Finite Element Method 1

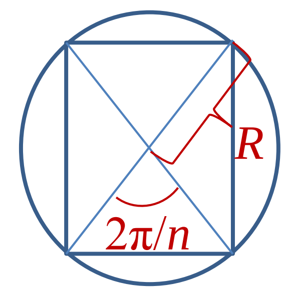
IPCST
Seoul National University

Area Estimation by Elements

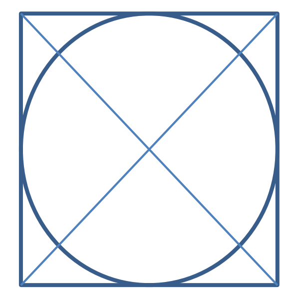
- Elements can be any shapes of any size.

1. Discretization
   - Elements: n triangles
   - Uniform mesh: same size
2. Element area
   - $a_A = ½·2R\sin(\pi/n)·R\cos(\pi/n)$
     $= ½·R^2\sin(2\pi/n)$
   - $a_B = ½·R·2R\tan(\pi/n) = R^2\tan(\pi/n)$

Mesh A



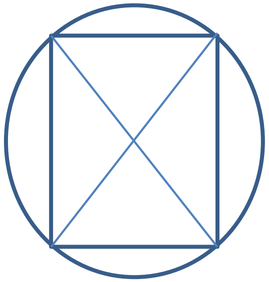
Mesh B



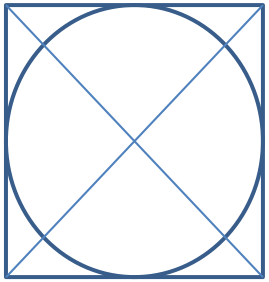
Area Estimation by Elements

3. Sum
   - $A_A = \sum a_A = \frac{1}{2} \cdot nR^2 \sin(2\pi/n)$
   - $A_B = \sum a_B = nR^2 \tan(\pi/n)$
4. Error
   - $E_A = \pi R^2 - A_A = R^2[\pi - \frac{1}{2} \cdot n \sin(2\pi/n)]$
   - $E_B = \pi R^2 - A_B = R^2[\pi - n \tan(\pi/n)]$
5. Convergence
   - $\lim_{n \to \infty} E_A = R^2[\pi - \pi \cdot 1] = 0$
   - $\lim_{n \to \infty} E_B = R^2[\pi - \pi \cdot 1/\cos(0)] = 0$

