第3章 一维问题

3.4 一维问题的近似函数

Approximations in 1D Problems

[Fish]-4.1 ~ 4.5

3.4 近似函数 Accuracy and Convergence

- Example 3.2: How to further improve the solution ?
  - Higher order functions

$$u(x)=\alpha_0+\alpha_1 x+\alpha_2 x^2+\alpha_3 x^3$$

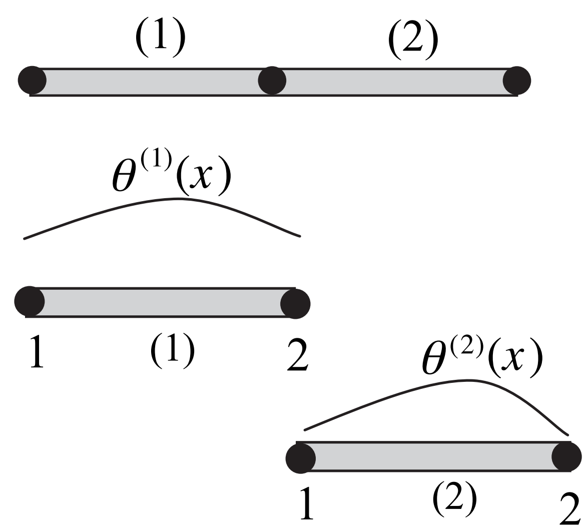
$$w(x)=\beta_0+\beta_1 x+\beta_2 x^2+\beta_3 x^3$$

  - Finite element method — constructed by subdividing the domain of the problem into elements and constructing functions within each element
- Convergence
  - The accuracy of a correctly developed FEM improves with mesh refinement
  - Element size *h* decreases, the solution tends to the ***correct solution***
- Convergence criteria
  - **Continuity** — The trail solutions and test functions are sufficiently smooth so that all integrals in the weak form are well defined
  - **Completeness** — As the element size approach zero, the trial solutions and test functions and their derivatives up to the highest order derivative appearing in the weak form be capable of assuming constant values
    - For elasticity — FEM can present rigid body motion and constraint strain states exactly

3.4 近似函数 Accuracy and Convergence

- Notation and nomenclature
  - ✧ $\theta^h(x)$ — The global finite element approximation
  - ✧ $\theta^e(x)$ — The approximation for a particular element e, which is nonzero only in element e
  - ✧ Numerical superscripts refer to a specific element: $\theta^{(1)}(x)$
  - ✧ For nodal variables, a subscript denotes the global node number; x_2
    For element-related nodal variables, local node numbers are used: $x_2^{(1)}$
- Construction of approximation

$$\theta^e(x) = \alpha_0^e + \alpha_1^e x + \alpha_2^e x^2 + \alpha_3^e x^3 + \cdots$$



  - ✧ C^0 continuity requirement

$$\theta^{(1)}(x_2^{(1)}) = \theta^{(2)}(x_1^{(2)}) \Longrightarrow \theta_2^{(1)} = \theta_1^{(2)}$$

  - ✧ Completeness requirement

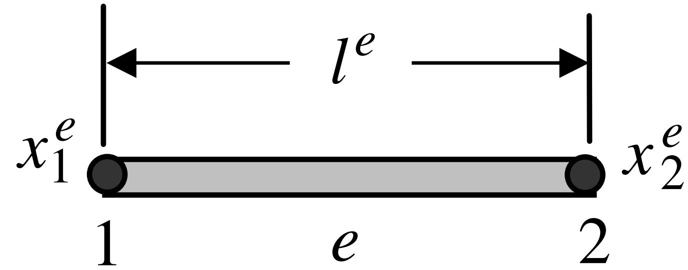
$$\theta^e(x) = \alpha_0^e + \alpha_1^e x \qquad \theta^e(x) = \alpha_0^e + \alpha_1^e x + \alpha_2^e x^2$$

$$\theta^e(x) = \alpha_0^e + \alpha_1^e x + \alpha_1^e x^4$$

?

