MAE4700/5700

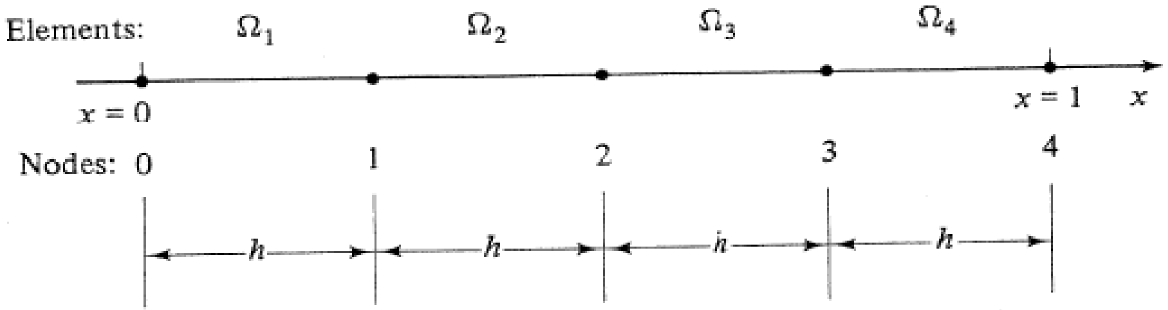
Finite Element Analysis for Mechanical and Aerospace Design

Cornell University, Fall 2009

Nicholas Zabaras
Materials Process Design and Control Laboratory
Sibley School of Mechanical and Aerospace Engineering
101 Rhodes Hall
Cornell University
Ithaca, NY 14853-3801

Finite element basis functions in 1D

- The domain is divided into finite elements and the weight functions and trial solutions are constructed within each element.

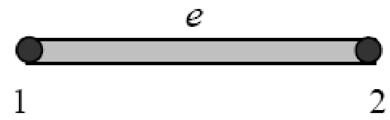


- These functions have to be chosen so that the FEM converges, i.e. as element size, denoted by *h*, decreases, the solution tends to the correct solution (mesh refinement).
- Convergence of the FEM requires basis functions that satisfy continuity and completeness.

Continuity and completeness

- Continuity: the trial solutions and weight functions need to be sufficiently smooth.
  - ✓ The degree of smoothness required depends on the order of the derivatives that appear in the weak form.
  - ✓ For the 2nd order ODEs considered, the derivatives in the weak form are 1st order, thus the weight functions and trial solutions must be H^1 (C^0 piecewise polynomials).
- *Completeness:* the basis functions need to be able to approximate a given smooth function with arbitrary accuracy. As the element size *h* approaches zero, the trial solutions and weight functions and their derivatives up to, and including the highest-order derivative appearing in the weak form, must be capable of assuming constant values
  - ✓ **finite elements can represent rigid body motion and constant strain states exactly**

Linear one-dimensional elements

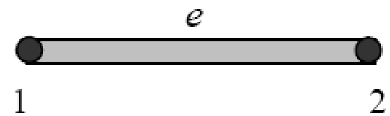


- We need to approximate (interpolate) the trial solution and weight functions in an element *e*.
- To demonstrate the process of constructing the basis functions, we use a general approach even if the answer is rather obvious. For linear elements, we write:

$$u^e(x) = a_0^e + a_1^e x$$

- We want to compute the parameters a_0^e and a_1^e in terms of the nodal values $u^e(x_1^e)$ and $u^e(x_2^e)$.

Linear one-dimensional elements



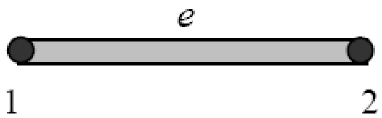
$$u^e(x) = \underbrace{[1 \quad x]}_{p(x)} \underbrace{\begin{Bmatrix} a_0^e \\ a_1^e \end{Bmatrix}}_{a^e} \equiv p(x) a^e$$

$$\begin{aligned} u^e(x_1^e) &= a_0^e + a_1^e x_1^e \\ u^e(x_2^e) &= a_0^e + a_1^e x_2^e \end{aligned} \longrightarrow \{d^e\} = \begin{Bmatrix} u^e(x_1^e) \\ u^e(x_2^e) \end{Bmatrix} = \underbrace{\begin{bmatrix} 1 & x_1^e \\ 1 & x_2^e \end{bmatrix}}_{M^e} \underbrace{\begin{Bmatrix} a_0^e \\ a_1^e \end{Bmatrix}}_{a^e}$$

- Combining the above eqs. gives us the element shape functions

$$u^e(x) = p(x) a^e = \underbrace{p(x)(M^e)^{-1}}_{N^e(x)} d^e$$

Element shape functions



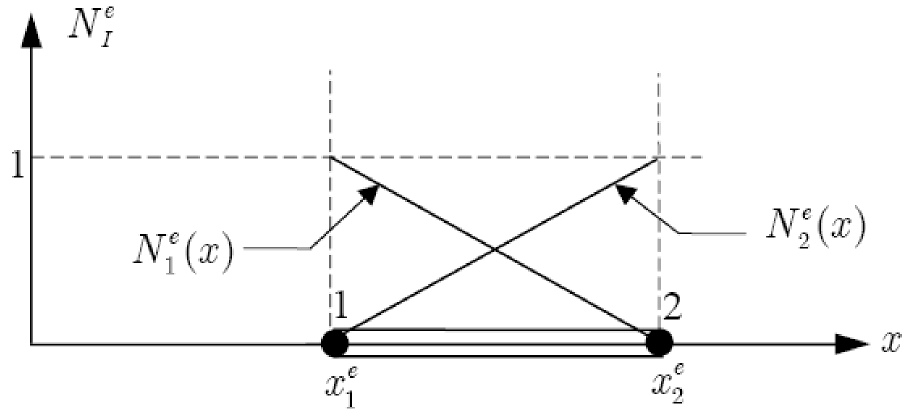
$$(M^e)^{-1} = \underbrace{\begin{bmatrix} 1 & x_1^e \\ 1 & x_2^e \end{bmatrix}}_{M^e}^{-1} = \frac{1}{L^e}\begin{bmatrix} x_2^e & -x_1^e \\ -1 & 1 \end{bmatrix}$$

