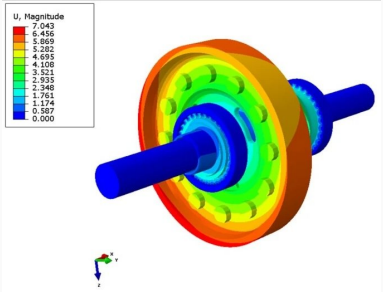
Solid elements and numerical integration

Linear elastodynamics

ME473 Dynamic finite element analysis of structures

Stefano Burzio

2025



Where do we stand?

Week	Module	Lecture topic	Mini-projects
1	Linear elastodynamics	Strong and weak forms	
2		Galerkin method	
3		Finite element method	Groups formation
4		Systematization of the procedure	Project 1 statement
5		3d elements, numerical integration	

Summary

- Recap week 4
- Solid finite elements
- Numerical integration
- MATLAB PDE Toolbox examples
- Abaqus example

Recommended readings

1. Gmür, Dynamique des structures (§3.1, §3.2, and §3.3)

Recap week 4

Localization and assembly

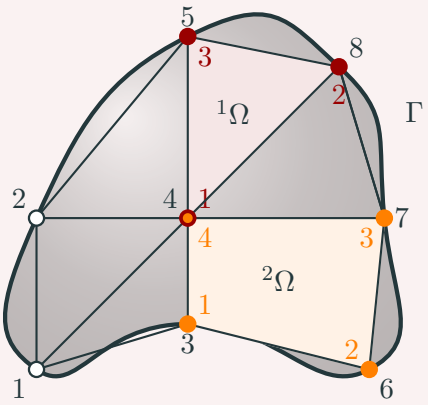
- **Localization:** real and virtual displacements are restricted to each finite element and are expressed in terms of local shape functions and nodal displacements:

$$ {}^e\mathbf{u}^h(\mathbf{x},t) = {}^e\mathbf{H}(\mathbf{x}){}^e\mathbf{q}(t) \quad \text{and} \quad {}^e\boldsymbol{\delta}\mathbf{u}^h(\mathbf{x}) = {}^e\mathbf{H}(\mathbf{x}){}^e\boldsymbol{\delta}\mathbf{q}. $$

- **Localization matrices ${}^e\mathbf{L}$:** boolean matrices that map local to global node numbering.
- **Elementary quantities:** elementary stiffness matrix ${}^e\mathbf{K}$, elementary mass matrix ${}^e\mathbf{M}$, and elementary force vector ${}^e\mathbf{r}$ are computed for each element.
- **Assembly:** global matrices and vectors are assembled using the localization matrices:

$$ \mathbf{K} = \mathop{\mathsf{A}}_{e=1}^{m} {}^e\mathbf{K} = \sum_{e=1}^{m} {}^e\mathbf{L}^T {}^e\mathbf{K} {}^e\mathbf{L} \quad \text{and} \quad \mathbf{r} = \mathop{\mathsf{A}}_{e=1}^{m} {}^e\mathbf{r} = \sum_{e=1}^{m} {}^e\mathbf{L}^T {}^e\mathbf{r} $$

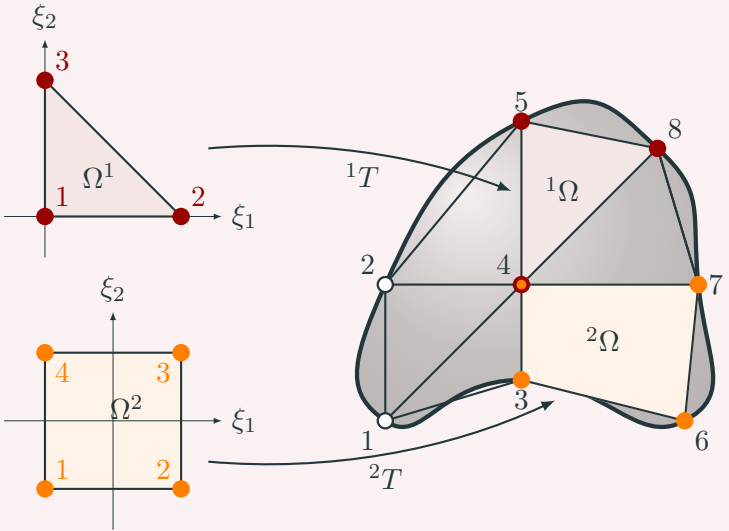
The local finite element point of view (in 2d)



Every finite element ${}^e\Omega$ has:

- Number of nodes ${}^e p$,
- Connectivity $({}^e p \times 1)$,
- ${}^e\mathbf{q}$ vector of nodal displacements $(2\,{}^e p \times 1)$,
- ${}^e\mathbf{u}$ vector of displacements (2×1),
- ${}^e h_i$ local shape functions $(i = 1, \ldots, {}^e p)$,
- ${}^e\mathbf{H}$ local shape functions matrix $(2 \times 2\,{}^e p)$,
- ${}^e\mathbf{K}$ local stiffness matrix $(2\,{}^e p \times 2\,{}^e p)$,
- ${}^e\mathbf{M}$ local mass matrices $(2\,{}^e p \times 2\,{}^e p)$,
- ${}^e\mathbf{r}$ local loads vector $(2\,{}^e p \times 1)$.