$$\theta^e(x) = \alpha_0^e + \alpha_1^e x^2$$

3.4 近似函数　Two-node linear element



✧ Completeness requirement

$$\theta^e(x) = \alpha_0^e + \alpha_1^e x = \underbrace{\begin{bmatrix} 1 & x \end{bmatrix}}_{\boldsymbol{p}(x)} \underbrace{\begin{bmatrix} \alpha_0^e \\ \alpha_1^e \end{bmatrix}}_{\boldsymbol{a}^e} = \boldsymbol{p}(x)\boldsymbol{a}^e$$

$$\theta^e(x) = \alpha_0^e + \alpha_1^e x + \alpha_2^e x^2 \quad ?$$

✧ C^0 continuity requirement

$$\theta^e(x_1^e) \equiv \theta_1^e = \alpha_0^e + \alpha_1^e x_1^e$$
$$\theta^e(x_2^e) \equiv \theta_2^e = \alpha_0^e + \alpha_1^e x_2^e$$

$$\Rightarrow \underbrace{\begin{bmatrix} \theta_1^e \\ \theta_2^e \end{bmatrix}}_{\boldsymbol{d}^e} = \underbrace{\begin{bmatrix} 1 & x_1^e \\ 1 & x_2^e \end{bmatrix}}_{\boldsymbol{M}^e} \underbrace{\begin{bmatrix} \alpha_0^e \\ \alpha_1^e \end{bmatrix}}_{\boldsymbol{a}^e}$$

$$\boldsymbol{a}^e = \left(\boldsymbol{M}^e\right)^{-1} \boldsymbol{d}^e$$

$$\theta^e(x) = \boldsymbol{p}(x)\left(\boldsymbol{M}^e\right)^{-1} \boldsymbol{d}^e = \boldsymbol{N}^e(x)\boldsymbol{d}^e$$

$$\boldsymbol{N}^e(x) = \boldsymbol{p}(x)\left(\boldsymbol{M}^e\right)^{-1}$$ — Element shape function matrix

3.4 近似函数 Two-node linear element

$$\boldsymbol{M}^e=\begin{bmatrix}1 & x_1^e\\ 1 & x_2^e\end{bmatrix}\Longrightarrow\left(\boldsymbol{M}^e\right)^{-1}=\frac{1}{x_2^e-x_1^e}\begin{bmatrix}x_2^e & -x_1^e\\ -1 & 1\end{bmatrix}=\frac{1}{l^e}\begin{bmatrix}x_2^e & -x_1^e\\ -1 & 1\end{bmatrix}$$

$$\boldsymbol{N}^e(x)=[\,N_1^e\quad N_2^e\,]=\boldsymbol{p}(x)\left(\boldsymbol{M}^e\right)^{-1}=\frac{1}{l^e}[\,x_2^e-x\quad x-x_1^e\,]$$

$N_1^e(x)=\dfrac{1}{l^e}(x_2^e-x)$ — Element shape functions corresponding to node 1

$N_2^e(x)=\dfrac{1}{l^e}(x-x_1^e)$ — Element shape functions corresponding to node 2