- The element shape function matrix can now be computed as:

$$N^e = [N_1^e \quad N_2^e] = p(x)(M^e)^{-1} = [1 \quad x]\frac{1}{L^e}\begin{bmatrix} x_2^e & -x_1^e \\ -1 & 1 \end{bmatrix} \Rightarrow$$

$$N^e = [N_1^e \quad N_2^e] = p(x)(M^e)^{-1} = \frac{1}{L^e}\begin{bmatrix} x_2^e - x & x - x_1^e \end{bmatrix} \Rightarrow \quad \begin{aligned} N_1^e &= \frac{1}{L^e}(x_2^e - x) \\ N_2^e &= \frac{1}{L^e}(x - x_1^e) \end{aligned}$$

Two-node linear element: Shape functions



$$N_1^e = \frac{1}{L^e}(x_2^e - x)$$

$$N_2^e = \frac{1}{L^e}(x - x_1^e)$$

- Verify that (as desired) the following holds: $N_i^e(x_j^e) = \delta_{ij}$

Note:

$$u^e(x_1^e) = N_1^e(x_1^e)d_1^e + N_2^e(x_1^e)d_2^e = 1d_1^e + 0d_2^e = d_1^e$$

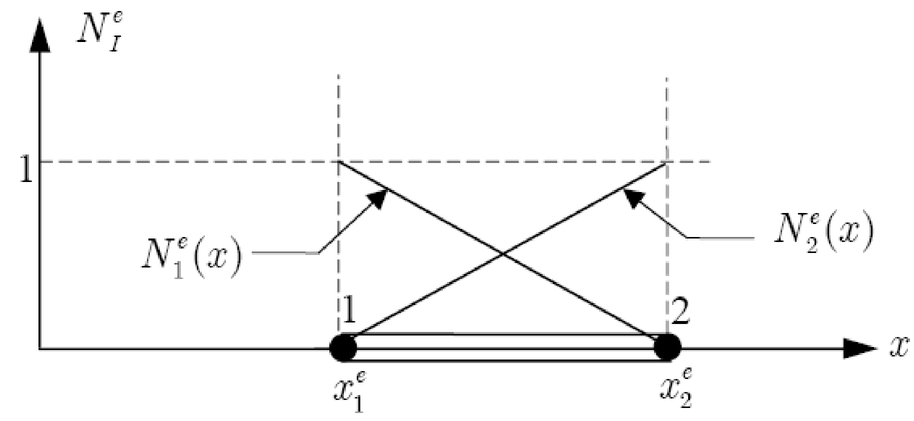
$$u^e(x_2^e) = N_1^e(x_2^e)d_1^e + N_2^e(x_2^e)d_2^e = 0d_1^e + 1d_2^e = d_2^e$$

- The interpolation property is now summarized as:

$$u^e(x) = N^e(x)d^e = [N_1^e \quad N_2^e]\begin{Bmatrix} d_1^e \\ d_2^e \end{Bmatrix} = N_1^e d_1^e + N_2^e d_2^e \Rightarrow$$

$$u^e(x) = N_1^e d_1^e + N_2^e d_2^e = \sum_{i=1}^{2} N_i^e d_i^e$$

Two-node linear element: Shape functions



$$N_1^e = \frac{1}{L^e}(x_2^e - x)$$

$$N_2^e = \frac{1}{L^e}(x - x_1^e)$$

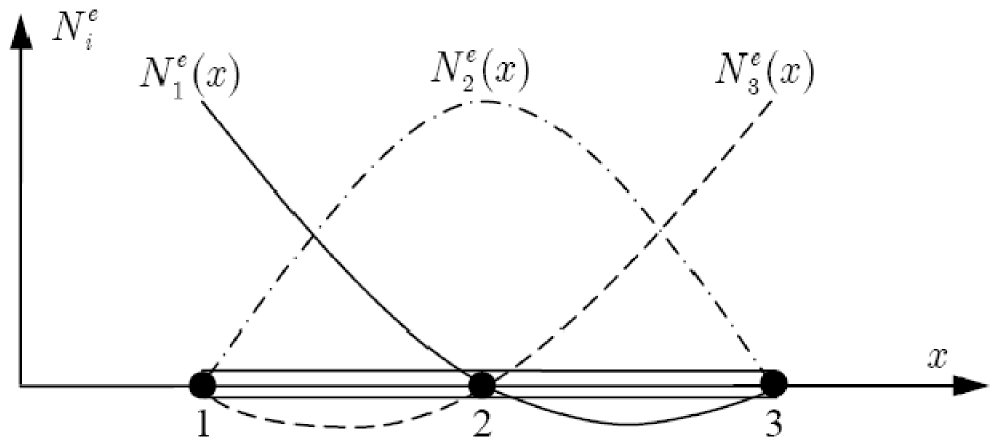
- The derivative of $u^e(x)$ can now be approximated as:

$$\frac{du^e(x)}{dx} = \frac{dN^e(x)}{dx} d^e = \frac{dN_1^e}{dx} d_1^e + \frac{dN_2^e}{dx} d_2^e$$

