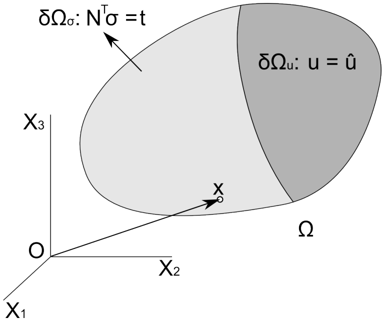
Finite Element Method applied to linear statics of deformable solids

Joel Cugnoni, LMAF / EPFL

March 1, 2017

Definitions



- Linear statics = statics of linear elastic solids in infinitesimally small deformations
- Elastic body = represented in Lagrangian form by its original, stress free, configuration Ω
- Boundary conditions:
  - Imposed displacement $\hat{u}(\mathbf{x})$ on $\delta\Omega_u$
  - Imposed surface traction $t(\mathbf{x})$ on $\delta\Omega_\sigma$

Some Maths

- Symmetric 2nd order tensors written in vector forms:

$$\sigma = \{\sigma_{11}, \sigma_{22}, \sigma_{33}, \sigma_{23}, \sigma_{31}, \sigma_{12}\}^T$$

- Differential operator:

$$\nabla = \begin{bmatrix} \partial/\partial x_1 & 0 & 0 \\ 0 & \partial/\partial x_2 & 0 \\ 0 & 0 & \partial/\partial x_3 \\ 0 & \partial/\partial x_3 & \partial/\partial x_2 \\ \partial/\partial x_3 & 0 & \partial/\partial x_1 \\ \partial/\partial x_2 & \partial/\partial x_1 & 0 \end{bmatrix}$$

hence we have:

$$div(\sigma) = \nabla^T \sigma$$

and:

$$grad(\mathbf{u}) = \nabla \mathbf{u}$$

Linear elasticity

- The stress is defined by the symmetric Cauchy stress tensor σ

$$\sigma(\mathbf{x}) = \{\sigma_{11}, \sigma_{22}, \sigma_{33}, \sigma_{23}, \sigma_{31}, \sigma_{12}\}^T$$

- For small deformations, the (infinitesimal) strain tensor $\epsilon(\mathbf{x}) = \{\epsilon_{11}, \epsilon_{22}, \epsilon_{33}, \epsilon_{23}, \epsilon_{31}, \epsilon_{12}\}^T$ is defined by:

$$\epsilon(\mathbf{x}) = \nabla \mathbf{u}(\mathbf{x})$$

- The constitutive stress - strain relationship of linear elasticity is given by the Hook's law:

$$\sigma(\mathbf{x}) = \mathbf{C}(\mathbf{x})\ \epsilon(\mathbf{x})$$

where $\mathbf{C}(\mathbf{x})$ is the 4th order elasticity tensor

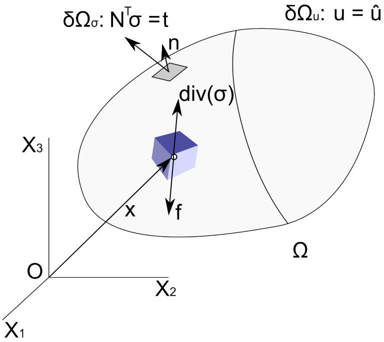
Elasticity tensor

- For an isotropic material in 3D, the 4th order symmetric elasticity tensor is defined by two constants: the Young's modulus E and the Poisson's ration ν
- With the previous definitions, **C** can be written in matrix form:

$$\mathbf{C} = \Delta \begin{bmatrix} 1-\nu & \nu & \nu & 0 & 0 & 0 \\ \nu & 1-\nu & \nu & 0 & 0 & 0 \\ \nu & \nu & 1-\nu & 0 & 0 & 0 \\ 0 & 0 & 0 & \frac{(1-2\nu)}{2} & 0 & 0 \\ 0 & 0 & 0 & 0 & \frac{(1-2\nu)}{2} & 0 \\ 0 & 0 & 0 & 0 & 0 & \frac{(1-2\nu)}{2} \end{bmatrix}$$

with: $\Delta = \frac{E}{(1+\nu)(1-2\nu)}$

Stress equilibrium



Internal Stress Equilibrium

Equilibrium is achieved if, at any point $\mathbf{x}$ of Ω:

$$div(\sigma(\mathbf{x})) + \mathbf{f}(\mathbf{x}) = 0$$

or

$$\nabla^T \sigma(\mathbf{x}) + \mathbf{f}(\mathbf{x}) = 0$$

where:

- σ is the internal Cauchy stress tensor
- $\mathbf{f}$ is the body (volumic) force vector

Differential problem of linear statics

Linear elasticity problem

Using the constitutive relationship in the equlibrium equation, we can formulate the linear static problem as:

- find the displacement field $\mathbf{u}(\mathbf{x})$ such that :

$$\nabla^T(\mathbf{C}\ \nabla\mathbf{u}) + \mathbf{f} = 0\ , \forall\mathbf{x} \in \Omega$$

- satisfying the boundary conditions:
  - imposed displacement:

$$\mathbf{u}(\mathbf{x}) = \hat{\mathbf{u}}(\mathbf{x}),\ \forall\mathbf{x} \in \delta\Omega_u$$

  - imposed surface traction:

$$\mathbf{N}^T\mathbf{C}\ \nabla\mathbf{u}(\mathbf{x}) = \mathbf{t}(\mathbf{x}), \forall\mathbf{x} \in \delta\Omega_\sigma$$

Virtual Work Principle

Virtual Work Principle

The weak form of the elastostatic problem can be obtained using the Virtual Work Principle. Taking the differential equation, we multiply it by an arbitrary admissible test function $\delta\mathbf{u}$ and integrate it over the domain Ω.

$$\int_{\Omega} \delta\mathbf{u} \left[\nabla^T \mathbf{C}\ \nabla\mathbf{u} + \mathbf{f} \right] d\Omega \ = \ 0\ ,\ \forall\ \delta\mathbf{u}$$

Note that the initial $\forall\ \mathbf{x}$ of the local equilibrium diff. equation has been replaced by $\forall\ \delta\mathbf{u}$ in the integral form.

Virtual Work Principle, development..

Applying integration by part, and imposing that $\delta \mathbf{u}$ is *continuous*, *differentiable* and satisfies the homogeneous form of the displacement boundary condition:

$$\delta \mathbf{u}(\mathbf{x}) = \mathbf{0} \ , \quad \forall \ \mathbf{x} \in \delta\Omega_u$$

We obtain:

$$-\int_{\Omega} (\nabla \delta \mathbf{u})^T \, \mathbf{C} \, \nabla \mathbf{u} \, d\Omega + \int_{\delta\Omega_\sigma} \delta \mathbf{u}^T \, \mathbf{t} \, d(\delta\Omega) + \int_{\Omega} \delta \mathbf{u}^T \, \mathbf{f} \, d\Omega \ = \ 0 \, , \forall \, \delta \mathbf{u}$$

which terms corresponds to the virtual work of *internal stresses*, *surface tractions* and *body forces*.

Comments on the function space

From the previous assumptions, the function space of the displacement field $\mathbf{u}(\mathbf{x})$ is :

$$\mathcal{U}_i = \left\{ u_i(\mathbf{x}) \mid u_i(\mathbf{x}) \in H^1(\Omega); u_i(\mathbf{x}) = \hat{u}_i(\mathbf{x}),\ \forall\ \mathbf{x} \in \delta\Omega_u \right\} \quad , i = (1,2,3)$$

And the function space of the displacement field $\delta\mathbf{u}(\mathbf{x})$ is :

$$\mathcal{V}_i = \left\{ \delta u_i(\mathbf{x}) \mid \delta u_i(\mathbf{x}) \in H^1(\Omega); \delta u_i(\mathbf{x}) = 0,\ \forall\ \mathbf{x} \in \delta\Omega_u \right\} \quad , i = (1,2,3)$$

where H^1 represents the 1st order Sobolev space of functions on Ω satisfying:

$$\int_\Omega (\nabla\mathbf{u})^T \nabla\mathbf{u} + \mathbf{u}^T\mathbf{u}\ d\Omega\ <\ \infty$$

Weak form of linear statics

Linear elasticity problem in Weak Form

find the displacement field $\mathbf{u}(\mathbf{x}) \in \mathcal{U}$ such that :

$$\int_{\Omega} (\nabla \delta \mathbf{u})^T \mathbf{C} \, \nabla \mathbf{u} \, d\Omega \; = \; \int_{\delta\Omega_\sigma} \delta \mathbf{u}^T \, \mathbf{t} \, d(\delta\Omega) + \int_{\Omega} \delta \mathbf{u}^T \, \mathbf{f} \, d\Omega \, ,$$

$$\forall \delta \mathbf{u} \; \in \; \mathcal{V}$$

where:

$$\mathcal{U}_i = \left\{ u_i(\mathbf{x}) \mid u_i(\mathbf{x}) \in H^1(\Omega); u_i(\mathbf{x}) = \hat{u}_i(\mathbf{x}), \; \forall \; \mathbf{x} \in \delta\Omega_u \right\}$$
$$\mathcal{V}_i = \left\{ \delta u_i(\mathbf{x}) \mid \delta u_i(\mathbf{x}) \in H^1(\Omega); \delta u_i(\mathbf{x}) = 0, \; \forall \; \mathbf{x} \in \delta\Omega_u \right\}$$

Comments

- The displacement boundary conditions have been moved in the class of functions $\mathcal{U}$. This type of boundary condition is called "essential" because it is related to the "essence" of the problem = the displacement field!!
- The surface traction boundary conditions $\mathbf{t}$ on $\delta\Omega_\sigma$ are now explicitly introduced in the weak form. This type of boundary condition is called "natural" because it falls "naturally" in the weak form of the problem!!
- The name "Weak Form" comes from the fact that the requirements of differentiability / continuity of the fields $\mathbf{u}$ and $\delta\mathbf{u}$ are relaxed compared to the initial differential equation form. Note that the functions do not necessarilly need to be continuous!