Archetypal to local



Paradigm shift: many with few

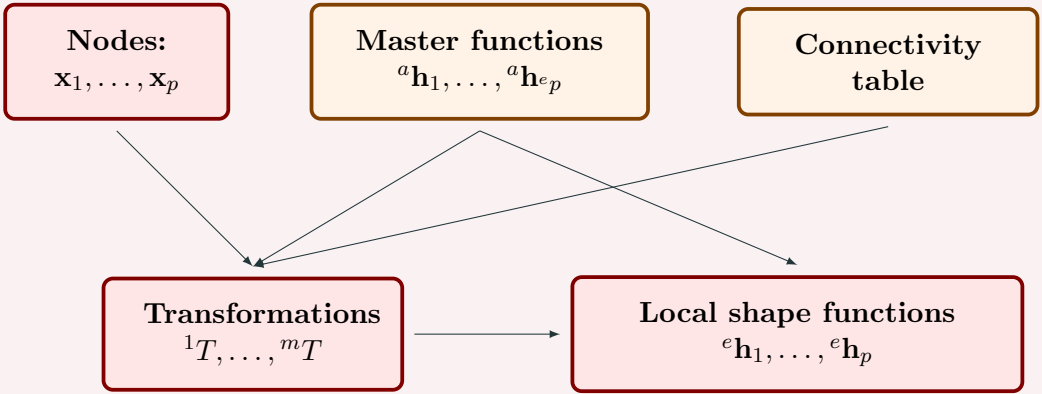
☞ **Main idea:** instead of working with local shape functions ${}^{e}h_i$ (one per node!) we will deduce them from a **small set of archetypal (or master) shape functions** ${}^{a}h_i$ defined on an archetypal element via a transformation ${}^{e}T$.

☞ **Clever idea:** the transformation ${}^{e}T$ itself is defined in terms of archetypal shape functions and nodes coordinates:

$$ {}^{e}T : \mathbf{x}(\boldsymbol{\xi}) = {}^{a}\mathbf{H}(\boldsymbol{\xi}) {}^{e}\mathbf{x} = \sum_{i=1}^{{}^{e}p} {}^{a}h_i(\boldsymbol{\xi}) {}^{e}\mathbf{x}_i $$

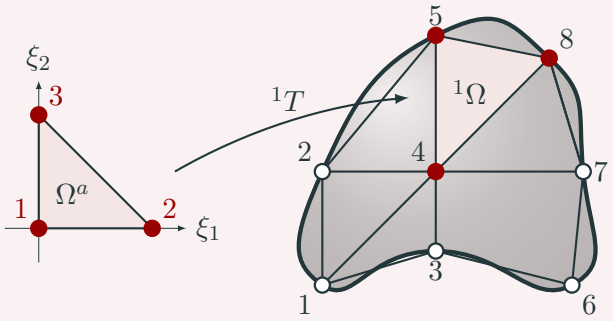
☞ Archetypal elements follows *established conventions* (node numbering and nodes coordinates) that one needs to adopt.

The clever idea



Concrete example: bilinear triangular element (2d)

Node id local	Local Coord	Node id global	Global coord.
1	$(0,0)$	4	(x_4, y_4)
2	$(1,0)$	8	(x_8, y_8)
3	$(0,1)$	5	(x_5, y_5)



Base (archetypal) functions:

$$^a h_1 = 1 - \xi_1 - \xi_2$$
$$^a h_2 = \xi_1$$
$$^a h_3 = \xi_2$$

Transformation:

$$^1T : \begin{cases} x(\xi_1, \xi_2) = x_4 h_1 + x_8 h_2 + x_5 h_3 = x_4 + (x_8 - x_4)\xi_1 + (x_5 - x_4)\xi_2 \\ y(\xi_1, \xi_2) = y_4 h_1 + y_8 h_2 + y_5 h_3 = y_4 + (y_8 - y_4)\xi_1 + (y_5 - y_4)\xi_2 \end{cases}$$

Consequence 1: explicit use of local shape functions is avoided

- The coordinate transformation ${}^eT^{-1}$ maps shape functions on ${}^e\Omega$ to the master element space Ω^a:

$$ {}^e\mathbf{H}(\mathbf{x}) = {}^a\mathbf{H}({}^eT^{-1}(\mathbf{x})) $$

- An example of a relationship between local and archetypal shape function:

$$ {}^1h_8(x,y) = {}^ah_2\big(\xi_1(x,y),\xi_2(x,y)\big) $$

☞ In general, the inverse transformations ${}^eT^{-1} : \big(\xi_1(x,y),\xi_2(x,y)\big)$ are not easy to compute since the expressions in $\mathbf{x}(\boldsymbol{\xi})$ are nonlinear.

☞ Although the local functions ${}^eh_j$ can have intricate expressions, their explicit computation is not required !

Consequence 2: integration over archetypal elements

Thanks to ${}^eT$, the integrals over ${}^e\Omega$ in the definitions of elementary stiffness and mass matrices and elementary loads vectors can be computed over the regular element Ω^a by performing a change of variables:

$$\int_{{}^e\Omega}(\dots)d\Omega \quad \overset{\text{replaced by}}{\longmapsto} \quad \int_{\Omega^a}(\dots)\,{}^ej\,d\xi_1 d\xi_2$$

$$\frac{\partial}{\partial x_i} \quad \overset{\text{replaced by}}{\longmapsto} \quad \frac{\partial \xi_1}{\partial x_i}\partial_{\xi_1} + \frac{\partial \xi_2}{\partial x_i}\partial_{\xi_2}$$

where ${}^ej$ is the determinant of the Jacobian matrix ${}^e\mathbf{J}$ of the transformation ${}^eT$.