$$N_1^e(x_1^e)=1,\quad N_1^e(x_2^e)=0$$
$$N_2^e(x_1^e)=0,\quad N_2^e(x_2^e)=1$$

$$N_I^e(x_J^e)=\delta_{IJ}$$

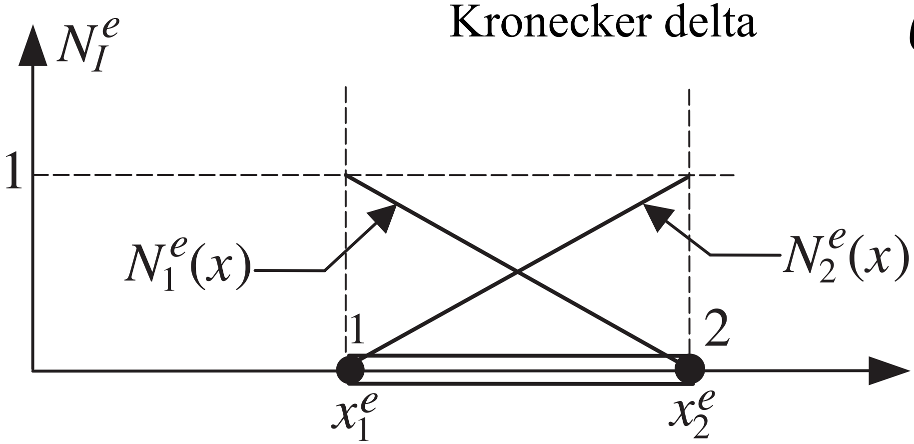
Kronecker delta

$$\theta^e(x)=\boldsymbol{N}^e(x)\boldsymbol{d}^e=\sum_{I=1}^{n_{\text{en}}}N_I^e(x)\theta_I^e$$

$$\Big\downarrow\; x=x_J^e$$

$$\theta^e(x_J^e)=\sum_{I=1}^{2}N_I^e(x_J^e)\theta_I^e=\sum_{I=1}^{2}\delta_{IJ}\theta_I^e=\theta_J^e$$

Interpolation property



$$\sum_{I=1}^{2}N_I^e(x)=1$$

Partition of Unity

Computational Dynamics 计算动力学实验室

3.4 近似函数 Two-node linear element

$$\theta^e(x) = \boldsymbol{N}^e(x)\boldsymbol{d}^e = \sum_{I=1}^{n_{\mathrm{en}}} N_I^e(x)\theta_I^e \qquad \boldsymbol{N}^e = \frac{1}{l^e}\begin{bmatrix} x_2^e - x & x - x_1^e \end{bmatrix}$$

$$\begin{aligned}
\frac{\mathrm{d}\theta^e}{\mathrm{d}x} &= \frac{\mathrm{d}\boldsymbol{N}^e}{\mathrm{d}x}\boldsymbol{d}^e \\
&= \frac{\mathrm{d}N_1^e}{\mathrm{d}x}\theta_1^e + \frac{\mathrm{d}N_2^e}{\mathrm{d}x}\theta_2^e \\
&= \begin{bmatrix} \dfrac{\mathrm{d}N_1^e}{\mathrm{d}x} & \dfrac{\mathrm{d}N_2^e}{\mathrm{d}x} \end{bmatrix}\begin{bmatrix} \theta_1^e \\ \theta_2^e \end{bmatrix} \\
&= \boldsymbol{B}^e\boldsymbol{d}^e
\end{aligned}$$

$$\boldsymbol{B}^e = \begin{bmatrix} \dfrac{\mathrm{d}N_1^e}{\mathrm{d}x} & \dfrac{\mathrm{d}N_2^e}{\mathrm{d}x} \end{bmatrix} = \frac{1}{l^e}\begin{bmatrix} -1 & 1 \end{bmatrix}$$

Two-node linear element

- Weight (test) functions ?
  - Can be different from that used for the trail solutions
  - Galerkin method — Use the same approximation for the weight functions and trial solutions

$$\theta^e(x) = \boldsymbol{N}^e(x)\boldsymbol{d}^e \qquad w^e(x) = \boldsymbol{N}^e(x)\boldsymbol{w}^e \qquad \frac{\mathrm{d}w^e(x)}{\mathrm{d}x} = \boldsymbol{B}^e(x)\boldsymbol{w}^e$$

- Global approximation and continuity

$$\theta^h(x) = \sum_{e=1}^{n_{\mathrm{el}}} \boldsymbol{N}^e(x)\boldsymbol{d}^e \Leftarrow \boldsymbol{d}^e = \boldsymbol{L}^e\boldsymbol{d}$$

$$= \left(\sum_{e=1}^{n_{\mathrm{el}}} \boldsymbol{N}^e\boldsymbol{L}^e\right)\boldsymbol{d} = \boldsymbol{N}\boldsymbol{d} = \sum_{I=1}^{n_{\mathrm{np}}} N_I d_I$$

$$w^h(x) = \sum_{e=1}^{n_{\mathrm{el}}} \boldsymbol{N}^e(x)\boldsymbol{w}^e \qquad \boldsymbol{N} = \sum_{e=1}^{n_{\mathrm{el}}} \boldsymbol{N}^e\boldsymbol{L}^e \text{ — Global shape function matrix}$$

$$= \left(\sum_{e=1}^{n_{\mathrm{el}}} \boldsymbol{N}^e\boldsymbol{L}^e\right)\boldsymbol{w} = \boldsymbol{N}\boldsymbol{w} = \sum_{e=1}^{n_{\mathrm{np}}} N_I w_I$$

