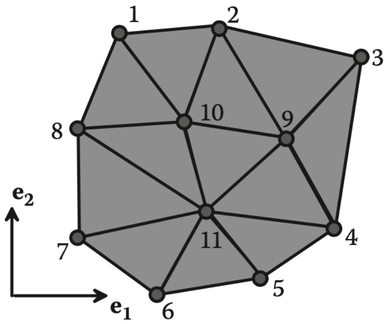
2.13 Extend 1D FEM to 2D and 3D

1. Basic ideas

It is straightforward to extend the 1D case to more general problems. The basic ideas are:

(1) In both two and three dimensions, we divide up our solid of interest into a number of elements, shown schematically for a 2D region in the figure.



(2) We define interpolation functions $N^a(\xi_i)$ for each element in terms of a local, dimensionless, coordinate system within the element. The displacement field and the position of a point inside an element are computed in terms of the interpolation functions as:

$$\text{(2.82)}\quad u_i = \sum_{a=1}^{N_e} N^a(\xi_j)\, u_i^a$$

$$\text{(2.83)}\quad x_i = \sum_{a=1}^{N_e} N^a(\xi_j)\, x_i^a$$

where $N^a(\xi_j)$ denote the shape functions at the position ξ_j in *normalized coordinates*, u_i^a , x_i^a denote the displacement values and coordinates of the nodes on the element respectively, and N_e is the number of nodes on the element.

3). Recalling the stiffness matrix and the force vector of solids presented in Eqs. (2.56) and (2.57):

$$K_{aibk} = \int_R C_{ijk\ell} \frac{\partial N^a(\boldsymbol{x})}{\partial x_j} \frac{\partial N^b(\boldsymbol{x})}{\partial x_\ell} dV$$

$$F_i^a = \int_R b_i N^a(\boldsymbol{x})\, dV + \int_{\partial_2 R} t_i^* N^a(\boldsymbol{x})\, dA,$$

the element stiffness matrix and force vector for each element can be defining as:

$$\text{(2.84)}\quad k_{aibk}^{(\ell)} = \int_{V_e^{(\ell)}} C_{ijk\ell} \frac{\partial N^a(\boldsymbol{x})}{\partial x_j} \frac{\partial N^b(\boldsymbol{x})}{\partial x_\ell} dV$$

$$\text{(2.85)}\quad f_i^{a(\ell)} = \int_{V_e^{(\ell)}} b_i N^a(\boldsymbol{x})\, dV + \int_{\partial_2 V_e^{(\ell)}} t_i^* N^a(\boldsymbol{x})\, dA$$

where $k_{aibk}^{(\ell)}$ denote the element stiffness matrix for the ℓ-th element, $f_i^{a(\ell)}$ denote the element force vector for the ℓ-th element, and $V_e^{(\ell)}$ denotes the volume (in 3D) or area (in 2D) of the ℓ-th element, whereas $\partial_2 V_e^{(\ell)}$ denotes the surface of the ℓ-th element.

(4) The volume integrals over each element are calculated by expressing the volume or surface integral in terms of the dimensionless coordinates and then evaluating the integrals numerically, using a Gauss quadrature formula of the form:

$$\text{(2.86)}\quad \int_{V_e} f(\xi_i)\, dV_\xi = \sum_{I=1}^{M} w_I\, f(\xi_i^I)$$

where w_I are a set of $I = 1...M$ integration weights (just numbers usually available by checking a reference table), ξ_i^I are a set of coordinates that are selected to make the integration scheme as accurate as possible (also just numbers and available in the reference table).

(5) The global stiffness matrix (Eq. 2.56) and the force vector (Eq. 2.57) of solids can be then computed by summing the contribution from each element as:

$$\text{(2.87)}\quad K_{aibk} = \sum_{\ell}^{N_m} k_{aibk}^{(\ell)}$$

$$\text{(2.88)}\quad F_i^a = \sum_{\ell}^{N_m} f_i^{a(\ell)}$$

(6) The stiffness matrix is modified to enforced any prescribed displacements as usual.

(7) The system of equations:

$$(2.89)\ K_{aibk}\, u_k^b = F_i^a \qquad \forall : x_k^a \text{ not on } \partial_1 R$$
$$(2.90)\ u_i^a = u_i^*(x_k^a) \qquad \forall : x_k^a \text{ on } \partial_1 R$$

is solved for the unknown nodal displacements.

(8) The stress and strain within each element are then obtained from the solved nodal displacements.

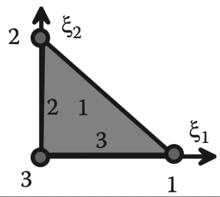
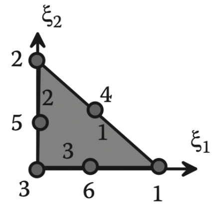
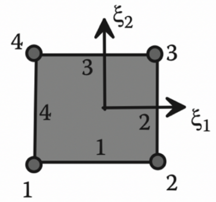
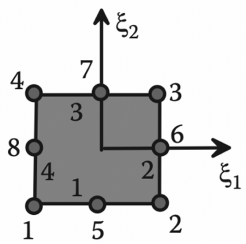
To implement the procedure discussed above, we need to (1) define the element interpolation functions, (2) express the integrals for the element stiffness matrices and force vectors in terms of normalized coordinates, and (3) formulate a numerical integration scheme to evaluate the element stiffness matrices and force vectors. These details are addressed in the sections to follow.

2. Interpolation functions

(1) Interpolation functions for 2D elements

The interpolation functions for 2D elements are listed in the following table. They are defined for the region $0 \le \xi_1 \le +1$ and $0 \le \xi_2 \le +1$ for triangular elements and $-1 \le \xi_1 \le +1$ and $-1 \le \xi_2 \le +1$ for quadrilateral elements. The numbers shown inside the element show the convention used to number the element faces.