From which we can write:

$$\frac{du^e(x)}{dx} = \underbrace{\left[-\frac{1}{L^e} \quad \frac{1}{L^e} \right]}_{B^e} \begin{Bmatrix} d_1^e \\ d_2^e \end{Bmatrix} = B^e \left\{ d^e \right\}$$

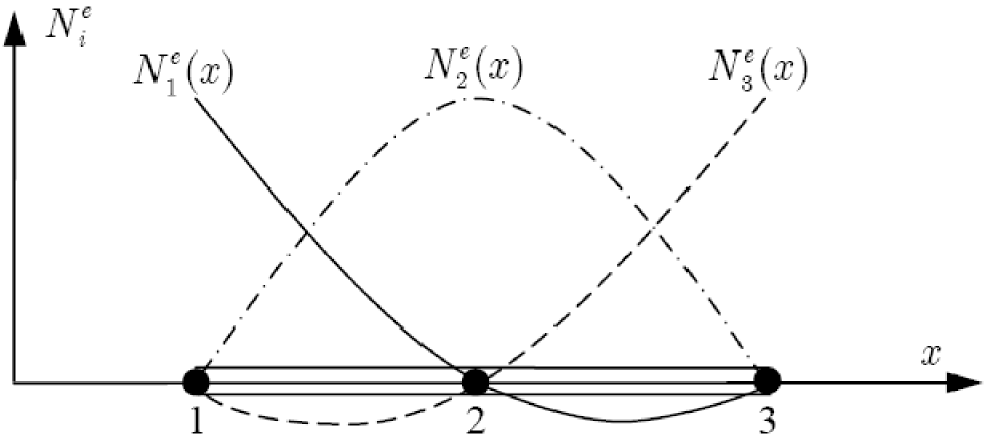
Three node element: quadratic interpolation



- We can repeat the same process as for the 2-node element to derive the shape functions for the 3-node element.
- However, we can use Lagrange interpolants to find out directly the shape functions using simple arguments.
- Lets derive N_1^e . It's a quadratic function, and it needs to be zero at nodes x_2^e and x_3^e . So it needs to be of the form: $\sim (x - x_2^e)(x - x_3^e)$. It also needs to take a value of 1 at node x_1^e . We finally arrive at:

$$N_1^e = \frac{(x - x_2^e)(x - x_3^e)}{(x_1^e - x_2^e)(x_1^e - x_3^e)}$$

3-node (quadratic) and 4-node (cubic) elements

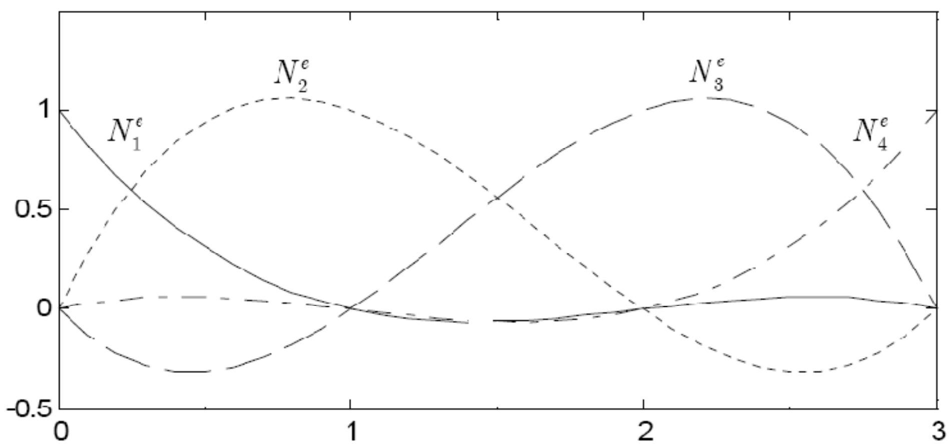


$$N_1^e = \frac{(x - x_2^e)(x - x_3^e)}{(x_1^e - x_2^e)(x_1^e - x_3^e)}$$

$$N_2^e = \frac{(x - x_1^e)(x - x_3^e)}{(x_2^e - x_1^e)(x_2^e - x_3^e)}$$

$$N_3^e = \frac{(x - x_1^e)(x - x_2^e)}{(x_3^e - x_1^e)(x_3^e - x_2^e)}$$

- We can generalize this process to higher order interpolation. For cubic interpolation, we need 4 nodes.



$$N_1^e = \frac{(x - x_2^e)(x - x_3^e)(x - x_4^e)}{(x_1^e - x_2^e)(x_1^e - x_3^e)(x_1^e - x_4^e)}$$

$$N_2^e = \frac{(x - x_1^e)(x - x_3^e)(x - x_4^e)}{(x_2^e - x_1^e)(x_2^e - x_3^e)(x_2^e - x_4^e)}$$