Approximate solution of the Weak Form

We are looking for an approximate solution of the Weak Form problem. We give us two subsets $\mathcal{U}^h$ and $\mathcal{V}^h$ of the function spaces $\mathcal{U}$ and $\mathcal{V}$ to form the basis of the approximations:

$$\mathbf{u}^h \approx \mathbf{u}\ ,\ \delta\mathbf{u}^h \approx \delta\mathbf{u}$$

$$\mathbf{u}^h \in \mathcal{U}^h \subset \mathcal{U}\ ,\ \delta\mathbf{u}^h \in \mathcal{V}^h \subset \mathcal{V}$$

By considering an approximation $\mathcal{U}^h$ based on a linear combination of a finite number of basis functions, we can write:

$$\mathbf{u}^h(\mathbf{x}) = \mathbf{H}(\mathbf{x})\,\mathbf{q}$$

where $\mathbf{H}(\mathbf{x})$ is the 3xn matrix of shape functions and $\mathbf{q}$ is the vector of displacement components. Using Galerkin method, we choose the same base function space for $\delta\mathbf{u}$:

$$\delta\mathbf{u}^h(\mathbf{x}) = \mathbf{H}(\mathbf{x})\,\delta\mathbf{q}$$

Approximation of the Weak Form

Using the approximations $\mathbf{u}^h = \mathbf{H}\,\mathbf{q}$ and $\delta\mathbf{u}^h = \mathbf{H}\,\delta\mathbf{q}$, the weak form can be rewritten as:

$$\delta\mathbf{q}^T \left\{ \int_\Omega (\nabla\mathbf{H})^T\,\mathbf{C}\,\nabla\mathbf{H}\,\mathbf{q}\,d\Omega - \int_\Omega \mathbf{H}^T\mathbf{f}\,d\Omega - \int_{\delta\Omega_\sigma} \mathbf{H}^T\mathbf{t}\,d(\delta\Omega) \right\} = 0$$

This relation must hold $\forall\delta\mathbf{u}^h \in \mathcal{V}^h$ and thus $\forall\delta\mathbf{q}$, thus:

$$\int_\Omega \mathbf{B}^T\mathbf{C}\mathbf{B}\,\mathbf{q}\,d\Omega = \int_\Omega \mathbf{H}^T\mathbf{f}\,d\Omega + \int_{\delta\Omega_\sigma} \mathbf{H}^T\mathbf{t}\,d(\delta\Omega)$$

where $\mathbf{B} = \nabla\mathbf{H}$ is called the displacement-strain interpolation matrix because $\epsilon = \nabla(\mathbf{H}\mathbf{q}) = \nabla\mathbf{H}\,\mathbf{q} = \mathbf{B}\mathbf{q}$

Approximate linear statics problem

Approximate linear statics problem

find the displacement components **q** such that :

$$\mathbf{K}\,\mathbf{q} \;=\; \mathbf{r}$$

and:

$$\mathbf{u}^h(\mathbf{x}) = \mathbf{H}(\mathbf{x})\,\mathbf{q} = \hat{\mathbf{u}}\;,\;\; \forall \mathbf{x} \in \delta\Omega_u$$

with:

$$\mathbf{K} = \int_\Omega \mathbf{B}^T \mathbf{C} \mathbf{B}\; d\Omega$$

$$\mathbf{r} = \int_\Omega \mathbf{H}^T \mathbf{f}\, d\Omega + \int_{\delta\Omega_\sigma} \mathbf{H}^T \mathbf{t}\, d(\delta\Omega)$$

Comments

- $\mathbf{K}$ is called the stiffness matrix of the system, $\mathbf{r}$ is the force vector containing contributions of both surface tractions and body forces
- By choosing a finite subset of basis functions to build the approximation of the weak, we have turned the algebraic differential equation problem into a linear system of equations: all terms of $\mathbf{K}$ and $\mathbf{r}$ are constant, hence the name "linear static problem".
- As the displacement is a linear combination of functions defined over Ω, enforcing exact displacement boundary conditions $\mathbf{u}^h(\mathbf{x}) = \hat{\mathbf{u}}$ is not trivial except for zero imposed displacement.
- Integrating the matrices & force vector over an arbitrary domain can become very complex, and thus numerical integration methods may be used.

