,1) 0 G(2,1) 0 G(3,1) 0 ... 
   G(4,1) 0 G(5,1) 0 G(6,1) 0; 
   0 G(1,2) 0 G(2,2) 0 G(3,2) ... 
   0 G(4,2) 0 G(5,2) 0 G(6,2); 
   G(1,2) G(1,1) G(2,2) G(2,1) ... 
   G(3,2) G(3,1) G(4,2) G(4,1) ... 
   G(5,2) G(5,1) G(6,2) G(6,1)];
Ke=Ke+B'*A*B*detJ*w_gauss(g);
end
Homeworks for next lesson

Revise chapters 1-2-3 of the book

In particular:

• isoparametric elements
• element integrals and numerical integration
• assemblage procedure
• solution
• convergence

operative knowledge of code genlin !!

play with examples and exercises provided and create your owns