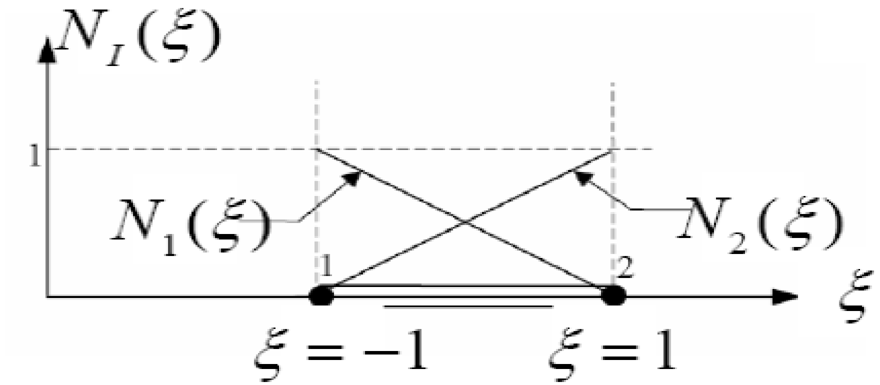
$$N_3^e = \frac{(x - x_1^e)(x - x_2^e)(x - x_4^e)}{(x_3^e - x_1^e)(x_3^e - x_2^e)(x_3^e - x_4^e)}$$

$$N_4^e = \frac{(x - x_1^e)(x - x_2^e)(x - x_3^e)}{(x_4^e - x_1^e)(x_4^e - x_2^e)(x_4^e - x_3^e)}$$

Master elements in natural coordinate ξ

- It is more common to define the shape functions on a master element in coordinate ξ with $-1 \le \xi \le 1$.
- For linear elements, we can then easily map to real coordinate *x* for each element *e* with the isoparametric transformation (the same interpolation we used for u(x)):

$$x = x_1^e + \frac{1}{2}(x_2^e - x_1^e)(1+\xi) = \underbrace{\frac{1-\xi}{2}}_{N_1^e} x_1^e + \underbrace{\frac{1+\xi}{2}}_{N_2^e} x_2^e = \left[N_1^e \;\; N_2^e \right] \begin{Bmatrix} x_1^e \\ x_2^e \end{Bmatrix} = \left[N^e \right] \left\{ x^e \right\}$$

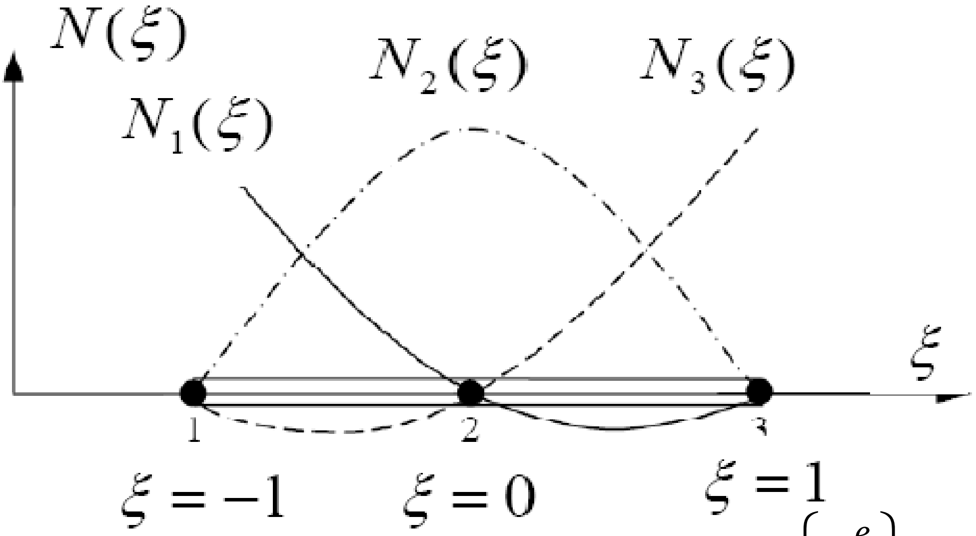


$$N_1^e = \frac{1}{2}(1-\xi)$$

$$N_2^e = \frac{1}{2}(1+\xi)$$

Master elements in natural coordinate ξ

- Similarly a quadratic element in natural coordinate has nodes at $\xi = 0, \pm 1$.



$$N_1^e = \frac{1}{2}\xi(\xi - 1)$$

$$N_2^e = 1 - \xi^2$$

$$N_3^e = \frac{1}{2}(1 + \xi)\xi$$

$$x = \begin{bmatrix} N_1^e & N_2^e & N_3^e \end{bmatrix} \begin{Bmatrix} x_1^e \\ x_2^e \\ x_3^e \end{Bmatrix} = \begin{bmatrix} N^e \end{bmatrix} \begin{Bmatrix} x^e \end{Bmatrix}$$

Global and element shape functions

- In an earlier lecture, we introduced the transformation from elemental to global degrees of freedom:

$$\{d^e\} = [L^e]\{d\}$$

- By assembling the local element approximations, we can compute our global finite element approximation:

$$u^h(x) = \sum_e [N^e]\{d^e\} = \underbrace{\sum_e [N^e][L^e]}_{N}\{d\}$$

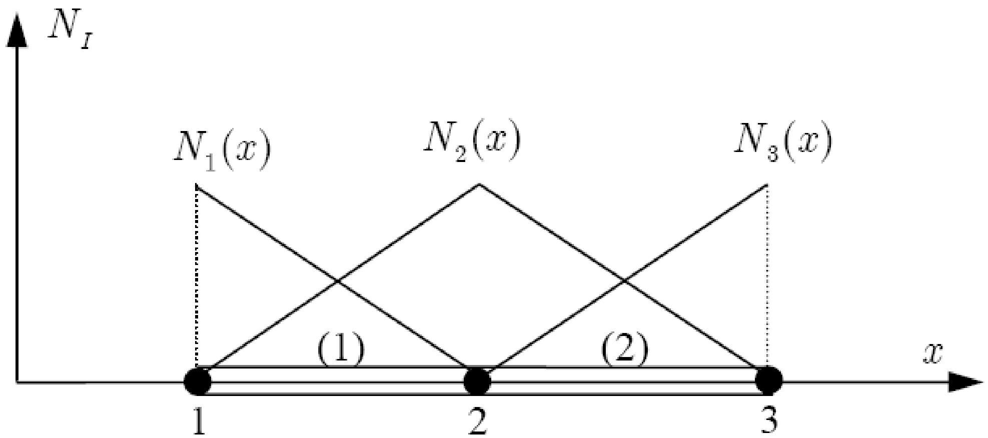
← Global shape functions (row matrix)