2D shape function

$N^1 = \xi_1 \qquad N^2 = \xi_2$ $N^3 = 1 - \xi_1 - \xi_2$	
$N^1 = (2\xi_1 - 1)\xi_1 \qquad N^2 = (2\xi_2 - 1)\xi_2$ $N^3 = (2(1 - \xi_1 - \xi_2) - 1)(1 - \xi_1 - \xi_2)$ $N^4 = 4\xi_1\xi_2 \qquad N^5 = 4\xi_2(1 - \xi_1 - \xi_2)$ $N^6 = 4\xi_1(1 - \xi_1 - \xi_2)$	
$N^1 = 0.25(1 - \xi_1)(1 - \xi_2)$ $N^2 = 0.25(1 + \xi_1)(1 - \xi_2)$ $N^3 = 0.25(1 + \xi_1)(1 + \xi_2)$ $N^4 = 0.25(1 - \xi_1)(1 + \xi_2)$	
$N^1 = -(1-\xi_1)(1-\xi_2)(1+\xi_1+\xi_2)/4$ $N^2 = (1+\xi_1)(1-\xi_2)(\xi_1-\xi_2-1)/4$ $N^3 = (1+\xi_1)(1+\xi_2)(\xi_1+\xi_2-1)/4$ $N^4 = (1-\xi_1)(1+\xi_2)(\xi_2-\xi_1-1)/4$ $N^5 = (1-\xi_1^2)(1-\xi_2)/2 \qquad N^6 = (1+\xi_1)(1-\xi_2^2)/2$ $N^7 = (1-\xi_1^2)(1+\xi_2)/2 \qquad N^8 = (1-\xi_1)(1-\xi_2^2)/2$	

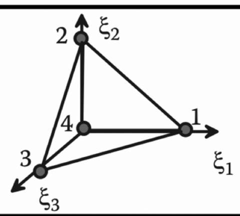
(2) Interpolation functions for 3D elements

The interpolation functions for 3D elements are listed in the table. The tetrahedral elements are defined for the region $0 \le \xi_i \le +1$; while the cubic elements are defined for the region $-1 \le \xi_i \le +1$. The element faces are also numbered as listed in the following table.

3D shape function

$N^1 = \xi_1 \qquad N^2 = \xi_2$

$N^3 = \xi_3 \qquad N^4 = 1 - \xi_1 - \xi_2 - \xi_3$



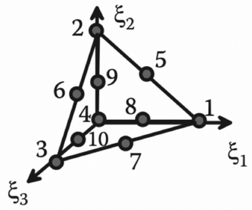
$N^1 = (2\xi_1 - 1)\,\xi_1 \qquad N^2 = (2\xi_2 - 1)\,\xi_2$

$N^3 = (2\xi_3 - 1)\,\xi_3 \qquad N^4 = (2\xi_4 - 1)\,\xi_4$

$N^5 = 4\xi_1\xi_2 \qquad N^6 = 4\xi_2\xi_3$

$N^7 = 4\xi_3\xi_1 \qquad N^8 = 4\xi_4\xi_1 \qquad N^9 = 4\xi_2\xi_4 \qquad N^{10} = 4\xi_3\xi_4$

$\xi_4 = 1 - \xi_1 - \xi_2 - \xi_3$

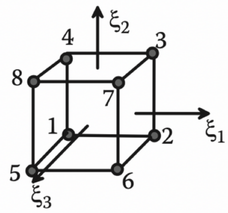


$N^1 = (1 - \xi_1)(1 - \xi_2)(1 - \xi_3)/8 \qquad N^2 = (1 + \xi_1)(1 - \xi_2)(1 - \xi_3)/8$

$N^3 = (1 + \xi_1)(1 + \xi_2)(1 - \xi_3)/8 \qquad N^4 = (1 - \xi_1)(1 + \xi_2)(1 - \xi_3)/8$

$N^5 = (1 - \xi_1)(1 - \xi_2)(1 + \xi_3)/8 \qquad N^6 = (1 + \xi_1)(1 - \xi_2)(1 + \xi_3)/8$

$N^7 = (1 + \xi_1)(1 + \xi_2)(1 + \xi_3)/8 \qquad N^8 = (1 - \xi_1)(1 + \xi_2)(1 + \xi_3)/8$



$N^1 = (1-\xi_1)(1-\xi_2)(1-\xi_3)(-\xi_1-\xi_2-\xi_3-2)/8$

$N^2 = (1+\xi_1)(1-\xi_2)(1-\xi_3)(\xi_1-\xi_2-\xi_3-2)/8$

$N^3 = (1+\xi_1)(1+\xi_2)(1-\xi_3)(\xi_1+\xi_2-\xi_3-2)/8$

$N^4 = (1-\xi_1)(1+\xi_2)(1-\xi_3)(-\xi_1+\xi_2-\xi_3-2)/8$

$N^5 = (1-\xi_1)(1-\xi_2)(1+\xi_3)(-\xi_1-\xi_2+\xi_3-2)/8$

$N^6 = (1+\xi_1)(1-\xi_2)(1+\xi_3)(+\xi_1-\xi_2+\xi_3-2)/8$

$N^7 = (1+\xi_1)(1+\xi_2)(1+\xi_3)(+\xi_1+\xi_2+\xi_3-2)/8$

$N^8 = (1-\xi_1)(1+\xi_2)(1+\xi_3)(-\xi_1+\xi_2+\xi_3-2)/8$

$N^9 = (1-\xi_1^2)(1-\xi_2)(1-\xi_3)/4 \qquad N^{10} = (1+\xi_1)(1-\xi_2^2)(1-\xi_3)/4$

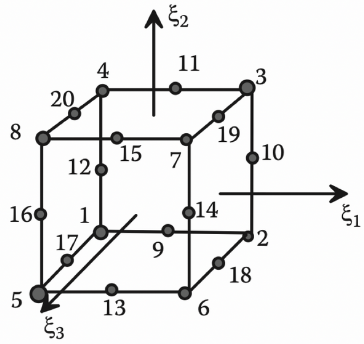
$N^{11} = (1-\xi_1^2)(1+\xi_2)(1-\xi_3)/4 \qquad N^{12} = (1-\xi_1)(1-\xi_2^2)(1-\xi_3)/4$

$N^{13} = (1-\xi_1^2)(1-\xi_2)(1+\xi_3)/4 \qquad N^{14} = (1+\xi_1)(1-{\xi_2}^2)(1+\xi_3)/4$

$N^{15} = (1-\xi_1^2)(1+\xi_2)(1+\xi_3)/4 \qquad N^{16} = (1-\xi_1)(1-\xi_2^2)(1+\xi_3)/4$