In some sense, the Finite Element Method is one way of defining a *systematic approximate function space* with the aim to:

1. build approximate solutions over *arbitrarily complex domains*
2. simplify the treatment of *essential boundary conditions*
3. simplify the *integration* of the global system matrice(s) and load vector(s)
4. automate the resolution procedure: integration of the linear system & solution

Now let's see how this can be achieved !!

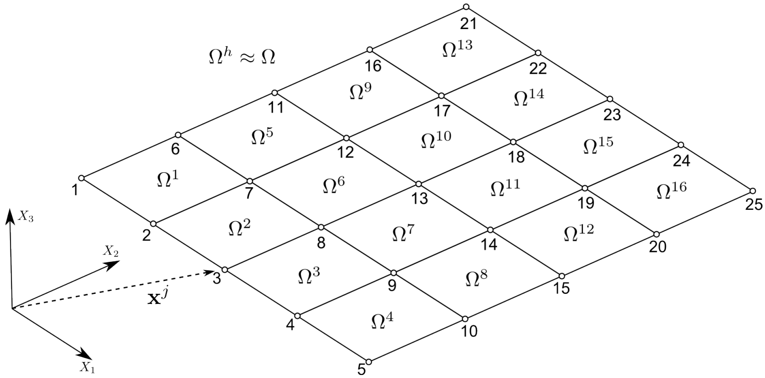
How to process complex domain?

By *discretizing* the domain in a set of smaller & simpler domains. The discretization Ω^h of Ω is defined by a set of discrete points called *nodes* and a set of sub-domains of basic topology called *elements*, which forms what we call a *mesh*.

$$\Omega^h \approx \Omega \quad , \quad \Omega^h = \Omega^1 \cup \Omega^2 \cup ... \cup \Omega^n$$

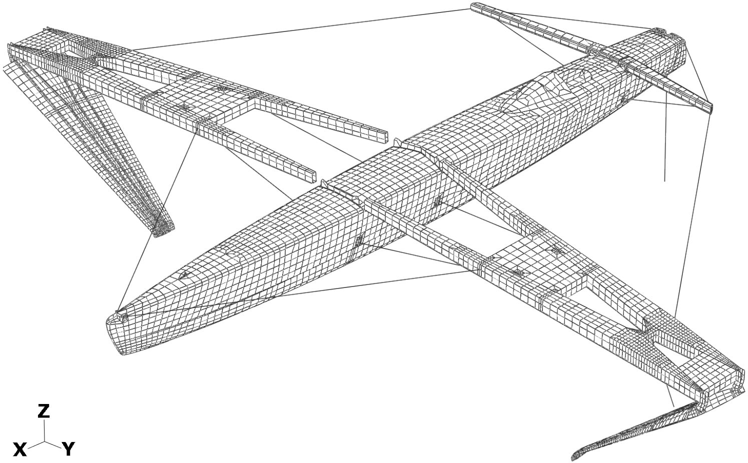
The j-th node is represented by its coordinates $\mathbf{x}^j$ and the each element is defined by its connectivities: an ordered list of nodes which serve as support for the element.

The key ideas behind the Finite Element Method



A simple mesh made of 16 elements and 25 nodes

The key ideas behind the Finite Element Method



An much more complex mesh of an Hydroptere

How to simplify the treatment of essential BC?

Use *discrete node-based shape functions* on Ω^h : each node of coordinates $\mathbf{x}^i$ is associated to its own shape function h_i. The approximation $\mathbf{u}^h$ is then written as :

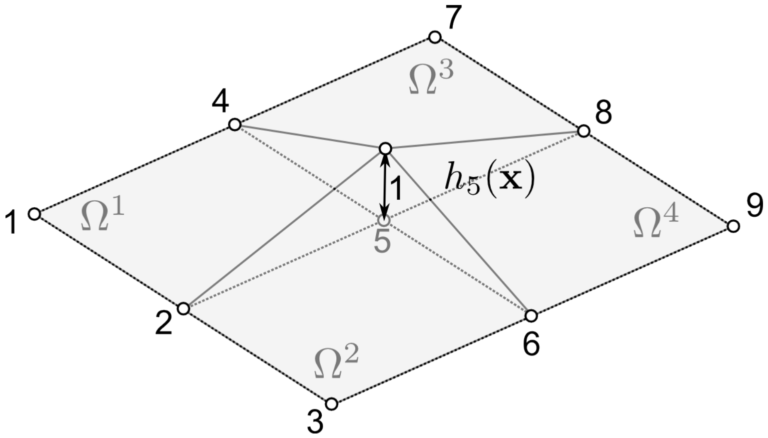
$$\mathbf{u}^h = \mathbf{H}\,\mathbf{q} \quad \textit{with} \quad \mathbf{H} = [h_1\,\mathbf{I}, h_2\,\mathbf{I}, ..., h_n\,\mathbf{I}]$$

$$\textit{and} \quad \mathbf{q} = \left\{\mathbf{q}^1, \mathbf{q}^2, ..., \mathbf{q}^p\right\}^T$$

Moreover, the shape functions are required to have *compact support* (Kronecker property):

$$\mathbf{h}_i(\mathbf{x}^j) = \delta_{ij} \quad \forall \mathbf{x}^j \in \Omega^h$$

The key ideas behind the Finite Element Method



An illustration of $h_5(\mathbf{x})$ showing the compact support property

How to simplify the treatment of essential BC?