3.4 近似函数 Two-node linear element

$$\boldsymbol{N} = \sum_{e=1}^{n_{\text{el}}} \boldsymbol{N}^e \boldsymbol{L}^e$$ — Global shape function matrix

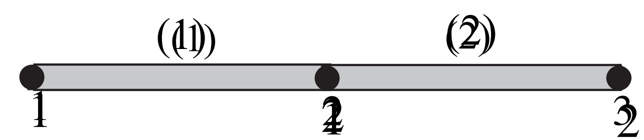
$$\boldsymbol{d}^{(1)} = \begin{bmatrix} \theta_1^{(1)} \\ \theta_2^{(1)} \end{bmatrix} = \begin{bmatrix} \theta_1 \\ \theta_2 \end{bmatrix} = \underbrace{\begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \end{bmatrix}}_{\boldsymbol{L}^{(1)}} \begin{bmatrix} \theta_1 \\ \theta_2 \\ \theta_3 \end{bmatrix} = \boldsymbol{L}^{(1)} \boldsymbol{d}$$

$$\boldsymbol{d}^{(2)} = \begin{bmatrix} \theta_1^{(2)} \\ \theta_2^{(2)} \end{bmatrix} = \begin{bmatrix} \theta_2 \\ \theta_3 \end{bmatrix} = \underbrace{\begin{bmatrix} 0 & 1 & 0 \\ 0 & 0 & 1 \end{bmatrix}}_{\boldsymbol{L}^{(2)}} \begin{bmatrix} \theta_1 \\ \theta_2 \\ \theta_3 \end{bmatrix} = \boldsymbol{L}^{(2)} \boldsymbol{d}$$

$$\boldsymbol{N} = \boldsymbol{N}^{(1)} \boldsymbol{L}^{(1)} + \boldsymbol{N}^{(2)} \boldsymbol{L}^{(2)} \Longleftarrow \boldsymbol{N}^{(1)} = [\; N_1^{(1)} \quad N_2^{(1)} \;] \qquad \boldsymbol{N}^{(2)} = [\; N_1^{(2)} \quad N_2^{(2)} \;]$$