$N^{17} = (1-\xi_1)(1-\xi_2)(1-\xi_3^2)/4 \qquad N^{18} = (1+\xi_1)(1-\xi_2)(1-\xi_3^2)/4$

$N^{19} = (1+\xi_1)(1+\xi_2)(1-\xi_3^2)/4 \qquad N^{20} = (1-\xi_1)(1+\xi_2)(1-\xi_3^2)/4$



Tetrahedral element:		Brick element:	
Face 1	Nodes 1, 2, 3	Face 1	Nodes 1, 2, 3, 4
Face 2	Nodes 1, 4, 2	Face 2	Nodes 5, 8, 7, 6
Face 3	Nodes 2, 4, 3	Face 3	Nodes 1, 5, 6, 3
Face 4	Nodes 3, 4, 1	Face 4	Nodes 2, 6, 7, 3
		Face 5	Nodes 3, 7, 8, 4
		Face 6	Nodes 4, 8, 5, 1

3. Integrals for stiffness and force in terms of normalized coordinates

To evaluate the integrals for stiffness and force in terms of normalized coordinates:

$$k_{aibk}^{(\ell)} = \int_{V_e^{(\ell)}} C_{ijk\ell} \frac{\partial N^a(\boldsymbol{x})}{\partial x_j} \frac{\partial N^b(\boldsymbol{x})}{\partial x_\ell} dV$$

$$f_i^{a(\ell)} = \int_{V_e^{(\ell)}} b_i N^a(\boldsymbol{x})\, dV + \int_{\partial_2 V_e^{(\ell)}} t_i^* N^a(\boldsymbol{x})\, dA$$

we need to 1). find a way to calculate the derivatives of the shape functions in terms of ξ_i, and 2). map the volume (or area) integral to the proper domain region of ξ_i.

(1) Calculating the shape function derivatives

The shape function derivatives can be evaluated by writing:

$$\frac{\partial N^a}{\partial x_j} = \frac{\partial N^a}{\partial \xi_i} \frac{\partial \xi_i}{\partial x_j} \tag{2.91}$$

where the derivatives $\dfrac{\partial N^a}{\partial \xi_i}$ are easy to compute by just differentiating the expressions given in the above table. As for computing $\dfrac{\partial \xi_i}{\partial x_j}$, recall that the coordinates of a point at position ξ_i within an element can be determined as

$$x_i = \sum_{a=1}^{Ne} N^a\, x_i^a \tag{2.92}$$

where N_e denotes the number of nodes on the element. Hence,

$$\frac{\partial x_i}{\partial \xi_j} = \frac{\partial}{\partial \xi_j} \left(\sum_{a=1}^{Ne} N^a\, x_i^a \right) = \sum_{a=1}^{Ne} \frac{\partial N^a}{\partial \xi_j} x_i^a \tag{2.93}$$

Note that $\partial x_i / \partial \xi_j$ is a 2 × 2 matrix (in two dimensions) or a 3 × 3 matrix (in three dimensions). Let's look at the relation between the coordinate systems x_i and ξ_i in two dimensions as an illustration:

$$dx_1 = \frac{\partial x_1}{\partial \xi_1} d\xi_1 + \frac{\partial x_1}{\partial \xi_2} d\xi_2 = \frac{\partial x_1}{\partial \xi_1} d\xi_1 + \frac{\partial x_1}{\partial \xi_2} \left(\frac{\partial \xi_2}{\partial x_1} dx_1 + \frac{\partial \xi_2}{\partial x_2} dx_2 \right) \tag{2.94}$$

Along the horizontal edge of an inifitesimal element, i.e., y is constant:

$$dx_2 = 0 \Rightarrow \frac{\partial x_1}{\partial \xi_1}\frac{\partial \xi_1}{\partial x_1} + \frac{\partial x_1}{\partial \xi_2}\frac{\partial \xi_2}{\partial x_1} = 1 \tag{2.95}$$

Along the vertical edge of an inifitesimal element, i.e., x is constant:

$$dx_1 = 0 \Rightarrow \frac{\partial x_1}{\partial \xi_1}\frac{\partial \xi_1}{\partial x_2} + \frac{\partial x_1}{\partial \xi_2}\frac{\partial \xi_2}{\partial x_2} = 0 \tag{2.96}$$

Similarly,

$$dx_2 = \frac{\partial x_2}{\partial \xi_1} d\xi_1 + \frac{\partial x_2}{\partial \xi_2} d\xi_2 = \frac{\partial x_2}{\partial \xi_1} d\xi_1 + \frac{\partial x_2}{\partial \xi_2} \left(\frac{\partial \xi_2}{\partial x_1} dx_1 + \frac{\partial \xi_2}{\partial x_2} dx_2 \right) \tag{2.97}$$

Along the horizontal edge of an inifitesimal element, i.e., y is constant:

$$dx_2 = 0 \Rightarrow \frac{\partial x_2}{\partial \xi_1}\frac{\partial \xi_1}{\partial x_1} + \frac{\partial x_2}{\partial \xi_2}\frac{\partial \xi_2}{\partial x_1} = 0 \tag{2.98}$$

Along the vertical edge of an inifitesimal element, i.e., x is constant:

$$dx_1 = 0 \Rightarrow \frac{\partial x_2}{\partial \xi_1}\frac{\partial \xi_1}{\partial x_2} + \frac{\partial x_2}{\partial \xi_2}\frac{\partial \xi_2}{\partial x_2} = 1 \tag{2.99}$$

Finally, Eqs. (2.90), (2.91), (2.93) and (2.94) can be represented in a matrix form:

$$
\begin{bmatrix} \dfrac{\partial x_1}{\partial \xi_1} & \dfrac{\partial x_1}{\partial \xi_2} \\ \dfrac{\partial x_2}{\partial \xi_1} & \dfrac{\partial x_2}{\partial \xi_2} \end{bmatrix} \begin{bmatrix} \dfrac{\partial \xi_1}{\partial x_1} & \dfrac{\partial \xi_1}{\partial x_2} \\ \dfrac{\partial \xi_2}{\partial x_1} & \dfrac{\partial \xi_2}{\partial x_2} \end{bmatrix} = \begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix} = \boldsymbol{I} \tag{2.100}
$$

$$
\begin{bmatrix} \dfrac{\partial \xi_1}{\partial x_1} & \dfrac{\partial \xi_1}{\partial x_2} \\ \dfrac{\partial \xi_2}{\partial x_1} & \dfrac{\partial \xi_2}{\partial x_2} \end{bmatrix} = \begin{bmatrix} \dfrac{\partial x_1}{\partial \xi_1} & \dfrac{\partial x_1}{\partial \xi_2} \\ \dfrac{\partial x_2}{\partial \xi_1} & \dfrac{\partial x_2}{\partial \xi_2} \end{bmatrix}^{-1} \tag{2.101}
$$