☞ For each transformation, it is necessary to compute the Jacobian matrix, its determinant, and its inverse.

Concrete example: bilinear triangular element (2d)

- **Transformation:**

$$ {}^1T : \begin{cases} x(\xi_1, \xi_2) = x_4 h_1 + x_8 h_2 + x_5 h_3 = x_4 + (x_8 - x_4)\xi_1 + (x_5 - x_4)\xi_2 \\ y(\xi_1, \xi_2) = y_4 h_1 + y_8 h_2 + y_5 h_3 = y_4 + (y_8 - y_4)\xi_1 + (y_5 - y_4)\xi_2 \end{cases} $$

- **Jacobian matrix:**

$$ {}^1\mathbf{J} = \begin{bmatrix} \partial x/\partial \xi_1 & \partial x/\partial \xi_2 \\ \partial y/\partial \xi_1 & \partial y/\partial \xi_2 \end{bmatrix} = \begin{bmatrix} x_8 - x_4 & x_5 - x_4 \\ y_8 - y_4 & y_5 - y_4 \end{bmatrix} $$

- **Jacobian determinant:**

$$ {}^1j = \det {}^1\mathbf{J} = (x_8 - x_4)(y_5 - y_4) - (x_5 - x_4)(y_8 - y_4) $$

- **Jacobian inverse matrix:**

$$ {}^1\mathbf{J}^{-1} = \frac{1}{{}^1j} \begin{bmatrix} x_8 - x_4 & -x_5 + x_4 \\ -y_8 + y_4 & y_5 - y_4 \end{bmatrix} $$

Concrete example: bilinear triangular element (2d)

- **Local mass matrix:**

$$
\begin{aligned}
{}^1\mathbf{M} &= \int_{\Omega^a} \rho\, {}^a\mathbf{H}^T {}^a\mathbf{H}\, {}^1j\, d\boldsymbol{\xi} \\
&= \int_{\Omega^a} \rho \begin{bmatrix} {}^ah_1^2\mathbf{I} & {}^ah_1{}^ah_2\mathbf{I} & {}^ah_1{}^ah_3\mathbf{I} \\ {}^ah_2{}^ah_1\mathbf{I} & {}^ah_2^2\mathbf{I} & {}^ah_2{}^ah_3\mathbf{I} \\ {}^ah_3{}^ah_1\mathbf{I} & {}^ah_3{}^ah_2\mathbf{I} & {}^ah_3^2\mathbf{I} \end{bmatrix} {}^1j\, d\xi_2 d\xi_1
\end{aligned}
$$

- **Local stiffness matrix:**

$$
{}^1\mathbf{K} = \int_{\Omega^a} \begin{bmatrix} \cdots & & \cdots \\ & \left(\nabla_{\boldsymbol{\xi}}\, {}^ah_i\, {}^1\mathbf{J}^{-1}\right)^T \mathbf{C} \left(\nabla_{\boldsymbol{\xi}}\, {}^ah_j\, {}^1\mathbf{J}^{-1}\right) & \\ \cdots & & \cdots \end{bmatrix} {}^1j\, d\xi_2 d\xi_1
$$

Assembly of the mass matrix: local to global

$$ {}^1\mathbf{M} = \int_{\Omega^a} \rho \begin{bmatrix} {}^a h_1^2 \mathbf{I} & {}^a h_1 {}^a h_2 \mathbf{I} & {}^a h_1 {}^a h_3 \mathbf{I} \\ {}^a h_2 {}^a h_1 \mathbf{I} & {}^a h_2^2 \mathbf{I} & {}^a h_2 {}^a h_3 \mathbf{I} \\ {}^a h_3 {}^a h_1 \mathbf{I} & {}^a h_3 {}^a h_2 \mathbf{I} & {}^a h_3^2 \mathbf{I} \end{bmatrix} {}^e j \, d\xi_2 d\xi_1 = \begin{bmatrix} {}^1\mathbf{M}_{11} & {}^1\mathbf{M}_{12} & {}^1\mathbf{M}_{13} \\ {}^1\mathbf{M}_{21} & {}^1\mathbf{M}_{22} & {}^1\mathbf{M}_{23} \\ {}^1\mathbf{M}_{31} & {}^1\mathbf{M}_{32} & {}^1\mathbf{M}_{33} \end{bmatrix} $$

We assembly local mass matrices into the global ones using the assembly operator:

$$ \mathbf{M} = \mathop{\mathbf{A}}_{e=1}^{m} {}^e\mathbf{M} = \sum_{e=1}^{m} {}^e\mathbf{L}^T {}^e\mathbf{M} {}^e\mathbf{L} $$

where

$$ {}^1\mathbf{L} = \begin{bmatrix} \mathbf{0} & \mathbf{0} & \mathbf{0} & \mathbf{I} & \mathbf{0} & \mathbf{0} & \mathbf{0} & \mathbf{0} \\ \mathbf{0} & \mathbf{0} & \mathbf{0} & \mathbf{0} & \mathbf{0} & \mathbf{0} & \mathbf{0} & \mathbf{I} \\ \mathbf{0} & \mathbf{0} & \mathbf{0} & \mathbf{0} & \mathbf{I} & \mathbf{0} & \mathbf{0} & \mathbf{0} \end{bmatrix} $$