Thanks to the Kronecker property, the approximate displacement $\mathbf{u}^h$ on nodes $\mathbf{x}^j$ is then:

$$\mathbf{u}^h(\mathbf{x}^j) = \sum_{i=1}^{p} h_i(\mathbf{x}^j)\,\mathbf{q}^i = \mathbf{q}^j$$

In other words, the vector $\mathbf{q}$ has now the *physical meaning*: it represents the *unknown displacements* at nodes!
Thus the essential boundary conditions can be simply rewritten as:

$$\mathbf{u}(\mathbf{x}^j) = \hat{\mathbf{u}}(\mathbf{x}^j) \quad \forall\ \mathbf{x}^j \in \delta\Omega_u \Longrightarrow \mathbf{q}^j = \hat{\mathbf{u}}(\mathbf{x}^j) = \hat{\mathbf{q}}^j \quad \forall\ j \mid \mathbf{x}^j \in \delta\Omega_u$$

How to simplify the integration?

Split integration in each element..

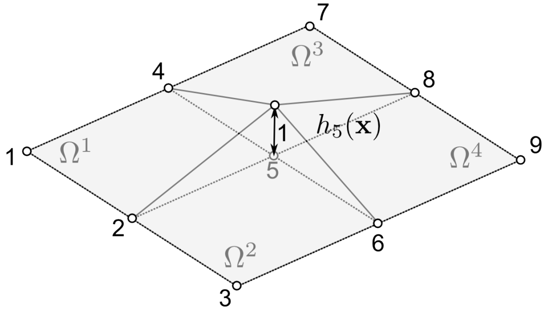
$$\int_{\Omega} d\Omega = \sum_{i=1}^{n} \int_{\Omega^i} d\Omega$$

$$\int_{\delta\Omega} d(\delta\Omega) = \sum_{i=1}^{k} \int_{\delta\Omega^i} d(\delta\Omega)$$

Note that local support means that, when integrating h_i, only contributions of element containing node i must be processed.

How to simplify the integration?

$$\int_{\Omega} \; d\Omega = \int_{\Omega^1} \; d\Omega + \int_{\Omega^2} \; d\Omega + \int_{\Omega^3} \; d\Omega + \int_{\Omega^4} \; d\Omega$$



How to automate the process?

By treating the problem "per element" : *localization* !
Local displacement approximation in the e-th element as :

$${}^{e}\mathbf{u}^{h} = {}^{e}\mathbf{H}\ {}^{e}\mathbf{q}$$

Where:

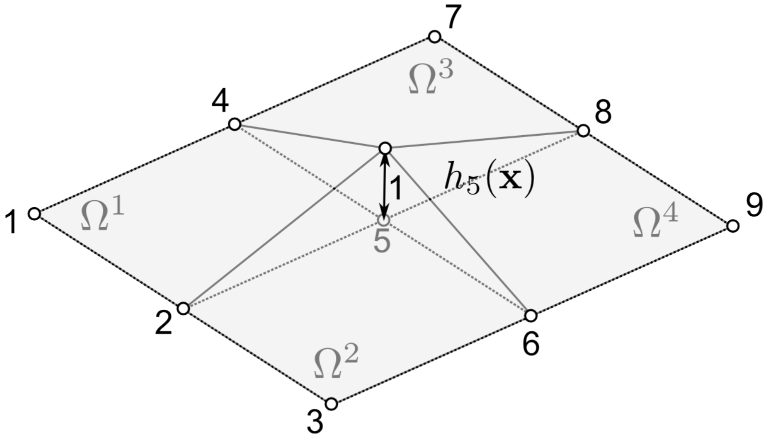
$${}^{e}\mathbf{H} = [{}^{e}h_1\mathbf{I},\ {}^{e}h_2\mathbf{I}, ...,\ {}^{e}h_p\mathbf{I}]$$

And:

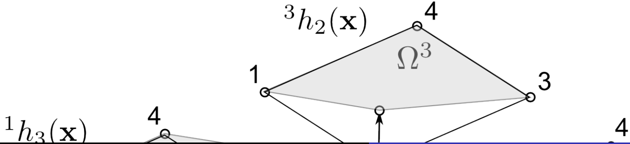
$${}^{e}\mathbf{q} = {}^{e}\mathbf{L}\ \mathbf{q}$$