Consequently, $\partial \xi_i / \partial x_j$ is the element at the i-th row and the j-th column of such matrix following the inverse of the matrix called *Jacobian matrix* $\boldsymbol{J}$ having the element $\partial x_i / \partial \xi_j$ at the i-th row and the j-th column, i.e.,

$$
\frac{\partial \xi_i}{\partial x_j} = \left(\frac{\partial x}{\partial \xi} \right)^{-1}_{ij} \tag{2.102}
$$

Hence, for the two-dimensional problems, the Jacobian matrix is:

$$
\boldsymbol{J} = \begin{bmatrix} \dfrac{\partial x_1}{\partial \xi_1} & \dfrac{\partial x_1}{\partial \xi_2} \\ \dfrac{\partial x_2}{\partial \xi_1} & \dfrac{\partial x_2}{\partial \xi_2} \end{bmatrix} \tag{2.103}
$$

for three-dimensional problems, the Jacobian matrix is thus:

$$
\boldsymbol{J} = \begin{bmatrix} \dfrac{\partial x_1}{\partial \xi_1} & \dfrac{\partial x_1}{\partial \xi_2} & \dfrac{\partial x_1}{\partial \xi_3} \\ \dfrac{\partial x_2}{\partial \xi_1} & \dfrac{\partial x_2}{\partial \xi_2} & \dfrac{\partial x_2}{\partial \xi_3} \\ \dfrac{\partial x_3}{\partial \xi_1} & \dfrac{\partial x_3}{\partial \xi_2} & \dfrac{\partial x_3}{\partial \xi_3} \end{bmatrix} \tag{2.104}
$$

(2) Mapping area and volume integrals

Consider the following two typical area and volume integrals:

$$
\int_{V_e} F(x_1, x_2)\, dA = \int_{\hat{V}_e} \hat{F}(\xi_1, \xi_2)\, |\boldsymbol{J}|\, d\xi_1 d\xi_2 \tag{2.105}
$$

$$
\int_{V_e} F(x_1, x_2, x_3)\, dV = \int_{\hat{V}_e} \hat{F}(\xi_1, \xi_2, \xi_3)\, |\boldsymbol{J}|\, d\xi_1 d\xi_2 d\xi_3 \tag{2.106}
$$

where $|\boldsymbol{J}|$ is the determinant of the Jacobian matrix $\boldsymbol{J}$ in Eqs. (2.98) and (2.99) respectively, commonly called the *Jacobian determinant* for precisely mapping the region of 2D area and 3D volume integration.

(3) Mapping boundary integrals in 2D and surface integrals in 3D dimensional problems

Consider the following typical boundary integrals in 2D and surface integrals in 3D:

$$
\int_{S_e} F(x_1, x_2)\, dS = \int_{\hat{S}_e} \hat{F}(\xi_1) \left| \frac{\partial}{\partial \xi_1}(x_1 \boldsymbol{i} + x_2 \boldsymbol{j}) \right| d\xi_1 = \int_{\hat{S}_e} \hat{F}(\xi_1) \sqrt{\left(\frac{\partial x_1}{\partial \xi_1} \right)^2 + \left(\frac{\partial x_2}{\partial \xi_1} \right)^2}\, d\xi_1 \tag{2.107}
$$

(2.108)

$$
\begin{aligned}
&\int_{V_e} F(x_1, x_2, x_3)\, dA \\
&= \int_{\hat{V}_e} \hat{F}(\xi_1, \xi_2) \left| \frac{\partial}{\partial \xi_1}(x_1\boldsymbol{i} + x_2\boldsymbol{j} + x_3\boldsymbol{k}) \times \frac{\partial}{\partial \xi_2}(x_1\boldsymbol{i} + x_2\boldsymbol{j} + x_3\boldsymbol{k}) \right| d\xi_1 d\xi_2 \\
&= \int_{\hat{V}_e} \hat{F}(\xi_1, \xi_2) \left\| \begin{vmatrix} \boldsymbol{i} & \boldsymbol{j} & \boldsymbol{k} \\ \dfrac{\partial x_1}{\partial \xi_1} & \dfrac{\partial x_2}{\partial \xi_1} & \dfrac{\partial x_3}{\partial \xi_1} \\ \dfrac{\partial x_1}{\partial \xi_2} & \dfrac{\partial x_2}{\partial \xi_2} & \dfrac{\partial x_3}{\partial \xi_2} \end{vmatrix} \right\| d\xi_1 d\xi_2 \\
&= \int_{\hat{V}_e} \hat{F}(\xi_1, \xi_2) \left| \left(\frac{\partial x_2}{\partial \xi_1}\frac{\partial x_3}{\partial \xi_2} - \frac{\partial x_3}{\partial \xi_1}\frac{\partial x_2}{\partial \xi_2} \right) \boldsymbol{i} + \left(\frac{\partial x_3}{\partial \xi_1}\frac{\partial x_1}{\partial \xi_2} - \frac{\partial x_1}{\partial \xi_1}\frac{\partial x_3}{\partial \xi_2} \right) \boldsymbol{j} + \left(\frac{\partial x_1}{\partial \xi_1}\frac{\partial x_2}{\partial \xi_2} - \frac{\partial x_2}{\partial \xi_1}\frac{\partial x_1}{\partial \xi_2} \right) \boldsymbol{k} \right| d\xi_1 d\xi_2 \\
&= \int_{\hat{V}_e} \hat{F}(\xi_1, \xi_2) \sqrt{ \left(\frac{\partial x_2}{\partial \xi_1}\frac{\partial x_3}{\partial \xi_2} - \frac{\partial x_3}{\partial \xi_1}\frac{\partial x_2}{\partial \xi_2} \right)^2 + \left(\frac{\partial x_3}{\partial \xi_1}\frac{\partial x_1}{\partial \xi_2} - \frac{\partial x_1}{\partial \xi_1}\frac{\partial x_3}{\partial \xi_2} \right)^2 + \left(\frac{\partial x_1}{\partial \xi_1}\frac{\partial x_2}{\partial \xi_2} - \frac{\partial x_2}{\partial \xi_1}\frac{\partial x_1}{\partial \xi_2} \right)^2 }\, d\xi_1 d\xi_2
\end{aligned}
$$

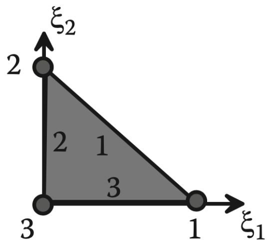
for precisely mapping the region of 2D boundary and 3D surface integration.