- *N* are our global shape functions. We can rewrite the global approximation as:

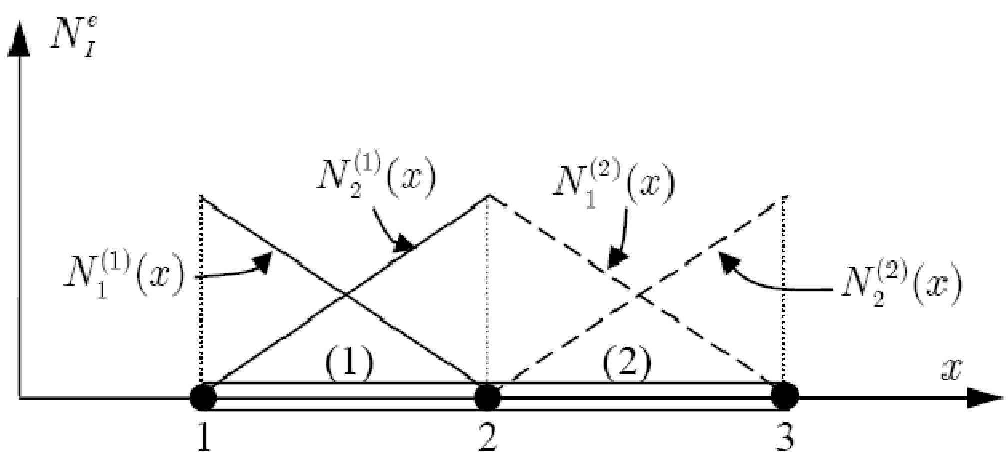
$$u^h(x) = [N]\{d\} = \sum_{i=1}^{N_{nodes}} N_i d_i$$

← Number of nodes in the mesh

Linear global and element shape functions



Global shape functions



Element shape functions

$$u^h(x) = [N]\{d\} = \sum_{i=1}^{N_{nodes}} N_i d_i$$

$$w^h(x) = [N]\{w\} = \sum_{i=1}^{N_{nodes}} N_i w_i$$

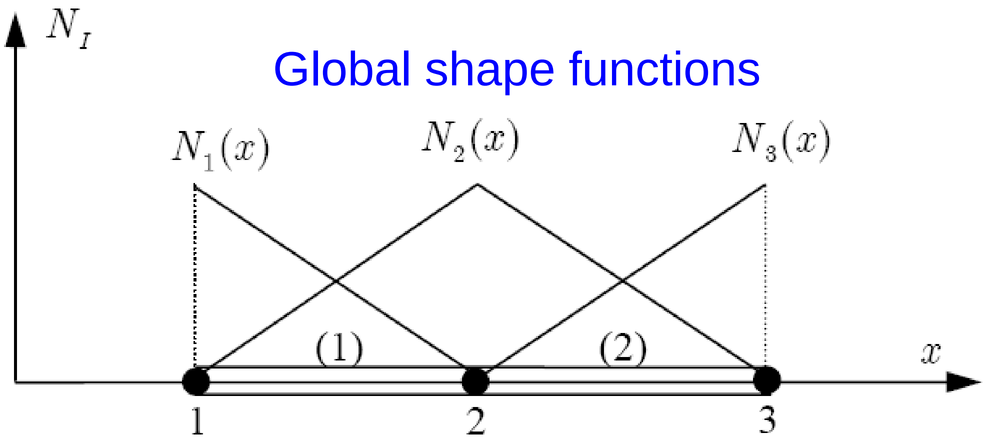
- The matrix [N] of global shape functions is a row matrix.

$$[N] = \sum_e [N^e][L^e]$$

- In a column form:

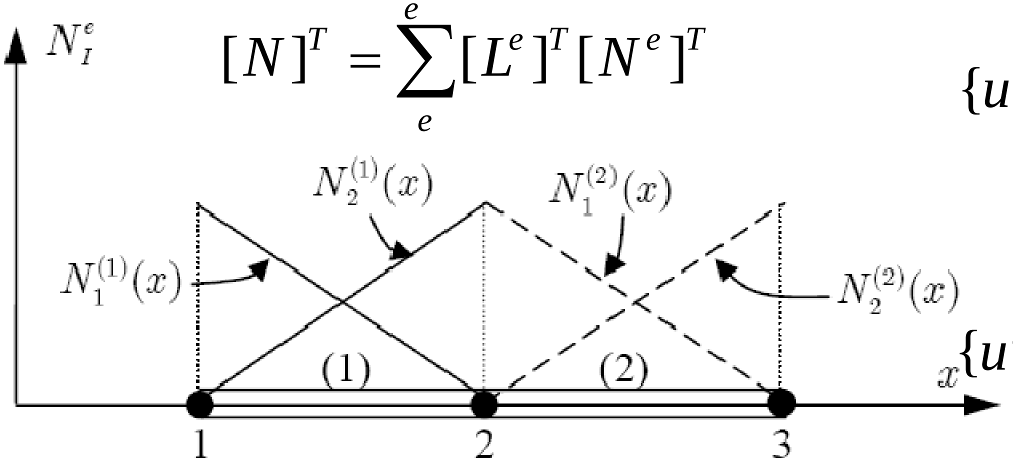
$$[N]^T = \sum_e [L^e]^T [N^e]^T$$

Linear global and element shape functions



$$[N] = \sum_e [N^e][L^e]$$

$$[N]^T = \sum_e [L^e]^T [N^e]^T$$



Element shape functions

For the mesh of the 2 linear elements shown, we can write:

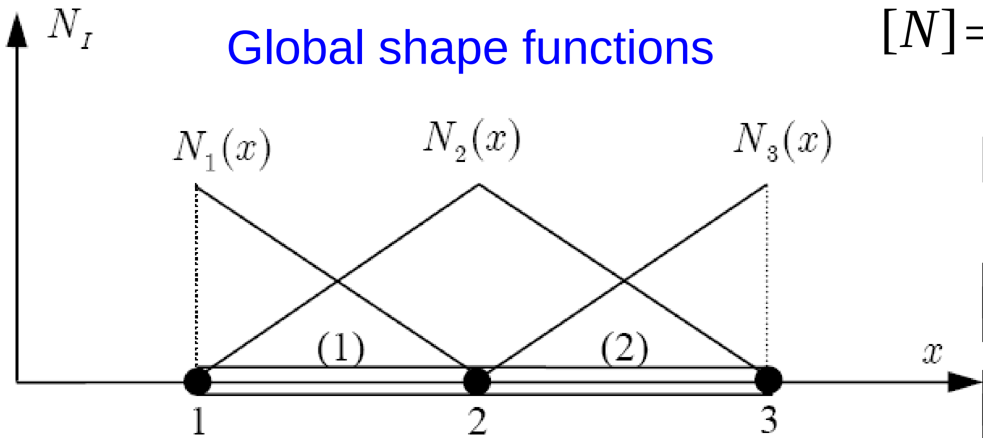
$$[N] = [N^1][L^1] + [N^2][L^2]$$

- The mappings $[L^1], [L^2]$ are:

$$\{u^1\} = \begin{Bmatrix} u_1^1 \\ u_2^1 \end{Bmatrix} = \begin{Bmatrix} u_1 \\ u_2 \end{Bmatrix} = \underbrace{\begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \end{bmatrix}}_{L^1} \begin{Bmatrix} u_1 \\ u_2 \\ u_3 \end{Bmatrix}$$

$$\{u^2\} = \begin{Bmatrix} u_1^2 \\ u_2^2 \end{Bmatrix} = \begin{Bmatrix} u_2 \\ u_3 \end{Bmatrix} = \underbrace{\begin{bmatrix} 0 & 1 & 0 \\ 0 & 0 & 1 \end{bmatrix}}_{L^2} \begin{Bmatrix} u_1 \\ u_2 \\ u_3 \end{Bmatrix}$$

Linear global and element shape functions



$$[N]=[N^1][L^1]+[N^2][L^2]=$$

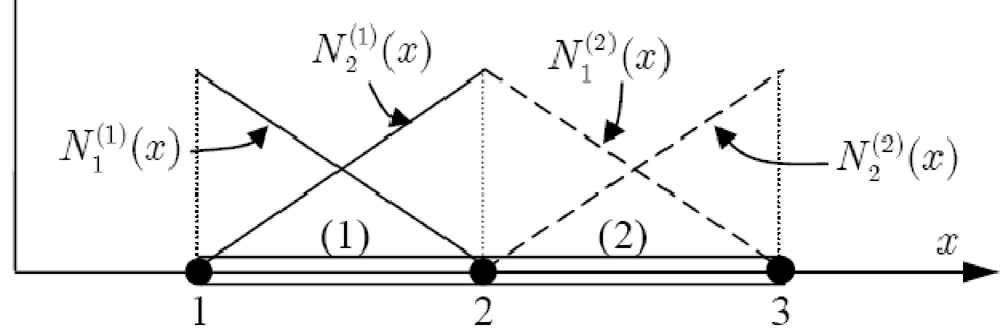
$$[N_1^1 \quad N_2^1]\begin{bmatrix}1 & 0 & 0\\ 0 & 1 & 0\end{bmatrix}+[N_1^2 \quad N_2^2]\begin{bmatrix}0 & 1 & 0\\ 0 & 0 & 1\end{bmatrix}=$$

$$\begin{bmatrix}N_1^1 & N_2^1 & 0\end{bmatrix}+\begin{bmatrix}0 & N_1^2 & N_2^2\end{bmatrix}=$$