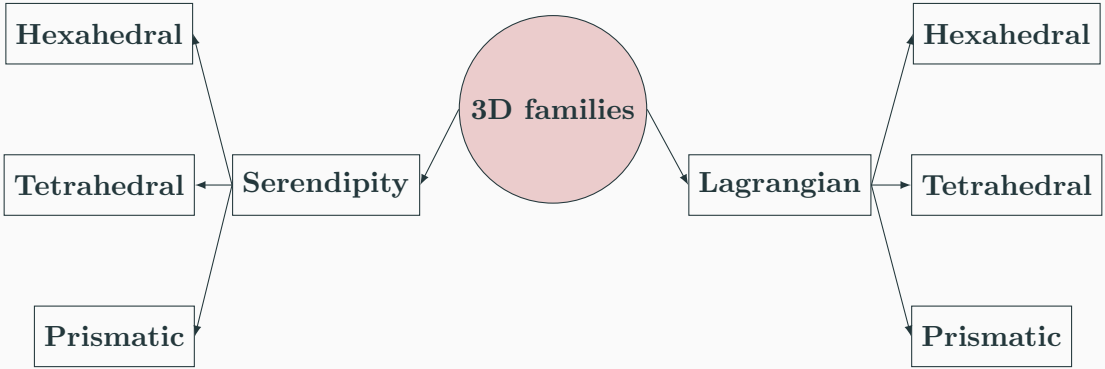
Node id. local	Node id. global
1	4
2	8
3	5

Assembly of the mass matrix: local to global

$$
{}^1\mathbf{L}^T {}^1\mathbf{M} {}^1\mathbf{L} = \begin{bmatrix}
\mathbf{0} & \mathbf{0} & \mathbf{0} & \mathbf{0} & \mathbf{0} & \mathbf{0} & \mathbf{0} & \mathbf{0} \\
\mathbf{0} & \mathbf{0} & \mathbf{0} & \mathbf{0} & \mathbf{0} & \mathbf{0} & \mathbf{0} & \mathbf{0} \\
\mathbf{0} & \mathbf{0} & \mathbf{0} & \mathbf{0} & \mathbf{0} & \mathbf{0} & \mathbf{0} & \mathbf{0} \\
\mathbf{0} & \mathbf{0} & \mathbf{0} & {}^1\mathbf{M}_{11} & {}^1\mathbf{M}_{13} & \mathbf{0} & \mathbf{0} & {}^1\mathbf{M}_{12} \\
\mathbf{0} & \mathbf{0} & \mathbf{0} & {}^1\mathbf{M}_{31} & {}^1\mathbf{M}_{33} & \mathbf{0} & \mathbf{0} & {}^1\mathbf{M}_{32} \\
\mathbf{0} & \mathbf{0} & \mathbf{0} & \mathbf{0} & \mathbf{0} & \mathbf{0} & \mathbf{0} & \mathbf{0} \\
\mathbf{0} & \mathbf{0} & \mathbf{0} & \mathbf{0} & \mathbf{0} & \mathbf{0} & \mathbf{0} & \mathbf{0} \\
\mathbf{0} & \mathbf{0} & \mathbf{0} & {}^1\mathbf{M}_{21} & {}^1\mathbf{M}_{23} & \mathbf{0} & \mathbf{0} & {}^1\mathbf{M}_{22}
\end{bmatrix}
$$

3D finite elements

Three-dimensional families of finite elements



Serendipity vs Lagrangian elements

Lagrangian finite elements

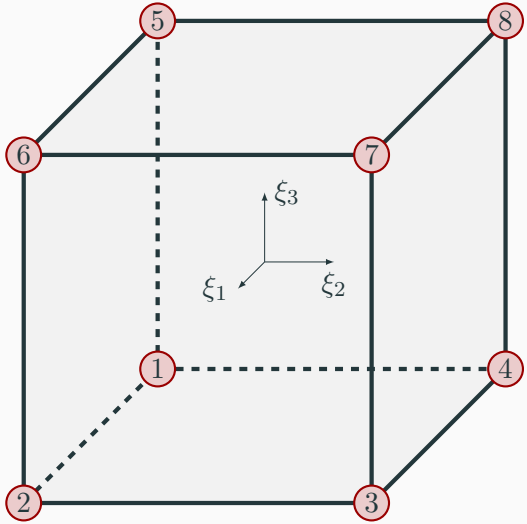
- Defined using Lagrange polynomials.
- Nodes are placed at both *element edges* and *inside the element*.
- Suitable for higher-order accuracy in both structured and unstructured meshes.

Serendipity finite elements

- Defined using a reduced number of nodes compared to Lagrangian elements.
- Nodes exist *only on the edges* (no interior nodes for 2D elements or nodes on faces for 3D elements).
- More computationally efficient but less accurate for high-order approximations.