$$= [\; N_1^{(1)} \quad N_2^{(1)} + N_1^{(2)} \quad N_2^{(2)} \;]$$

$$= [\; N_1 \quad N_2 \quad N_3 \;]$$



3.4 近似函数 Two-node linear element

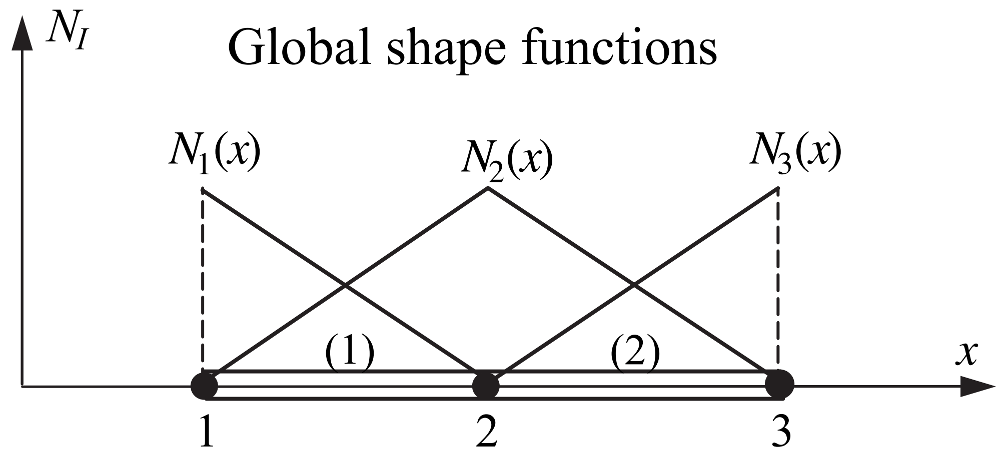
$$N_1 = N_1^{(1)}$$

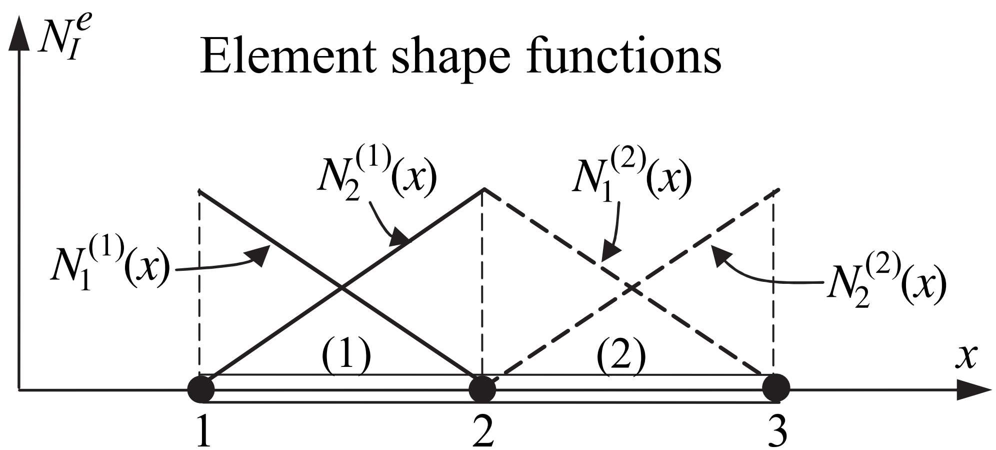
$$N_2 = N_2^{(1)} + N_1^{(2)}$$

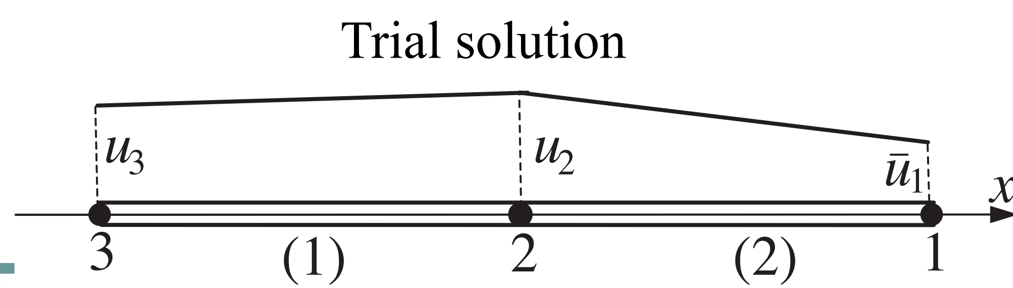
$$N_3 = N_2^{(2)}$$

Global shape functions

- Identical to the element shape functions over an element domain
- C^0 continuity of approximations are guaranteed as any combination of C^0 functions must be C^0
- The integrals in the weak form are finite due to $N_I \in H^1$

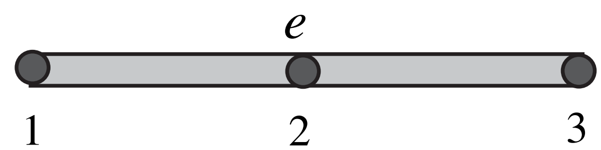






3.4 近似函数 Quadratic element

- Completeness requirement



$$\theta^e(x) = \alpha_0^e + \alpha_1^e x + \alpha_2^e x^2$$

$$= \underbrace{\begin{bmatrix} 1 & x & x^2 \end{bmatrix}}_{\boldsymbol{p}(x)} \underbrace{\begin{bmatrix} \alpha_0^e \\ \alpha_1^e \\ \alpha_2^e \end{bmatrix}}_{\boldsymbol{a}^e} = \boldsymbol{p}(x)\boldsymbol{a}^e$$