$$\begin{bmatrix}N_1^1 & N_2^1+N_1^2 & N_2^2\end{bmatrix}$$

$$[N]=\sum_e [N^e][L^e]$$

$$[N]^T=\sum_e [L^e]^T[N^e]^T$$



Element shape functions

$$N^1 = N_1^1$$

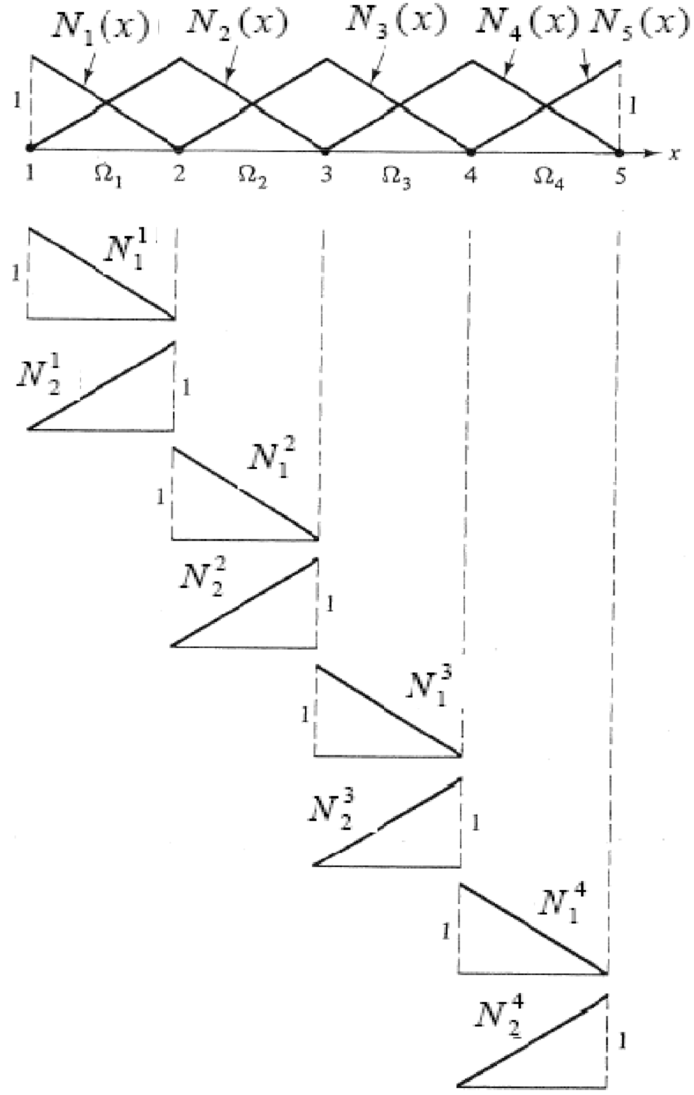
$$N^2 = N_2^1 + N_1^2$$

$$N^2 = N_2^2$$

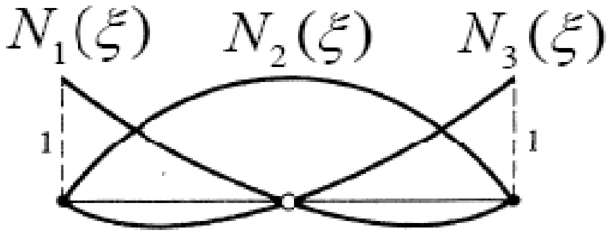
C^0 continuous → Global shape functions

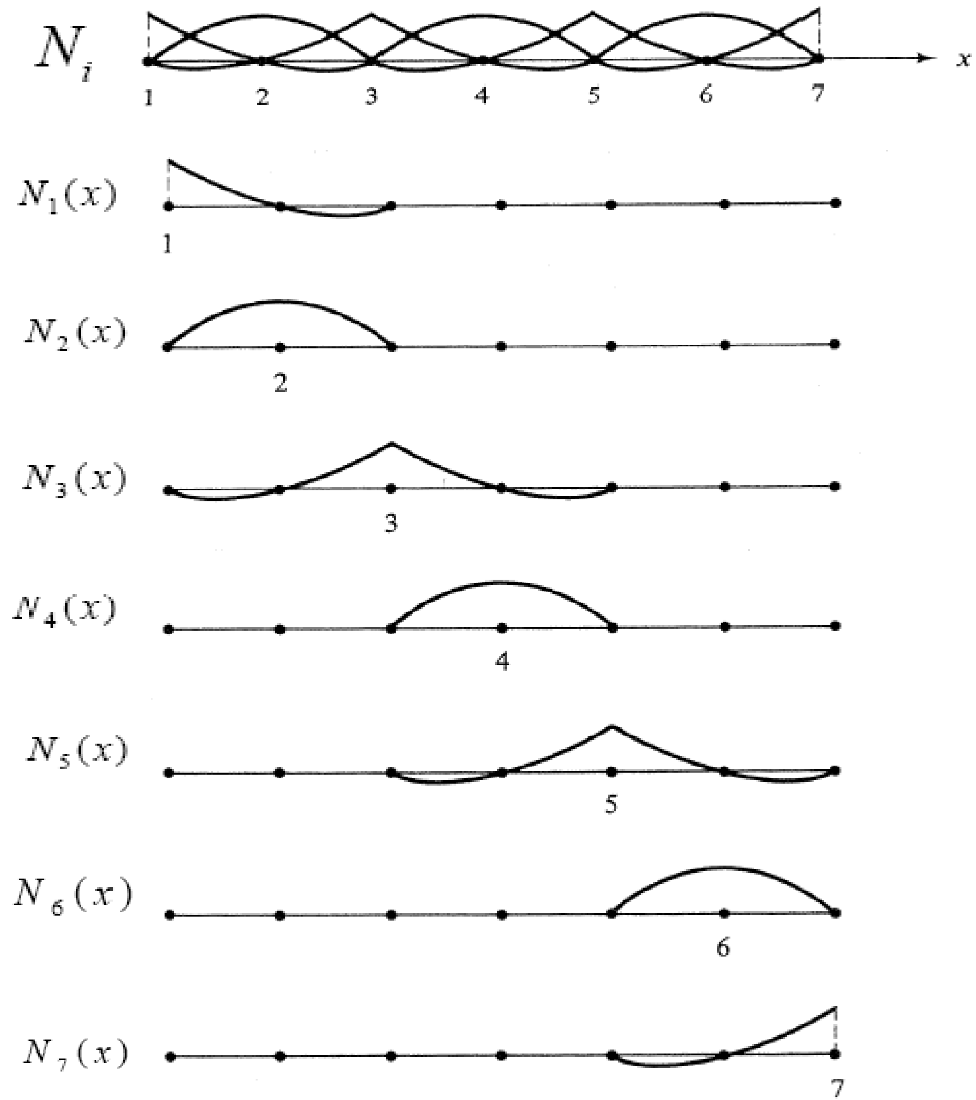
Linear global and element shape functions

Piecewise-linear basis functions N_i for a 4-element mesh generated by linear shape functions, N_i^e defined over each element.