4. Numerical Integration Schemes for 2D and 3D Elements

While implementing area, volume, boundary, or surface integration, the integrals over 1D, 2D, or 3D elements need to be evaluated. Finally, we once again adopt a quadrature scheme (The integrals over 1D elements evaluated by the *quadrature formula* have been introduced in Section 2.2.3.) for such numerical Integration:

$$
\int_{\Omega} f(\xi_i)\, dV_\xi = \sum_{I=1}^{M} w_I\, f(\xi_i^I) \tag{2.109}
$$

where ξ_j^I means the j-th normalized coordinate at the I-th integration point, and w_I is the corresponding weight depending the ξ_j^I, and $f(\xi_i)$ is the function evaluated in the normalized coordinate system. The following tables show normalized coordinate at integration points and the corresponding weights of triangular elements:

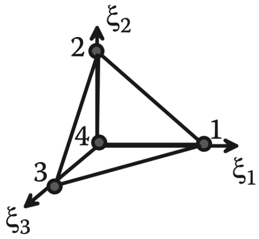


$M\ =\ 1$		
$\xi_1^1 = 1/3$	$\xi_2^1 = 1/3$	$w_1 = 1/2$

$M\ =\ 3$		
$\xi_1^1 = 0.6$	$\xi_2^1 = 0.2$	$w_1 = 1/6$
$\xi_1^2 = 0.2$	$\xi_2^2 = 0.6$	$w_2 = 1/6$
$\xi_1^3 = 0.2$	$\xi_2^3 = 0.2$	$w_3 = 1/6$

$M\ =\ 4$		
$\xi_1^1 = 1/3$	$\xi_2^1 = 1/3$	$w_1 = -27/96$
$\xi_1^2 = 0.6$	$\xi_2^2 = 0.2$	$w_2 = +25/96$
$\xi_1^3 = 0.2$	$\xi_2^3 = 0.6$	$w_3 = +25/96$
$\xi_1^4 = 0.2$	$\xi_2^4 = 0.2$	$w_4 = +25/96$

The following tables show normalized coordinate at integration points and the corresponding weights of tetrahedral elements:

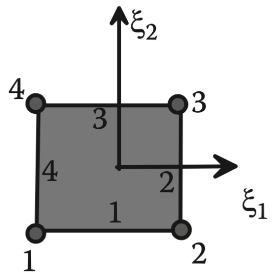


$M = 1$			
$\xi_1^1 = 1/4$	$\xi_2^1 = 1/4$	$\xi_3^1 = 1/4$	$w_1 = 1/6$

$M = 4$			
$\xi_1^1 = \alpha$	$\xi_2^1 = \beta$	$\xi_3^1 = \beta$	$w_1 = 1/24$
$\xi_1^2 = \beta$	$\xi_2^2 = \alpha$	$\xi_3^2 = \beta$	$w_2 = 1/24$
$\xi_1^3 = \beta$	$\xi_2^3 = \beta$	$\xi_3^3 = \alpha$	$w_3 = 1/24$
$\xi_1^4 = \beta$	$\xi_2^4 = \beta$	$\xi_3^4 = \beta$	$w_4 = 1/24$

where $\alpha = 0.58541020$, and $\beta = 0.13819660$

The following tables show normalized coordinate at integration points and the corresponding weights of quadrilateral elements:



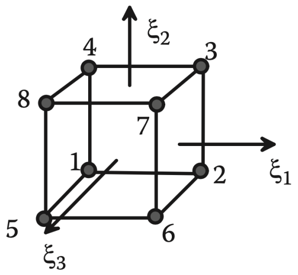
$M = 1$		
$\xi_1^1 = 0$	$\xi_2^1 = 0$	$w_1 = 4$

$M = 4$		
$\xi_1^1 = -0.5773502692$	$\xi_2^1 = -0.5773502692$	$w_1 = 1$
$\xi_1^2 = +0.5773502692$	$\xi_2^2 = -0.5773502692$	$w_2 = 1$
$\xi_1^3 = -0.5773502692$	$\xi_2^3 = +0.5773502692$	$w_3 = 1$
$\xi_1^4 = +0.5773502692$	$\xi_2^4 = +0.5773502692$	$w_4 = 1$

$M = 9$		
$\xi_1^1 = -0.7745966692$	$\xi_2^1 = -0.7745966692$	$w_1 = 0.308641974$
$\xi_1^2 = 0$	$\xi_2^2 = -0.7745966692$	$w_2 = 0.493827159$
$\xi_1^3 = +0.7745966692$	$\xi_2^3 = -0.7745966692$	$w_3 = 0.308641974$
$\xi_1^4 = -0.7745966692$	$\xi_2^4 = 0$	$w_4 = 0.493827159$
$\xi_1^5 = 0$	$\xi_2^5 = 0$	$w_5 = 0.790123455$
$\xi_1^6 = +0.7745966692$	$\xi_2^6 = 0$	$w_6 = 0.493827159$

$M = 9$		
$\xi_1^7 = -0.7745966692$	$\xi_2^7 = +0.7745966692$	$w_7 = 0.308641974$
$\xi_1^8 = 0$	$\xi_2^8 = +0.7745966692$	$w_8 = 0.493827159$
$\xi_1^9 = +0.7745966692$	$\xi_2^9 = +0.7745966692$	$w_9 = 0.308641974$

The following tables show normalized coordinate at integration points and the corresponding weights of cubic elements:



$M = 1$			
$\xi_1^1 = 0$	$\xi_2^1 = 0$	$\xi_3^1 = 0$	$w_1 = 8$

$M = 8$			
$\xi_1^1 = -0.5773502692$	$\xi_2^1 = -0.5773502692$	$\xi_3^1 = -0.5773502692$	$w_1 = 1$
$\xi_1^2 = +0.5773502692$	$\xi_2^2 = -0.5773502692$	$\xi_3^2 = -0.5773502692$	$w_2 = 1$
$\xi_1^3 = -0.5773502692$	$\xi_2^3 = +0.5773502692$	$\xi_3^3 = -0.5773502692$	$w_3 = 1$
$\xi_1^4 = +0.5773502692$	$\xi_2^4 = +0.5773502692$	$\xi_3^4 = -0.5773502692$	$w_4 = 1$
$\xi_1^5 = -0.5773502692$	$\xi_2^5 = -0.5773502692$	$\xi_3^5 = +0.5773502692$	$w_5 = 1$
$\xi_1^6 = +0.5773502692$	$\xi_2^6 = -0.5773502692$	$\xi_3^6 = +0.5773502692$	$w_6 = 1$
$\xi_1^7 = -0.5773502692$	$\xi_2^7 = +0.5773502692$	$\xi_3^7 = +0.5773502692$	$w_7 = 1$
$\xi_1^8 = +0.5773502692$	$\xi_2^8 = +0.5773502692$	$\xi_3^8 = +0.5773502692$	$w_8 = 1$

$M = 27$			
$\xi_1^1 = -0.7745966692$	$\xi_2^1 = -0.7745966692$	$\xi_3^1 = -0.7745966692$	$w_1 = 0.171467763$
$\xi_1^2 = 0$	$\xi_2^2 = -0.7745966692$	$\xi_3^2 = -0.7745966692$	$w_2 = 0.274348421$
$\xi_1^3 = +0.7745966692$	$\xi_2^3 = -0.7745966692$	$\xi_3^3 = -0.7745966692$	$w_3 = 0.171467763$
$\xi_1^4 = -0.7745966692$	$\xi_2^4 = 0$	$\xi_3^4 = -0.7745966692$	$w_4 = 0.274348421$
$\xi_1^5 = 0$	$\xi_2^5 = 0$	$\xi_3^5 = -0.7745966692$	$w_5 = 0.438957474$
$\xi_1^6 = +0.7745966692$	$\xi_2^6 = 0$	$\xi_3^6 = -0.7745966692$	$w_6 = 0.274348421$
$\xi_1^7 = -0.7745966692$	$\xi_2^7 = +0.7745966692$	$\xi_3^7 = -0.7745966692$	$w_7 = 0.171467763$
$\xi_1^8 = 0$	$\xi_2^8 = +0.7745966692$	$\xi_3^8 = -0.7745966692$	$w_8 = 0.274348421$
$\xi_1^9 = +0.7745966692$	$\xi_2^9 = +0.7745966692$	$\xi_3^9 = -0.7745966692$	$w_9 = 0.171467763$
$\xi_1^{10} = -0.7745966692$	$\xi_2^{10} = -0.7745966692$	$\xi_3^{10} = 0$	$w_{10} = 0.274348421$
$\xi_1^{11} = 0$	$\xi_2^{11} = -0.7745966692$	$\xi_3^{11} = 0$	$w_{11} = 0.438957474$

$M\ =\ 27$			
$\xi_1^{12} = +0.7745966692$	$\xi_2^{12} = -0.7745966692$	$\xi_3^{12} = 0$	$w_{12} = 0.274348421$
$\xi_1^{13} = -0.7745966692$	$\xi_2^{13} = 0$	$\xi_3^{13} = 0$	$w_{13} = 0.438957474$
$\xi_1^{14} = 0$	$\xi_2^{14} = 0$	$\xi_3^{14} = 0$	$w_{14} = 0.702331959$
$\xi_1^{15} = +0.7745966692$	$\xi_2^{15} = 0$	$\xi_3^{15} = 0$	$w_{15} = 0.438957474$
$\xi_1^{16} = -0.7745966692$	$\xi_2^{16} = +0.7745966692$	$\xi_3^{16} = 0$	$w_{16} = 0.274348421$
$\xi_1^{17} = 0$	$\xi_2^{17} = +0.7745966692$	$\xi_3^{17} = 0$	$w_{17} = 0.438957474$
$\xi_1^{18} = +0.7745966692$	$\xi_2^{18} = +0.7745966692$	$\xi_3^{18} = 0$	$w_{18} = 0.274348421$
$\xi_1^{19} = -0.7745966692$	$\xi_2^{19} = -0.7745966692$	$\xi_3^{19} = +0.7745966692$	$w_{19} = 0.171467763$
$\xi_1^{20} = 0$	$\xi_2^{20} = -0.7745966692$	$\xi_3^{20} = +0.7745966692$	$w_{20} = 0.274348421$
$\xi_1^{21} = +0.7745966692$	$\xi_2^{21} = -0.7745966692$	$\xi_3^{21} = +0.7745966692$	$w_{21} = 0.171467763$
$\xi_1^{22} = -0.7745966692$	$\xi_2^{22} = 0$	$\xi_3^{22} = +0.7745966692$	$w_{22} = 0.274348421$
$\xi_1^{23} = 0$	$\xi_2^{23} = 0$	$\xi_3^{23} = +0.7745966692$	$w_{23} = 0.438957474$
$\xi_1^{24} = +0.7745966692$	$\xi_2^{24} = 0$	$\xi_3^{24} = +0.7745966692$	$w_{24} = 0.274348421$
$\xi_1^{25} = -0.7745966692$	$\xi_2^{25} = +0.7745966692$	$\xi_3^{25} = +0.7745966692$	$w_{25} = 0.171467763$
$\xi_1^{26} = 0$	$\xi_2^{26} = +0.7745966692$	$\xi_3^{26} = +0.7745966692$	$w_{26} = 0.274348421$
$\xi_1^{27} = +0.7745966692$	$\xi_2^{27} = +0.7745966692$	$\xi_3^{27} = +0.7745966692$	$w_{27} = 0.171467763$

How to choose the number of integration points? There are two considerations. If too many integration points are used, time is wasted without gaining any accuracy. If too few integration points are used, the stiffness matrix may be singular or else the rate of convergence to the exact solution with mesh refinement will be reduced. Hence, *fully integrated elements* with proper numbers of integration points listed in the following table are suggested in FEM code:

Linear triangle (3 nodes): 1 point	Linear tetrahedron (4 nodes): 1 point
Quadratic triangle (6 nodes): 4 points	Quadratic tetrahedron (10 nodes): 4 points
Linear quadrilateral (4 nodes): 4 points	Linear brick (8 nodes): 8 points
Quadratic quadrilateral (8 nodes): 9 points	Quadratic brick (20 nodes): 27 points

5. Demonstration of finite element analysis code

The example FEA code written in MATLAB:

(1) *MatStif* function

```
function cmat = MatStif(ndime,mate)
  emod = mate(2);
  nu = mate(3);
  cmat = zeros(ndime,ndime,ndime,ndime);
  if(ndime == 2)
    if(mate(1) == 1)
      C = [1-nu, nu, 0; nu, 1-nu, 0; 0 0 0.5*(1-2*nu)]*emod/(1+nu)/(1-2*nu);
    else
      C = [1, nu, 0; nu, 1, 0; 0 0 0.5*(1-nu)]*emod/(1-nu*nu);
    end
    for i = 1:2
      for j = 1:2
        ind1 = switchInd2(i,j);
```

```
        for k = 1:2
          for l = 1:2
            ind2 = switchInd2(k,l);
            cmat(i,j,k,l) = C(ind1,ind2);
          end
        end
      end
    end
  else
    lambda = emod*nu/(1+nu)/(1-2*nu);
    mu = 0.5*emod/(1+nu);
    C = [ . . . ];
end

function ind = switchInd2(i,j)
  if(i == 1 && j == 1)
    ind = 1;
  elseif(i == 2 && j == 2)
    ind = 2;
  elseif((i == 1 && j == 2) || (i == 2 && j == 1))
    ind = 3;
  else
    disp('Wrong i & j.');
  end
end

function ind = switchInd3(i,j)
  . . .
end
```

(2) *ElemStif* function

```
function kel = ElemStif(iel,ndime,nelnd,coor,conn,mate)
  kel = zeros(ndime*nelnd,ndime*nelnd);
  coorie = zeros(ndime,nelnd);
  xii = zeros(ndime,1);
  dxdxi = zeros(ndime,ndime);
  dNdx = zeros(nelnd,ndime);
  M = numIntegPt(ndime,nelnd);
  xi = IntegPt(ndime,nelnd,M);
  w = integWt(ndime,nelnd,M);
  for a = 1:nelnd
    for i = 1:ndime
      coorie(i,a) = coor(i,conn(a,iel));
    end
  end
  for im = 1:M
    for i = 1:ndime
      xii(i) = xi(i,im);
    end
    N = ShpFunc(nelnd,ndime,xii);
    dNdxi = ShpFuncDeri(nelnd,ndime,xii);
    dxdxi = 0;
    for i = 1:ndime
      for j = 1:ndime
        for a = 1:nelnd
```

```
          dxdxi(i,j) = dxdxi(i,j)+coorie(i,a)*dNdxi(a,j);
        end
      end
    end
    dxidx = inv(dxdxi);
    jcb = det(dxdxi);
    dNdx = 0.;
    for a = 1:nelnd
      for i = 1:ndime
        for j = 1:ndime
          dNdx(a,i) = dNdx(a,i)+dNdxi(a,j)*dxidx(j,i);
        end
      end
    end
    cmat = MatStif(ndime,mate);
    for a = 1:nelnd
      for i = 1:ndime
        for b = 1:nelnd
          for k = 1:ndime
            ir = ndime*(a-1)+i;
            ic = ndime*(b-1)+k;
            for j = 1:ndime
              for l = 1:ndime
                kel(ir,ic)=kel(ir,ic)+cmat(i,j,k,l)*dNdx(b,l)*dNdx(a,j)*w(im)*jcb;
              end
            end
          end
        end
      end
    end
  end
end
```

(3) *numIntegPt* function

```
function M = numIntegPt(ndime,nelnd)
  if (ndime == 1)
    M = 2;
  elseif (ndime == 2)
    if (nelnd == 3)
      M = 1;
    end
    if (nelnd == 6)
      M = 4;
    end
    if (nelnd == 4)
      M = 4;
    end
    if (nelnd == 8)
      M = 9;
    end
  elseif (ndime == 3)
    if (nelnd == 4)
      M = 1 ;
    end
    if (nelnd == 10)
```

```
      M = 4;
    end
    if (nelnd == 8)
      M = 8;
    end
    if (nelnd == 20)
      M = 27;
    end
  end
end
```

(4) *IntegPt* function

```
function xi = IntegPt(ndime,nelnd,M)
  xi = zeros(ndime,M)
  if (ndime == 1)
    xi(1,1) = -0.577350269;
    xi(1,2) = 0.577350269;
  elseif (ndime == 2)
    if (nelnd == 3)
      xi(1,1) = 1/3.;
      xi(2,1) = 1/3.;
    end
    if (nelnd == 6)
      xi(1,1) = 1/3.;
      xi(2,1) = 1/3.;
      xi(1,2) = 0.6;
      xi(2,2) = 0.2;
      xi(1,3) = 0.2;
      xi(2,3) = 0.6;
      xi(1,4) = 0.2;
      xi(2,4) = 0.2;
    end
    if (nelnd == 4)
      xi(1,1) = -0.577350269;
      xi(2,1) = -0.577350269;
      xi(1,2) = 0.577350269;
      xi(2,2) = -0.577350269;
      xi(1,3) = -0.577350269;
      xi(2,3) = 0.577350269;
      xi(1,4) = 0.577350269;
      xi(2,4) = 0.577350269;
    end
    if (nelnd == 8)
      xi(1,1) = -0.774596669;
      xi(2,1) = -0.774596669;
      xi(1,2) = 0;
      xi(2,2) = -0.774596669;
      xi(1,3) = 0.774596669;
      xi(2,3) = -0.774596669;
      xi(1,4) = -0.774596669;
      xi(2,4) = 0;
      xi(1,5) = 0;
      xi(2,5) = 0;
      xi(1,6) = 0.774596669;
      xi(2,6) = 0;
```

```
      xi(1,7) = -0.774596669;
      xi(2,7) = 0.774596669;
      xi(1,8) = 0.;
      xi(2,8) = 0.774596669;
      xi(1,9) = 0.774596669;
      xi(2,9) = 0.774596669;
    end
  elseif (ndime == 3)
    if (nelnd == 4)
      . . .
    end
    if (nelnd == 10)
      . . .
    end
    if (nelnd == 8)
      . . .
    end
    if (nelnd == 20)
      . . .
    end
  end
end
```