- C^0 continuity requirement

$$\theta_1^e = \alpha_0^e + \alpha_1^e x_1^e + \alpha_2^e (x_1^e)^2$$
$$\theta_2^e = \alpha_0^e + \alpha_1^e x_2^e + \alpha_2^e (x_2^e)^2$$
$$\theta_3^e = \alpha_0^e + \alpha_1^e x_3^e + \alpha_2^e (x_3^e)^2$$

$$\Rightarrow \underbrace{\begin{bmatrix} \theta_1^e \\ \theta_2^e \\ \theta_3^e \end{bmatrix}}_{\boldsymbol{d}^e} = \underbrace{\begin{bmatrix} 1 & x_1^e & (x_1^e)^2 \\ 1 & x_2^e & (x_2^e)^2 \\ 1 & x_3^e & (x_3^e)^2 \end{bmatrix}}_{\boldsymbol{M}^e} \underbrace{\begin{bmatrix} \alpha_0^e \\ \alpha_1^e \\ \alpha_2^e \end{bmatrix}}_{\boldsymbol{a}^e}$$

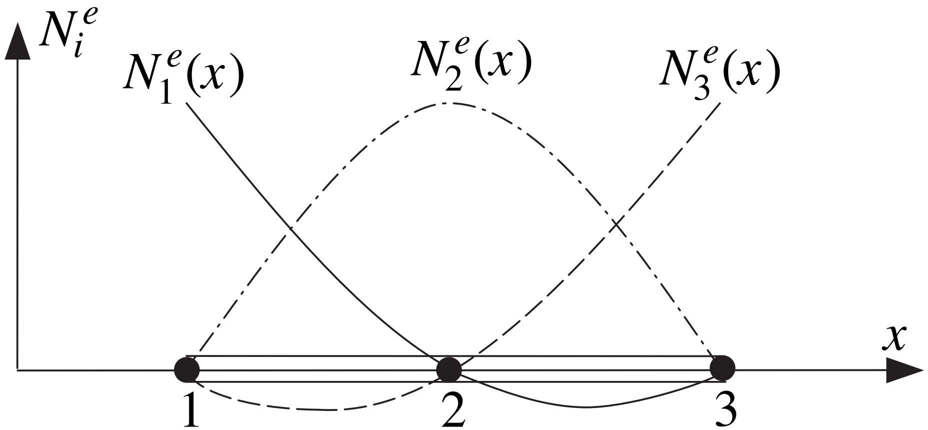
$$\boldsymbol{a}^e = \left(\boldsymbol{M}^e\right)^{-1} \boldsymbol{d}^e$$

$$\theta^e(x) = \boldsymbol{p}(x)\left(\boldsymbol{M}^e\right)^{-1} \boldsymbol{d}^e = \boldsymbol{N}^e(x)\boldsymbol{d}^e$$

$$\boldsymbol{N}^e(x) = \boldsymbol{p}(x)\left(\boldsymbol{M}^e\right)^{-1}$$

3.4 近似函数 Quadratic element

$$\boldsymbol{N}^e = \frac{2}{(l^e)^2}\left[\ (x-x_2^e)(x-x_3^e) \quad -2(x-x_1^e)(x-x_3^e) \quad (x-x_1^e)(x-x_2^e)\ \right]$$