Trilinear hexahedral Lagrangian element

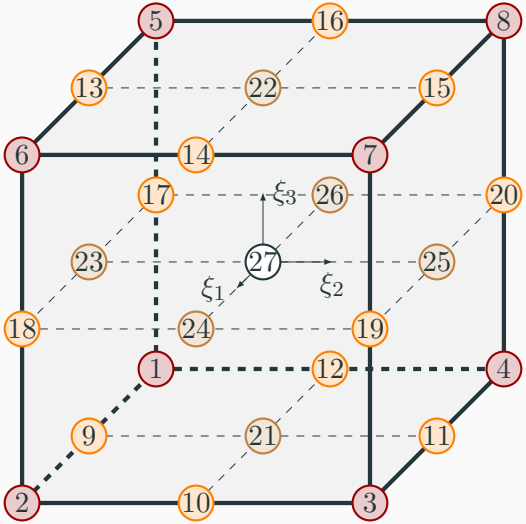
Node	Coordinates
1	$(-1,-1,-1)$
2	$(+1,-1,-1)$
3	$(+1,+1,-1)$
4	$(-1,+1,-1)$
5	$(-1,-1,+1)$
6	$(+1,-1,+1)$
7	$(+1,+1,+1)$
8	$(-1,+1,+1)$



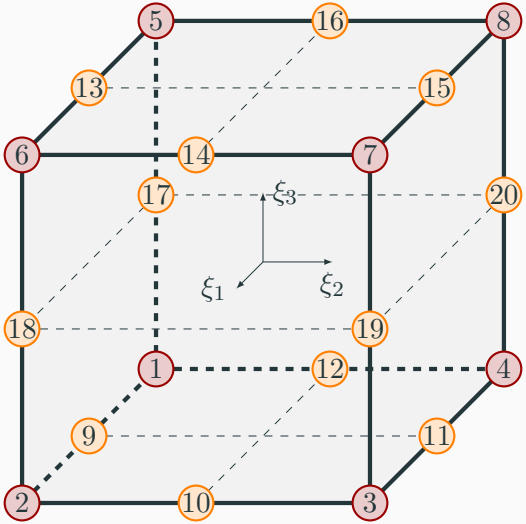
Base functions for trilinear Lagrangian hexahedral elements

$$
\begin{aligned}
{}^{a}h_1 &= 0.125(1-\xi_1)(1-\xi_2)(1-\xi_3)\\
{}^{a}h_2 &= 0.125(1+\xi_1)(1-\xi_2)(1-\xi_3)\\
{}^{a}h_3 &= 0.125(1+\xi_1)(1+\xi_2)(1-\xi_3)\\
{}^{a}h_4 &= 0.125(1-\xi_1)(1+\xi_2)(1-\xi_3)\\
{}^{a}h_5 &= 0.125(1-\xi_1)(1-\xi_2)(1+\xi_3)\\
{}^{a}h_6 &= 0.125(1+\xi_1)(1-\xi_2)(1+\xi_3)\\
{}^{a}h_7 &= 0.125(1+\xi_1)(1+\xi_2)(1+\xi_3)\\
{}^{a}h_8 &= 0.125(1-\xi_1)(1+\xi_2)(1+\xi_3)
\end{aligned}
$$

Triquadratic hexahedral Lagrangian element



Triquadratic hexahedral serendipity element

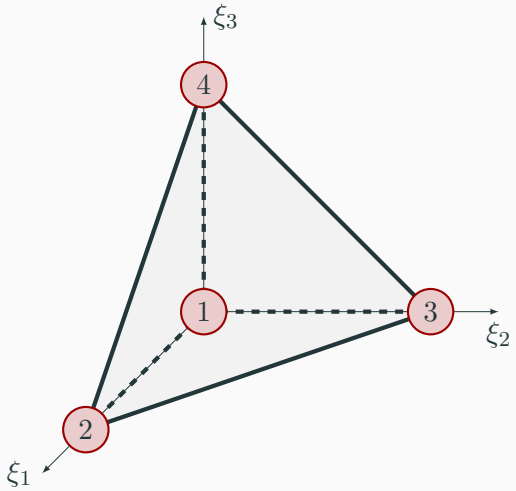


Trilinear tetrahedral Lagrangian element

Node	Coordinates
1	$(0,0,0)$
2	$(1,0,0)$
3	$(0,1,0)$
4	$(0,0,1)$

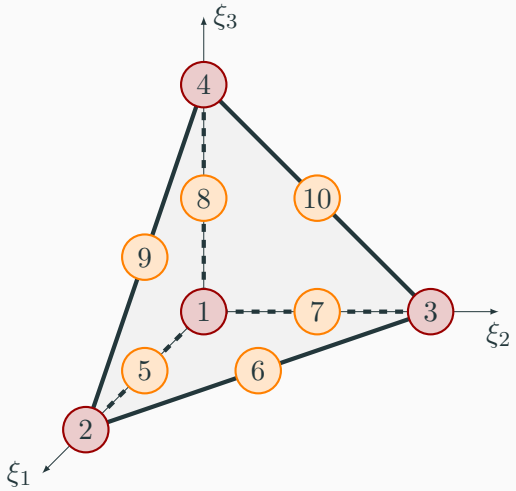
Base functions

$$^a h_1 = 1 - \xi_1 - \xi_2 - \xi_3$$
$$^a h_2 = \xi_1$$
$$^a h_3 = \xi_2$$
$$^a h_4 = \xi_3$$



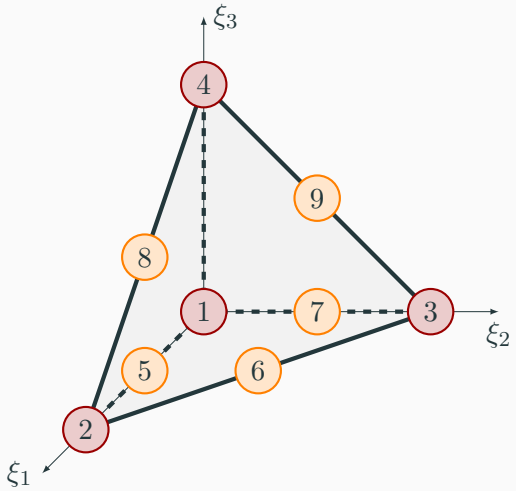
Triquadratic tetrahedral Lagrangian element