(5) *IntegWt* function

```
function w = integWt(ndime,nelnd,M)
  w = zeros(M,1);
  . . .
end
```

(6) *ShpFunc* function

```
function N = ShpFunc(nelnd,ndime,xii)
  N = zeros(nelnd,1);
  if (ndime == 1)
    N(1) = 0.5*(1.+xii(1));
    N(2) = 0.5*(1.-xii(1));
  elseif (ndime == 2)
    if (nelnd == 3)
      N(1) = xii(1);
      N(2) = xii(2);
      N(3) = 1-xii(1)-xii(2);
    end
    if (nelnd == 6)
      N(1) = (2*xii(1)-1)*xii(1);
      N(2) = (2*xii(2)-1)*xii(2);
      N(3) = (2*(1-xii(1)-xii(2))-1)*(1-xii(1)-xii(2));
      N(4) = 4*xii(1)*xii(2);
      N(5) = 4*xii(2)*(1-xii(1)-xii(2));
      N(6) = 4*xii(1)*(1-xii(1)-xii(2));
    end
    if (nelnd == 4)
      N(1) = 0.25*(1-xii(1))*(1-xii(2));
      N(2) = 0.25*(1+xii(1))*(1-xii(2));
      N(3) = 0.25*(1+xii(1))*(1+xii(2));
      N(4) = 0.25*(1-xii(1))*(1+xii(2));
    end
```

```
    if (nelnd == 8)
      N(1) = -0.25*(1-xii(1))*(1-xii(2))*(1+xii(1)+xii(2));
      N(2) = 0.25*(1+xii(1))*(1-xii(2))*(xii(1)-xii(2)-1);
      N(3) = 0.25*(1+xii(1))*(1+xii(2))*(xii(1)+xii(2)-1);
      N(4) = 0.25*(1-xii(1))*(1+xii(2))*(xii(2)-xii(1)-1);
      N(5) = 0.5*(1-xii(1)*xii(1))*(1-xii(2));
      N(6) = 0.5*(1+xii(1))*(1-xii(2)*xii(2));
      N(7) = 0.5*(1-xii(1)*xii(1))*(1+xii(2));
      N(8) = 0.5*(1-xii(1))*(1-xii(2)*xii(2));
    end
  elseif (ndime == 3)
    if (nelnd == 4)
      . . .
    end
    if (nelnd == 10)
      . . .
    end
    if (nelnd == 8)
      . . .
    end
    if (nelnd == 20)
      . . .
    end
  end
end
```

(7) *ShpFuncDeri* function

```
function dNdxi = ShpFuncDeri(nelnd,ndime,xii)
  dNdxi = zeros(nelnd,ndime);
  if (ndime == 1)
    dNdxi(1,1) = 0.5;
    dNdxi(2,1) = -0.5;
  elseif (ndime == 2)
    if (nelnd == 3)
      dNdxi(1,1) = 1;
      dNdxi(2,2) = 1;
      dNdxi(3,1) = -1;
      dNdxi(3,2) = -1;
    end
    if (nelnd == 6)
      dNdxi(1,1) = 4*xii(1)-1;
      dNdxi(2,2) = 4*xii(2)-1;
      dNdxi(3,1) = -(4*(1-xii(1)-xii(2))-1);
      dNdxi(3,2) = -(4*(1-xii(1)-xii(2))-1);
      dNdxi(4,1) = 4*xii(2);
      dNdxi(4,2) = 4*xii(1);
      dNdxi(5,1) = -4*xii(2);
      dNdxi(5,2) = 4*(1-xii(1)-xii(2))-4*xii(2);
      dNdxi(6,1) = 4*(1-xii(1)-xii(2))-4*xii(1);
      dNdxi(6,2) = -4*xii(1);
    end
    if (nelnd == 4)
      dNdxi(1,1) = -0.25*(1-xii(2));
      dNdxi(1,2) = -0.25*(1-xii(1));
      dNdxi(2,1) = 0.25*(1-xii(2));
```

```
      dNdxi(2,2) = -0.25*(1+xii(1));
      dNdxi(3,1) = 0.25*(1+xii(2));
      dNdxi(3,2) = 0.25*(1+xii(1));
      dNdxi(4,1) = -0.25*(1+xii(2));
      dNdxi(4,2) = 0.25*(1-xii(1));
    end
    if (nelnd == 8)
      dNdxi(1,1) = 0.25*(1-xii(2))*(2*xii(1)+xii(2));
      dNdxi(1,2) = 0.25*(1-xii(1))*(xii(1)+2*xii(2));
      dNdxi(2,1) = 0.25*(1-xii(2))*(2*xii(1)-xii(2));
      dNdxi(2,2) = 0.25*(1+xii(1))*(2*xii(2)-xii(1));
      dNdxi(3,1) = 0.25*(1+xii(2))*(2*xii(1)+xii(2));
      dNdxi(3,2) = 0.25*(1+xii(1))*(2*xii(2)+xii(1));
      dNdxi(4,1) = 0.25*(1+xii(2))*(2*xii(1)-xii(2));
      dNdxi(4,2) = 0.25*(1-xii(1))*(2*xii(2)-xii(1));
      dNdxi(5,1) = -xii(1)*(1-xii(2));
      dNdxi(5,2) = -0.5*(1-xii(1)*xii(1));
      dNdxi(6,1) = 0.5*(1-xii(2)*xii(2));
      dNdxi(6,2) = -(1+xii(1))*xii(2);
      dNdxi(7,1) = -xii(1)*(1+xii(2));
      dNdxi(7,2) = 0.5*(1-xii(1)*xii(1));
      dNdxi(8,1) = -0.5*(1-xii(2)*xii(2));
      dNdxi(8,2) = -(1-xii(1))*xii(2);
    end
  elseif (ndime == 3)
    if (nelnd == 4)
      . . .
    end
    if (nelnd == 10)
      . . .
    end
    if (nelnd == 8)
      . . .
    end
    if (nelnd == 20)
      . . .
    end
  end
end
```