${}^{e}\mathbf{L}$ is called the binary localization matrix representing the global - local numbering relationship (related to the element connectivity table)

The key ideas behind the Finite Element Method



An illustration of the global shape function $h_5(\mathbf{x})$



Connectivity table : Local - Global node numbering

${}^e\Omega$	Ω^1	Ω^2	Ω^3	Ω^4
1	1	2	4	5
2	2	3	5	6
3	5	6	8	9
4	4	5	7	8

Note:

- The connectivity list of each element correspond to a column of the above table.
- The localization operator ${}^e\mathbf{L}$ is constructed from the connectivity table.

Integration of the system of equation

Using the localized quantities ${}^e\mathbf{u}^h$ and ${}^e\mathbf{H}$, we can expand the integration:

$$\mathbf{K} = \int_{\Omega^h} \mathbf{B}^T \ \mathbf{C} \ \mathbf{B} \ d\Omega = \sum_{e=1}^{q} \ {}^e\mathbf{L}^T \left(\int_{\Omega^e} \ {}^e\mathbf{B}^T \ {}^e\mathbf{C} \ {}^e\mathbf{B} \ d\Omega \right) \ {}^e\mathbf{L}$$

where ${}^e\mathbf{B} = \nabla \ {}^e\mathbf{H}$ is the local displacement - strain matrix and ${}^e\mathbf{C}$ is the element elasticity matrix (strain - stress relationship).

We then define the **elementary stiffness matrix** ${}^e\mathbf{K}$:

$${}^e\mathbf{K} = \int_{\Omega^e} \ {}^e\mathbf{B}^T \ {}^e\mathbf{C} \ {}^e\mathbf{B} \ d\Omega$$

Assembly operation

We define the so-called **Assembly operator** $\biguplus$ as:

$$\biguplus_{e=1}^{p} \{\cdot\} = \sum_{e=1}^{p} {}^{e}\mathbf{L}^{T} \{\cdot\}^{e} \mathbf{L}$$

With this operator:

$$\mathbf{K} = \sum_{e=1}^{p} {}^{e}\mathbf{L}^{T\,e}\mathbf{K}^{e}\mathbf{L} = \biguplus_{e=1}^{p} {}^{e}\mathbf{K}$$

Automating the integration / standardizing the shape functions

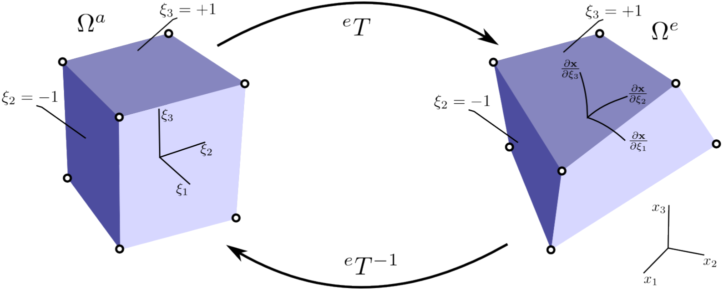
To automate the integration and simplify the definition of shape functions, we *transform* each "distorted" elementary domain Ω^e into a reference domain Ω^a where we can apply *standard numerical integration* procedures.

The coordinate transformation

$$^{e}T \; : \Omega^a \;\rightarrow\; \Omega^e \mid \xi \rightarrow \mathbf{x}(\xi)$$

maps any point of coordinate $\xi = \{\xi_1, \xi_2, \xi_3\}^T$ in the master domain Ω^a to its (single) corresponding point of coordinate $\mathbf{x} = \mathbf{x}(\xi)$ in the elementary domain Ω^e (bijective application).

The key ideas behind the Finite Element Method



An illustration of the coordinate transform ${}^eT$

Master element & shape functions

With the coordinate transform ${}^{e}T$ and ${}^{e}T^{-1}$, the shape functions ${}^{e}h_i(\mathbf{x})$ can be mapped to master shape functions ${}^{a}h_i(\xi)$ defined in the master element space Ω^a:

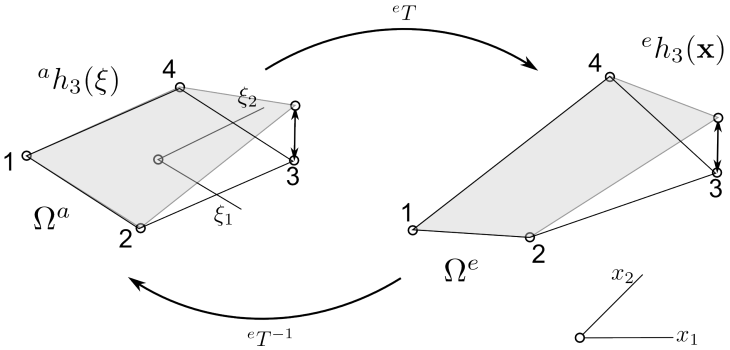
$$ {}^{e}\mathbf{H}(\mathbf{x}) = {}^{e}\mathbf{H}(\mathbf{x}(\xi)) = {}^{a}\mathbf{H}(\xi) $$