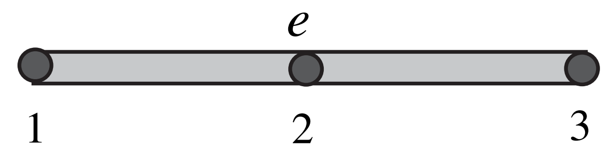


$$N_I^e(x_J^e) = \delta_{IJ}$$

$$\sum_{I=1}^{3} N_I^e(x) = 1$$

3.4 近似函数 Direct construction

Lagrange interpolants



$$N_I^e(x_J^e)=\delta_{IJ}$$

$$N_1^e(x)=\frac{(x-a)(x-b)}{c} \quad \Longleftarrow \quad N_1^e(x_2^e)=0,\quad N_1^e(x_3^e)=0$$

$$=\frac{(x-x_2^e)(x-x_3^e)}{c} \quad \Longleftarrow \quad N_1^e(x_1^e)=1$$

$$=\frac{(x-x_2^e)(x-x_3^e)}{(x_1^e-x_2^e)(x_1^e-x_3^e)} \quad \Longleftarrow \quad \text{If node 2 is at the center of the element}$$

$$=\frac{2}{(l^e)^2}(x-x_2^e)(x-x_3^e)$$

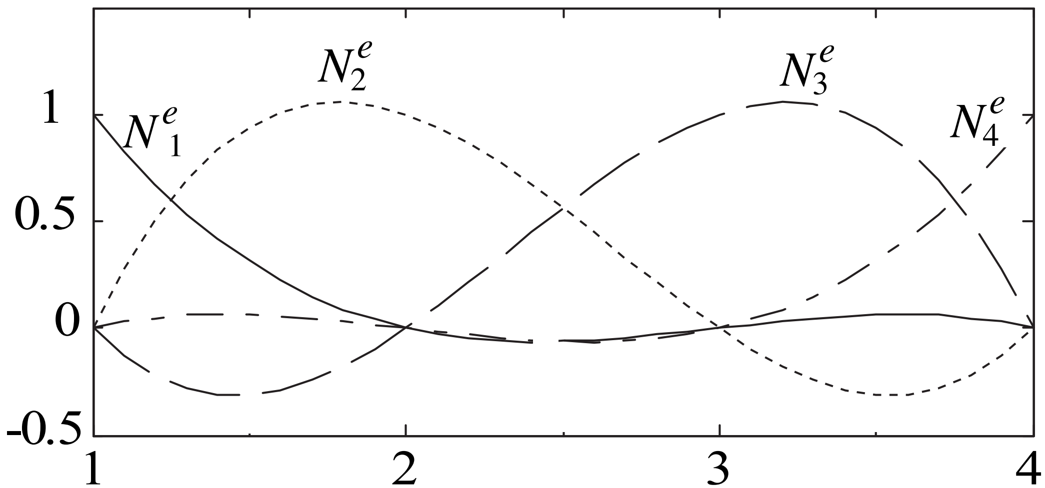
$$N_2^e=\frac{(x-x_1^e)(x-x_3^e)}{(x_2^e-x_1^e)(x_2^e-x_3^e)}=-\frac{4}{(l^e)^2}(x-x_1^e)(x-x_3^e)$$

$$N_3^e=\frac{(x-x_1^e)(x-x_2^e)}{(x_3^e-x_1^e)(x_3^e-x_2^e)}=\frac{2}{(l^e)^2}(x-x_1^e)(x-x_2^e)$$

Cubic shape functions

$$N_1^e = \frac{(x-x_2^e)(x-x_3^e)(x-x_4^e)}{(x_1^e-x_2^e)(x_1^e-x_3^e)(x_1^e-x_4^e)} \qquad N_3^e = \frac{(x-x_1^e)(x-x_2^e)(x-x_4^e)}{(x_3^e-x_1^e)(x_3^e-x_2^e)(x_3^e-x_4^e)}$$

$$N_2^e = \frac{(x-x_1^e)(x-x_3^e)(x-x_4^e)}{(x_2^e-x_1^e)(x_2^e-x_3^e)(x_2^e-x_4^e)} \qquad N_4^e = \frac{(x-x_1^e)(x-x_2^e)(x-x_3^e)}{(x_4^e-x_1^e)(x_4^e-x_2^e)(x_4^e-x_3^e)}$$



$$N_I^e = \prod_{J=1,\, J\neq I}^{n} \frac{x-x_J^e}{x_I-x_J^e} = \frac{(x-x_1^e)(x-x_2^e)\cdots(x-x_{I-1}^e)(x-x_{I+1}^e)\cdots(x-x_n^e)}{(x_I-x_1^e)(x_I-x_2^e)\cdots(x_I-x_{I-1}^e)(x_I-x_{I+1}^e)\cdots(x_I-x_n^e)}$$