Node	Coordinates
1	$(0, 0, 0)$
2	$(1, 0, 0)$
3	$(0, 1, 0)$
4	$(0, 0, 1)$
5	$(0.5, 0, 0)$
6	$(0.5, 0.5, 0)$
7	$(0, 0.5, 0)$
8	$(0, 0, 0.5)$
9	$(0.5, 0, 0.5)$
10	$(0, 0.5, 0.5)$



Triquadratic tetrahedral serendipidy element

Node	Coordinates
1	$(0, 0, 0)$
2	$(1, 0, 0)$
3	$(0, 1, 0)$
4	$(0, 0, 1)$
5	$(0.5, 0, 0)$
6	$(0.5, 0.5, 0)$
7	$(0, 0.5, 0)$
8	$(0.5, 0, 0.5)$
9	$(0, 0.5, 0.5)$

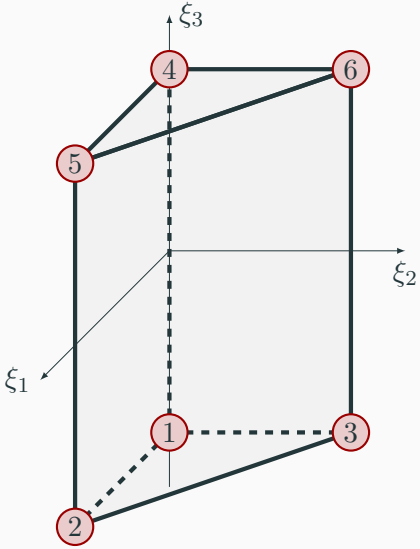


Trilinear prismatic Lagrangian element

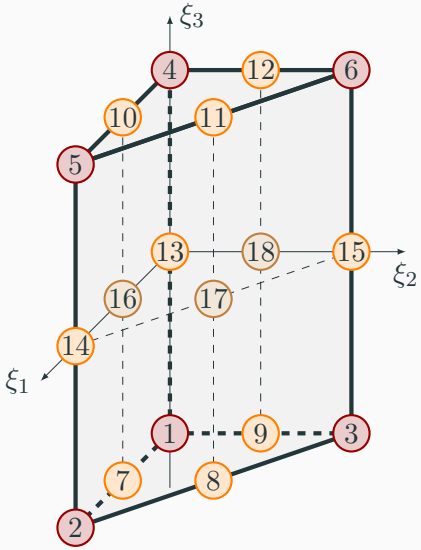
Node	Coordinates
1	$(0, 0, -1)$
2	$(1, 0, -1)$
3	$(0, 1, -1)$
4	$(0, 0, 1)$
5	$(1, 0, 1)$
6	$(0, 1, 1)$

Base functions

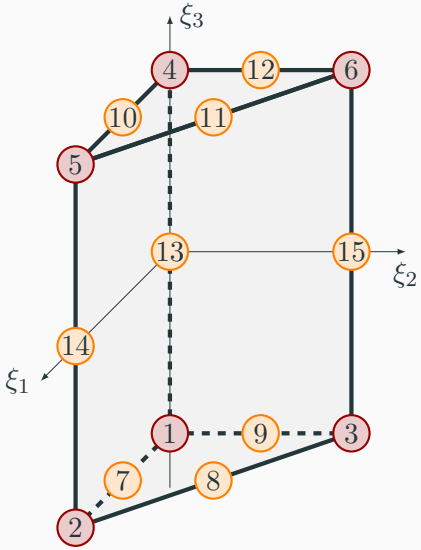
$$
\begin{aligned}
{}^a h_1 &= (1 - \xi_1 - \xi_2)(1 - \xi_3)/2 \\
{}^a h_2 &= \xi_1(1 - \xi_3)/2 \\
{}^a h_3 &= \xi_2(1 - \xi_3)/2 \\
{}^a h_4 &= (1 - \xi_1 - \xi_2)(1 + \xi_3)/2 \\
{}^a h_5 &= \xi_1(1 + \xi_3)/2 \\
{}^a h_6 &= \xi_2(1 + \xi_3)/2
\end{aligned}
$$



Triquadratic prismatic Lagrangian element



Triquadratic prismatic serendipidy element



Numerical integration

Gauss-Legendre quadrature

- Numerical integration helps automate the calculation of the elementary quantities ${}^e\mathbf{K}$, ${}^e\mathbf{M}$ and ${}^e\mathbf{r}$.
- Integration is approximated as:

$$\int_{\Omega^a} f(\boldsymbol{\xi}) d\boldsymbol{\xi} \approx \sum_{i=1}^{r_1} \sum_{j=1}^{r_2} \sum_{k=1}^{r_3} \omega_i^1 \omega_j^2 \omega_k^3 f(\xi_1^i, \xi_2^j, \xi_3^k)$$

where ξ_j^i are known as *Gauss points* and ω_i^j are their associated *Gauss weights*.

☞ If f is a polynomial of degree $d_i = 2r_i - 1$ in the variable ξ_i, then the Gauss-Legendre approximation formula is **exact**.