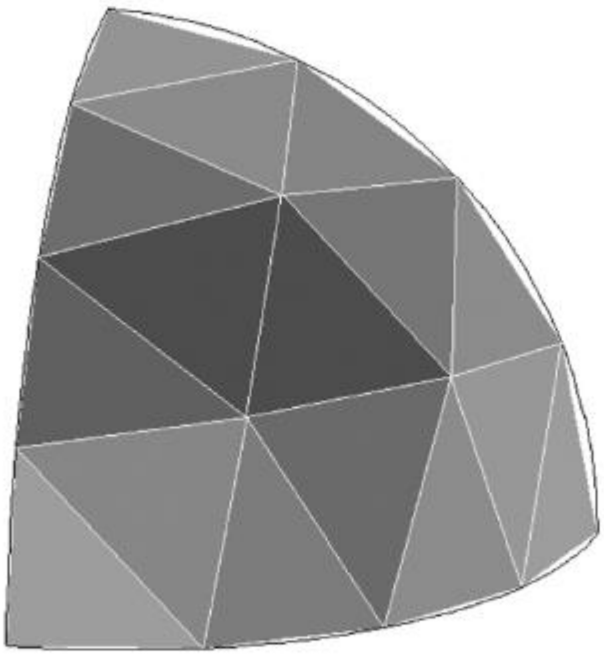
Mesh A

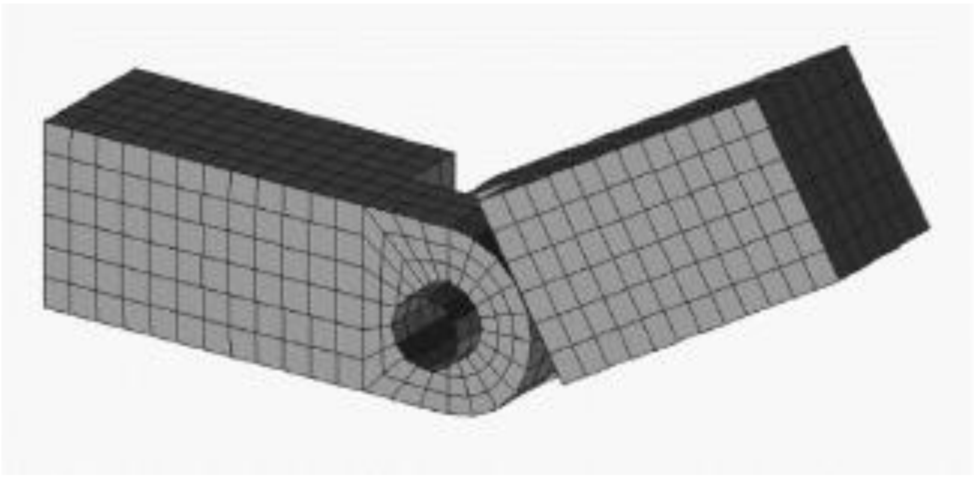


Mesh B



Mesh Examples for FEM





- Spherical surface replaced by triangles
- Hinge point: quadrilaterals

Figures from Liu & Quek

Spline Interpolation Revisited

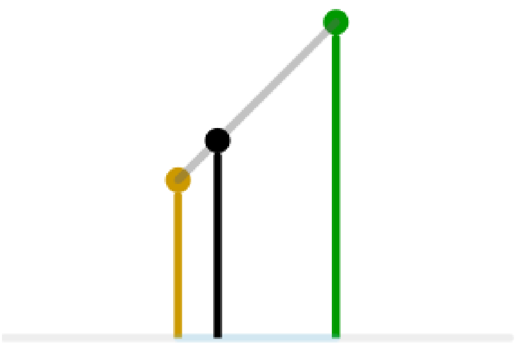
- Spline function
  - Concatenation of piecewise polynomials

$$S(x) \doteq \begin{cases} S_0(x), & x_0 \le x \le x_1 \\ S_1(x), & x_1 \le x \le x_2 \\ \vdots & \vdots \\ S_{n-1}(x), & x_{n-1} \le x \le x_n \end{cases}$$

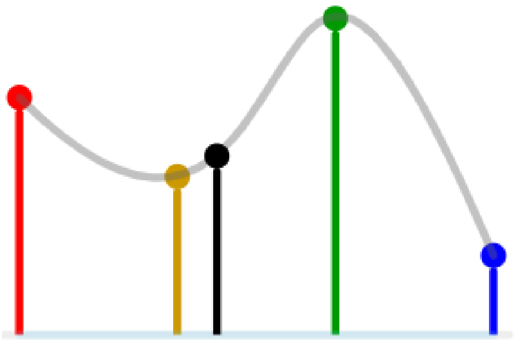
  - Continuous or smooth at $x = x_1, x_2, \ldots, x_{n-1}$

- A spline of degree k
  - $S_i(x)$ is a polynomial of degree $\le k$
  - $S(x)$ is $(k-1)$ times differentiable at $x = x_1, x_2, \ldots, x_{n-1}$ $(i = 1, \ldots, n-1)$

$$\begin{aligned} S_{i-1}(x_i) &= S_i(x_i), \\ S'_{i-1}(x_i) &= S'_i(x_i), \\ &\vdots \\ S_{i-1}^{(k-1)}(x_i) &= S_i^{(k-1)}(x_i). \end{aligned}$$



Linear



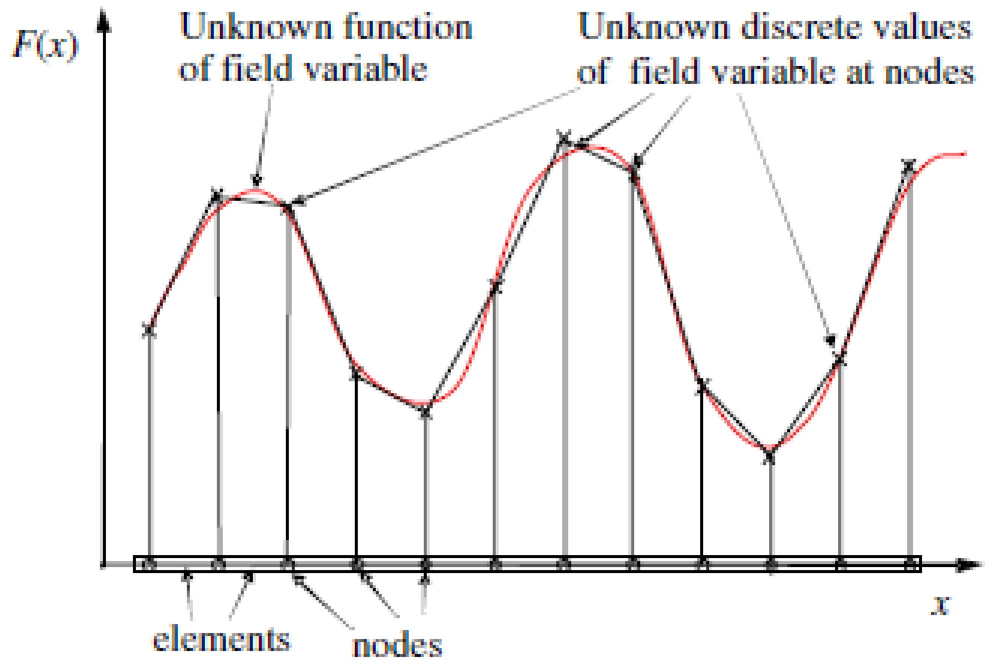
Cubic

Figures from Wikipedia

Piecewise Linear Function

- 1-D case
- $F(x)$
  - Field
  - Solution

  of the model
- Nodes
  - Element vertices



- The true solution is a continuous function which is often approximated as piecewise linear functions in 1-D FEM

Figure from Liu & Quek

What is FEM?