3.4 近似函数

```
% shape functions computed in the physical coordinate - xt
function N = Nmatrix1D(xt,xe)
include_flags;

if nen == 2         % linear shape functions
   N(1) = (xt-xe(2))/(xe(1)-xe(2));
   N(2) = (xt-xe(1))/(xe(2)-xe(1));
elseif nen == 3    % quadratic shape functions
   N(1)=(xt-xe(2))*(xt-xe(3))/((xe(1)-xe(2))*(xe(1)-xe(3)));
   N(2)=(xt-xe(1))*(xt-xe(3))/((xe(2)-xe(1))*(xe(2)-xe(3)));
   N(3)=(xt-xe(1))*(xt-xe(2))/((xe(3)-xe(1))*(xe(3)-xe(2)));
end
```

Optimize the code ?

3.4 近似函数

```
% derivative of the shape functions computed in the
% physical coordinate - xt
function B = Bmatrix1D(xt,xe)
include_flags;

if nen == 2   % derivative of linear shape functions (constant)
   B = 1/(xe(2)-xe(1))*[-1 1];
elseif nen == 3   % derivative of quadratic shape functions
   B(1)=(2*xt-xe(2)-xe(3))/((xe(1)-xe(2))*(xe(1)-xe(3)));
   B(2)=(2*xt-xe(1)-xe(3))/((xe(2)-xe(1))*(xe(2)-xe(3)));
   B(3)=(2*xt-xe(1)-xe(2))/((xe(3)-xe(1))*(xe(3)-xe(2)));
end
```

Homework

Problem 4.3 Derive the shape functions for a two-node one-dimensional element which is C^1 continuous. Note that the shape functions derived in Chapter 4 are C^0 continuous. To enforce C^1 continuity, it is necessary to enforce continuity of displacements and their derivatives. Start by considering a complete cubic approximation $u^e = a_0 + a_1 x + a_2 x^2 + a_3 x^3$ and derive four shape functions corresponding to the displacements and their derivatives at each node. For clarity of notation, denote the derivatives at the nodes by ϕ_i , $i = 1, 2$.

Problem 4.4 Consider the displacement field $u(x) = x^3, 0 \le x \le 1$. Write a MATLAB program that performs the following tasks.

a. Subdivide the interval [0, 1] into two elements. Compute the displacement field in each element by letting the nodal displacements be given by $u_I = x_I^3$ and using a linear two-node element so that the displacement field in each element is given by $u^e(x) = \boldsymbol{N}^e(x)\boldsymbol{d}^e = \boldsymbol{N}^e(x)\boldsymbol{L}^e\boldsymbol{d}$, where $\boldsymbol{N}^e(x)$ are the linear shape functions given by (4.6). Plot $u(x)$ and the finite element field $u^e(x)$ on the same plot in the interval [0, 1].

b. Compute the strain in each element by $\varepsilon^e(x) = \boldsymbol{B}^e(x)\boldsymbol{d}^e = \boldsymbol{B}^e(x)\boldsymbol{L}^e\boldsymbol{d}$ and plot the finite element strain and the exact strain. How do these compare?

Homework

c. Repeat parts (a) and (b) for meshes of four and eight elements. Does the interpolation of the strain improve?

d. The error of an interpolation is generally measured by what is called a L^2 norm. The error in the L^2 norm, which we denote by e, is given by

$$\mathrm{e}^2 = \int_0^L (u^e - u)^2 \, \mathrm{d}x$$

where $u(x) = x^3$ in this case. Compute the error e for meshes of two, four and eight linear displacement elements. Use Gauss quadrature for integration. Then plot (this can be done manually) the error versus the element size on a log-log plot. This should almost be a straight line. What is its slope? This slope is indicative of the rate of convergence of the element.

e. Repeat part (d) using quadratic three-node quadratic elements.