Numerical integration over hexahedral solid elements

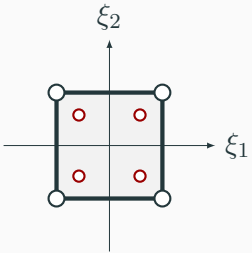
This allows us to compute the stiffness matrix, the mass matrix and the load vector numerically as:

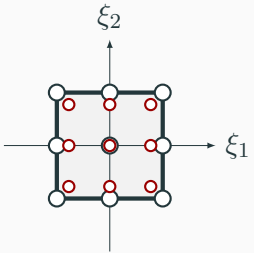
$$ {}^e\mathbf{K} \approx \sum_{i=1}^{r_1}\sum_{j=1}^{r_2}\sum_{k=1}^{r_3} \omega_i^1\omega_j^2\omega_k^3 \left[(\nabla_{\boldsymbol{\xi}}\, {}^a\mathbf{H}^e\mathbf{J}^{-1})^T\mathbf{C}(\nabla_{\boldsymbol{\xi}}\, {}^a\mathbf{H}^e\mathbf{J}^{-1})^e j\right]_{\xi_1=\xi_1^i,\xi_2=\xi_2^j,\xi_3=\xi_3^k} $$

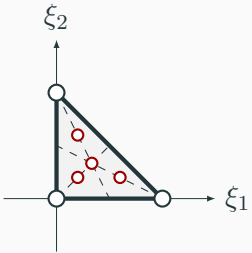
$$ {}^e\mathbf{M} \approx \sum_{i=1}^{r_1}\sum_{j=1}^{r_2}\sum_{k=1}^{r_3} \omega_i^1\omega_j^2\omega_k^3 \left[\rho\, {}^a\mathbf{H}^T {}^a\mathbf{H}\right]_{\xi_1=\xi_1^i,\xi_2=\xi_2^j,\xi_3=\xi_3^k} $$

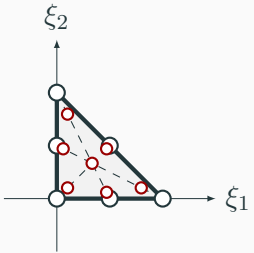
$$ {}^e\mathbf{r} \approx \sum_{i=1}^{r_1}\sum_{j=1}^{r_2} \tilde{\omega}_i^1\tilde{\omega}_j^2 \left[{}^a\mathbf{H}^T\hat{\mathbf{f}}\right]_{\xi_1=\tilde{\xi}_1^i,\xi_2=\tilde{\xi}_2^j} + \sum_{i=1}^{r_1}\sum_{j=1}^{r_2}\sum_{k=1}^{r_3} \omega_i^1\omega_j^2\omega_k^3 \left[{}^a\mathbf{H}^T\mathbf{f}\right]_{\xi_1=\xi_1^i,\xi_2=\xi_2^j,\xi_3=\xi_3^k} $$

An illustration of Gauss points - 2d elements

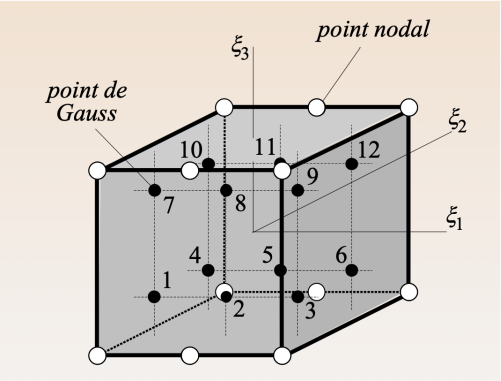


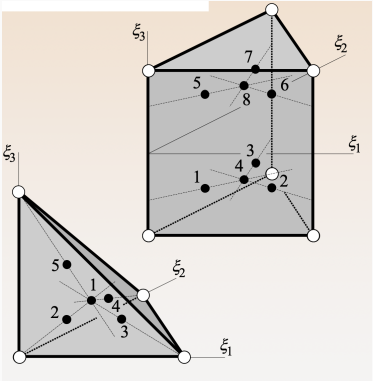






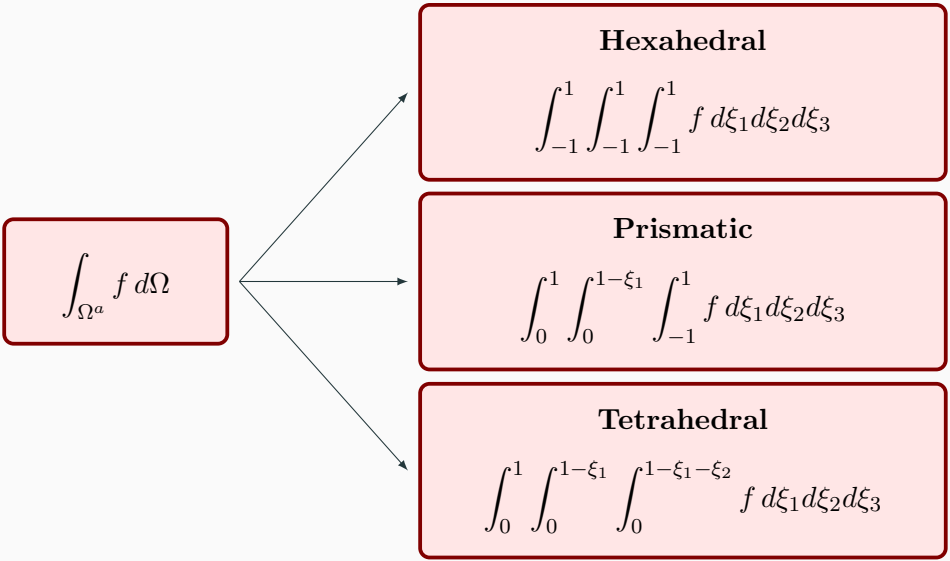
An illustration of Gauss points - 3d elements