Quadratic global and element shape functions





A mesh consisting of three quadratic elements and the global basis functions generated by these elements.

Gauss quadrature: Numerical integration

- In FEM, we need to compute element stiffness and load contributions. These are integrals that generally cannot be computed analytically.
- We need a numerical approach to computing integrals over elements in the form: $I = \int_{\Omega_e} f(x)dx$
- Let us assume a linear element from x_1^e to x_2^e. With the transformation $x = x_1^e + \frac{1}{2}(x_2^e - x_1^e)(1+\xi) = x_1^e + \frac{L^e}{2}(1+\xi)$, we can map the integral *I* to natural coordinates as:

$$I = \int_{-1}^{1} f(x(\xi))Jd\xi = J\int_{-1}^{1} \overline{f}(\xi)d\xi,\ J = \frac{dx}{d\xi} = \frac{L^e}{2}$$

- We need a numerical approximation to integrate in ξ .

Jacobian of the transformation from x to ξ

- We just showed that for a linear element the Jacobian of the transformation is derived as:

$$x = \underbrace{\frac{1-\xi}{2}}_{N_1^e} x_1^e + \underbrace{\frac{1+\xi}{2}}_{N_2^e} x_2^e = \left[N_1^e \ N_2^e \right] \begin{Bmatrix} x_1^e \\ x_2^e \end{Bmatrix} = \left[N^e \right] \left\{ x^e \right\} \Rightarrow J \equiv \frac{dx}{d\xi} = \left[\frac{dN_1^e}{d\xi} \ \frac{dN_2^e}{d\xi} \right] \begin{Bmatrix} x_1^e \\ x_2^e \end{Bmatrix} = \left[-\frac{1}{2} \ \frac{1}{2} \right] \begin{Bmatrix} x_1^e \\ x_2^e \end{Bmatrix} = \frac{x_2^e - x_1^e}{2} = \frac{L^e}{2}$$

- Similarly for higher-order elements (*n=3,4,..)* using the isoparametric transformation, we derive that:

$$x = \left[N_1^e \ N_2^e \ldots . N_n^e \right] \begin{Bmatrix} x_1^e \\ x_2^e \\ \ldots \\ x_n^e \end{Bmatrix} \Rightarrow J \equiv \frac{dx}{d\xi} = \left[\frac{dN_1^e}{d\xi} \ \frac{dN_2^e}{d\xi} \ldots . \frac{dN_n^e}{d\xi} \right] \begin{Bmatrix} x_1^e \\ x_2^e \\ \ldots \\ x_n^e \end{Bmatrix} = \left[\frac{dN^e}{d\xi} \right] \begin{Bmatrix} x_1^e \\ x_2^e \\ \ldots \\ x_n^e \end{Bmatrix}$$

- Please note that for higher-order elements, the Jacobian of the transformation generally changes from Gauss point to Gauss point.

Gauss quadrature: Numerical integration

$$\bar{I} = \int_{-1}^{1} \bar{f}(\xi) d\xi = \sum_{i=1}^{N_{Gauss\ Points}} W_i \bar{f}(\xi_i) = W_1 \bar{f}(\xi_1) + \ldots + W_{N_{Gauss\ Points}} \bar{f}(\xi_{N_{Gauss\ Points}})$$

- W_i are appropriate weights and ξ_i are appropriate evaluation points. The selection of these is optimal in the sense that the highest possible polynomial can be integrated exactly.
- An $N_{Gauss\ Points}$ Gauss integration formula can integrate exactly a polynomial of order $2N_{Gauss\ Points} - 1$.
- So if you have to integrate a polynomial of order *p*, you need to use

$$N_{Gauss\ Points} \geq \frac{p+1}{2}$$

- ✓ For *p=2*, you need a minimum of 2 Gauss points
- ✓ For *p=3*, you need a minimum of 2 Gauss points

Gauss quadrature

$N_{Gauss\ Points}$	ξ	W_i
1	*0.0*	2.0
2	$\pm 1/\sqrt{3}$	1.0
3	±0.7745966692 0	*0.5555555556* *0.8888888889*
4	±0.8611363116 ±0.3399810436	*0.3478548451* *0.6521451549*
5	±0.9061798459 ±0.5384693101 *0.0*	*0.2369268851* *0.4786286705* *0.5688888889*

An example of Gauss quadrature

- Let us compute the integral:

$$I = \int_{-1}^{1} (\xi^4 + 2\xi^2) d\xi \; (= \frac{2}{5} + \frac{4}{3} = \frac{26}{15} = 1.73333)$$

- We need: $N_{Gauss\ Points} \geq \frac{p+1}{2} = \frac{4+1}{2} = 2.5 \Rightarrow N_{Gauss\ Points} = 3.$

- Using Gauss quadrature:

$$I = W_1 f(\xi_1) + W_2 f(\xi_2) + W_3 f(\xi_3),$$

$$\xi_1 = 0.7745966692, \xi_2 = 0, \xi_3 = -0.7745966692 \Rightarrow f(\xi_2) = 0, f(\xi_1) = f(\xi_3) = 1.56$$

$$W_1 = 0.5555555556, W_2 = 0.8888888889, W_3 = 0.5555555556$$

$$I = 0.5555555556_1 \times 1.56 + 0.8888888889 \times 0 + 0.5555555556 \times 1.56 \Rightarrow I = 1.73333!$$