Such that now, in each element e:

$$ {}^{e}\mathbf{u}^{h}[\mathbf{x}(\xi)] \; = \; {}^{a}\mathbf{H}(\xi) \, {}^{e}\mathbf{q} $$

Thanks to that transformation, we only need to derive once the shape function formulations ${}^{a}\mathbf{H}(\xi)$ in the normalized "master" element space $\Omega^a \Rightarrow$ Standardized shape functions !!

The key ideas behind the Finite Element Method



Element & Master shape functions

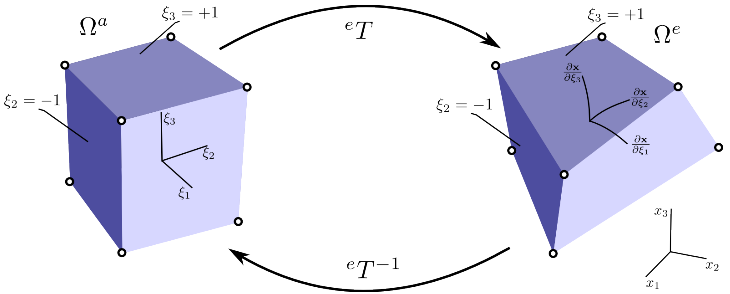
How to choose a simple form of coordinate transformation ?

Simply by using the shape functions to write the coordinate transform:

$$ {}^{e}T \ : \mathbf{x} = \mathbf{x}(\xi) = {}^{a}\mathbf{H}(\xi)\ {}^{e}\mathbf{x} $$

Which means that inside each element e, the coordinates are interpolated as a linear combination of the shape functions and nodal coordinates ${}^{e}\mathbf{x}$.
With the Kronecker property ${}^{e}h_i(x^j) = \delta_{ij} \Leftrightarrow {}^{a}h_i(\xi^j) = \delta_{ij}$, this ensures that each node of the master element corresponds to a node in the "deformed" element e.

The key ideas behind the Finite Element Method



Coordinate transformation

Automating the integration

Using the coordinate transform ${}^{e}T$ and master shape functions, the integrals can be rewritten to be carried out directly on a standard domain Ω^a:

$$ {}^{e}\mathbf{K} = \int_{\Omega^a} \left(\{ {}^{e}\mathbf{B}(\mathbf{x}(\xi)) \}^T \; \mathbf{C} \; \{ {}^{e}\mathbf{B}(\mathbf{x}(\xi)) \} \right) \; {}^{e}j \; d\Omega^a $$

Where ${}^{e}j$ is the determinant of the jacobian matrix ${}^{e}\mathbf{J}$

$$ {}^{e}j \; = \; det(\, {}^{e}\mathbf{J}) $$

with the definition:

$$ {}^{e}\mathbf{J}_{ij} = \frac{\partial x_i}{\partial \xi_j} $$

Master element & derivatives

The spatial derivative operator ∇ is defined in the global coordinate system x_1, x_2, x_3. Applying the coordinate transform ${}^{e}T$, we can then extend it to be applied on the master element Ω^a. Using the standard derivation rules:

$$\partial/\partial x_i \;=\; \partial/\partial\xi_j \, \partial\xi_j/\partial x_i$$

or in matrix - vector form:

$$\partial/\partial\mathbf{x} \;=\; {}^{e}\mathbf{J}^{-1} \, \partial/\partial\xi$$

With this definition, the elementary strain-displacement matrix ${}^{e}\mathbf{B}$ can be directly derived from the master shape functions ${}^{a}\mathbf{H}$ and the integration is completely carried out in the master domain Ω^a:

$${}^{e}\mathbf{B} \;=\; \nabla\, {}^{a}\mathbf{H}(\xi)$$

Numerical integration

On the master element, standard Gauss-Legendre numerical integration schemes can be used to automate the calculation of the elementary stiffness matrix ${}^e\mathbf{K}$ and load vector. For each dimension, the integrals are transformed into a weighted sum:

$$\int_{-1}^{+1} (\cdot)\ d\xi_j \ \approx \ \sum_{i=1}^{r} \ \omega_j^i(\cdot)\ |_{\xi_j=\xi_j^i}$$

where the points of coordinates ξ_j^i are called "Gauss points" or integration points and ω_j^i are the associated weights.
Using the integration schemes in the three directions:

$$\int_{-1}^{+1}\int_{-1}^{+1}\int_{-1}^{+1} (\cdot)\ d\xi_3 d\xi_2 d\xi_1 \ \approx \ \sum_{i=1}^{r}\sum_{j=1}^{s}\sum_{k=1}^{t} \ \omega_1^i\omega_2^j\omega_3^k(\cdot)\ |_{\xi_1=\xi_1^i,\xi_2=\xi_2^j,\xi_3=\xi_3^k}$$

$${}^e\mathbf{K} = \int_{\Omega^a} \left(\{\ {}^e\mathbf{B}(\mathbf{x}(\xi))\}^T\ \ \mathbf{C}\ \ \{\ {}^e\mathbf{B}(\mathbf{x}(\xi))\}\right)\ {}^e j\ d\Omega^a$$

Numerical integration

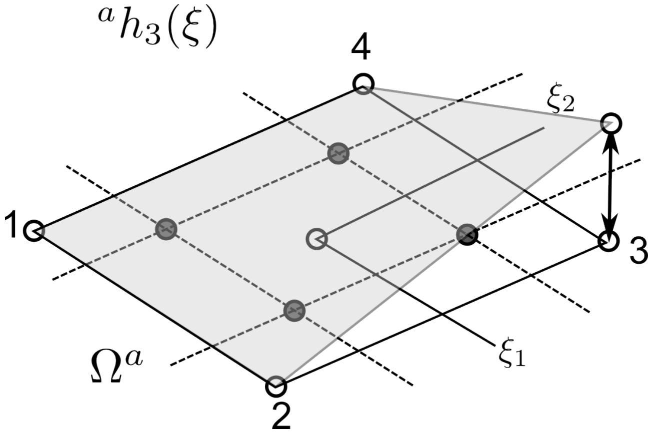
Using the integration schemes in the three directions:

$$\int_{-1}^{+1}\int_{-1}^{+1}\int_{-1}^{+1}(\cdot)\,d\xi_3 d\xi_2 d\xi_1 \approx \sum_{i=1}^{r}\sum_{j=1}^{s}\sum_{k=1}^{t}\omega_1^i\omega_2^j\omega_3^k(\cdot)\,|_{\xi_1=\xi_1^i,\xi_2=\xi_2^j,\xi_3=\xi_3^k}$$

Thus, the elementary stiffness matrix ${}^e\mathbf{K}$ can be computed numerically as:

$${}^e\mathbf{K} \approx \sum_{i=1}^{r}\sum_{j=1}^{s}\sum_{k=1}^{t}\omega_1^i\omega_2^j\omega_3^k\left[\,(\nabla^a\mathbf{H})^T\ \mathbf{C}\ \ (\nabla^a\mathbf{H})\ {}^e j\right]_{\xi_1=\xi_1^i,\xi_2=\xi_2^j,\xi_3=\xi_3^k}$$

Note that the internal terms $(\nabla^a\mathbf{H})^T\ \mathbf{C}\ (\nabla^a\mathbf{H})\ j$ are evaluated directly in the master element coordinate system ξ_1, ξ_2, ξ_3.



Gauss points & numerical integration

So finally, what is a Finite Element ?

Using these principles, a standard, displacement based, Finite Element formulation is defined by:

1. The basic topology of its "master element" Ω^a
2. The coordinate transform ${}^eT$ mapping the master domain Ω^a to the real element domain Ω^e
3. The shape functions ${}^a\mathbf{H}$ and their derivatives $\partial h_i/\partial \xi_j$ in the master element
4. The numerical integration scheme (Gauss points and weights) in the master element.

So finally, what is a Finite Element Model?

A Finite Element Model is thus defined by :

1. The geometrical mesh Ω^h consisting of all elementary domains Ω^e (defined by nodes and connectivity table)
2. The definition of the problem to be solved: $\mathbf{K}\ \mathbf{q} = \mathbf{r}$
3. A list of Finite Elements (formulations) supported by geometrical cells of the mesh
4. Boundary conditions: imposed nodal displacement $\hat{\mathbf{q}}$ on $\delta\Omega_u$ and imposed surface tractions on $\delta\Omega_\sigma$
5. Material properties (${}^e\mathbf{C}$) assigned to each Finite Element

Shape functions

Requirements

1. **Continuity**: to have a continuous solution, shape functions must be *continuous along the interfaces* between elements
2. **Completeness**: to be able to represent *uniform strain & displacement*, shape functions must at least be *affine functions*
3. **Differentiability**: shape functions & derivatives must be sufficiently regular to be *square integrable* over each element

Consequences

- **Partition of unity**: at each point $\mathbf{x}$, $\sum_i h^i(\mathbf{x}) = 1$
- Basic choice of shape function: **piecewise polynomials**, at least piecewise trilinear functions in 3D