(Credit: Thomas Gmür - Dynamique numérique des solides et des structures)

Numerical integration in 3d



Gauss-Legendre numerical integration

Hexahedral

$$\int_{-1}^{1}\int_{-1}^{1}\int_{-1}^{1} f\, d\xi_1 d\xi_2 d\xi_3 \rightarrow \approx \sum_{i=1}^{r_1}\sum_{j=1}^{r_2}\sum_{k=1}^{r_3} \omega_i^1 \omega_j^2 \omega_k^3 f(\xi_1^i, \xi_2^j, \xi_3^k)$$

Prismatic

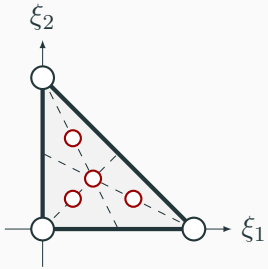
$$\int_{0}^{1}\int_{0}^{1-\xi_1}\int_{-1}^{1} f\, d\xi_1 d\xi_2 d\xi_3 \rightarrow \approx \sum_{i=1}^{r_1}\sum_{k=1}^{r_2} \omega_i^{12} \omega_k^3 f(\xi_1^i, \xi_2^i, \xi_3^k)$$

Tetrahedral

$$\int_{0}^{1}\int_{0}^{1-\xi_1}\int_{0}^{1-\xi_1-\xi_2} f\, d\xi_1 d\xi_2 d\xi_3 \rightarrow \approx \sum_{i=1}^{r_1} \omega_i^{123} f(\xi_1^i, \xi_2^i, \xi_3^i)$$

Example of numerical integration using Gauss-Legendre quadrature in 2d

$$\int_{\Omega^a} f(\xi_1, \xi_2) d\xi_1 d\xi_2 \approx \sum_{i=1}^{4} \omega_i f(\xi_1^i, \xi_2^i)$$



Gauss points (ξ_1^i, ξ_2^i)	Weights (ω_i)
$(1/5, 1/5)$	$25/96$
$(3/5, 1/5)$	$25/96$
$(1/5, 3/5)$	$25/96$
$(1/3, 1/3)$	$9/32$

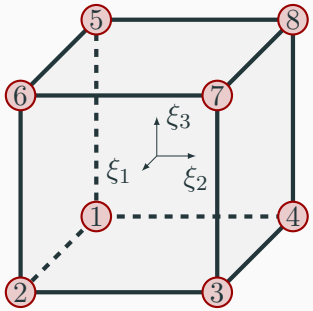
- Since we are using 2 Gauss points in each *direction*, the approximation is exact for polynomials of degree up to 3 in each variable.

Number of Gauss points for exact integration

Trilinear hexahedral element

- Basis functions are linear in each variable.
- Exact integration with:
  - $2 \times 2 \times 2$ Gauss points for ${}^e\mathbf{M}$
  - $1 \times 1 \times 1$ Gauss points for ${}^e\mathbf{K}$

  if the element is not distorted.
- Note that in case of distortion:
  - Jacobian is a function of $\boldsymbol{\xi}$: ${}^e j = {}^e j(\boldsymbol{\xi})$
  - ${}^e\mathbf{J}^{-1} = {}^e\mathbf{J}^{-1}(\boldsymbol{\xi})$ function of $\boldsymbol{\xi}$ at the denominator.
  - $\frac{\partial^e \mathbf{H}}{\partial \mathbf{x}} = \frac{\partial^a \mathbf{H}}{\partial \boldsymbol{\xi}} {}^e\mathbf{J}^{-1}$ is not a polynomial function of $\boldsymbol{\xi}$ anymore.



☞ Gauss quadrature will never be exact in case of distortion.

Number of Gauss points for exact integration

Rule for exact integration: evaluate the number of Gauss points on the archetypal element Ω^a (not on ${}^e\Omega$). Let n_i be the maximum number of nodes in direction ξ_i.

- **Hexahedral:** $n_1 \times n_2 \times n_3$ Gauss points.
- **Prismatic:** $n_{12} \times n_3$ Gauss points.
- **Tetrahedral:** n_{123} Gauss points

Example of exact integration for **mass matrix**:

- Trilinear Hexahedral: $n_1 \times n_2 \times n_3 = 2 \times 2 \times 2$
- Trilinear Tetrahedral: $n_{123} = 5$
- Trilinear Prismatic: $n_{12} \times n_3 = 4 \times 2$

Comparison of exact and reduced integration

Exact integration

- ✓ Integrals are computed exactly
- ✓ Monotonicity of convergence
- ✗ Overestimation of stiffness
- ✗ Risk of locking in linear elements

Reduced integration

- ✓ Reduction of computational costs
- ✗ Loss of monotonicity of convergence
- ✗ Hourglass effect in hexahedral structured meshes

Selective reduced integration:

- involves using fewer integration points in certain areas and more in others, based on the specific requirements of the analysis,
- reduce computational costs while maintaining accuracy in critical areas,
- mitigate the hourglass effect.

MATLAB PDE Toolbox

Examples

1. Error analysis: exact vs numerical integration
2. Tuning fork example from MATLAB PDE Toolbox
3. Transient linear analysis of cantilever beam

▸ Go to Matlab Drive