- Numerical method for field problems
  - Field values at nodes
- Discretization
  - Infinite degree of freedom (DOF) → finite DOF
  - A structure → several elements
    - Different types of elements are available.
  - Equation in each element
    - Solution: piecewise continuous function
  - Reconnecting solutions at nodes
    - Approximation: kind of interpolation

(cf.: spline interpolation)

Advantages of FEM

- Easy to handle complex shapes
- Easy to construct heterogeneous systems
- Easy to increase accuracy
  - By increasing the number of elements or changing piecewise functions
- Easy to include nonlinear effects
- Easy to apply boundary conditions
- Not difficult to solve dynamic problems

Disadvantages of FEM

- Large computational cost in comparison with other PDE methods
- Requirement of exercises to interpret results particularly in engineering
- Large amount of input data is often required.
- Result considerably depends on the input data and the modeling degree.

Stem of the FEM

1. An integral form (= A weak form)
2. Discretization: elements and nodes
3. Solution interpolated by a linear combination of basis functions
4. Linear algebraic equations

Weak Formulation

- An integral form (later approximated by discretization and simple functions – an analogy to adaptive quadrature)

- Types of weak forms
  1. Lagrangian functional (or the action functional from it)
     - Mechanics (See Liu & Quek)
  2. Weighted residual (or weighted integral)
     - All kinds of PDEs

Weight Integral Type Weak Form

- (Ideal) weak form
  - Completely equivalent to the strong form
  - For a PDE in the strong form of $f(u(\mathbf{x})) = 0$,

$$\int_\Omega W(\mathbf{x})\, f(u(\mathbf{x}))\; d\mathbf{x} = 0$$

    must satisfy for every smooth function W on infinite-dimensional function spaces with the conditions to make u keep the boundary conditions.

Weight Integral Type Weak Form

- (Ideal) weak form
  - Ex.)
    - A PDE $\sigma,_x + F = 0$ ($\sigma = Eu,_x$; F = force density) in the strong form can be transformed into

$$\int_0^L (W\sigma,_x + WF)Adx = 0$$

    - Integration by parts gives ($W \in V = \{W \,|\, W(0) = 0\}$)

$$\int_0^L W,_x\sigma \; Adx \; = \int_0^L WFAdx \; + \; W(L)P_0$$

Weight Integral Type Weak Form

- (Ideal) weak form
  - Ex.)

    - Proof of equivalence to the strong form

$$\int_0^L (W\sigma_{,x} + WF) A\,dx = 0$$

➢ If you input $W = g(x)(\sigma_{,x} + F)$ where

$g(x) > 0$ for $0 < x < L$ and $g(x) = 0$ at $x = 0$

then it must satisfy $g(x)(\sigma_{,x} + F)^2 = 0$ for $0 < x < L$

$\rightarrow \sigma_{,x} + F = 0$

Spaces of Integrable Functions

– Süli p. 8

- L_p space
  - Set of real-valued functions on Ω (an open subset of $\mathbb{R}^n$) with the condition,

$$\int_\Omega |u(x)|^p \,\mathrm{d}x < \infty.$$

  - p-norm

$$\|u\|_{L_p(\Omega)} := \left(\int_\Omega |u(x)|^p \,\mathrm{d}x \right)^{1/p}$$

Sobolev Spaces

– Süli pp. 10~11

- $W_p{}^k$
  - Function space including its derivatives: precisely, a Sobolev space of order *k* is
    $W_p{}^k(\Omega) = \{u \in L_p(\Omega) \mid D^\alpha u \in L_p(\Omega)\ ,\ |\alpha| \leq \mathrm{k}\}$
    - $D^\alpha = \dfrac{\partial^{|\alpha|}}{\partial x_1^{\alpha_1} \ldots \partial x_n^{\alpha_n}}$ $(|\alpha| = \alpha_1 + \alpha_2 + \ldots + \alpha_n)$
  - norm
    $$\|u\|_{W_p^k(\Omega)} := \left(\sum_{|\alpha| \leq k} \|D^\alpha u\|_{L_p(\Omega)}^p \right)^{1/p}$$

Hilbertian Sobolev Spaces

– Süli pp. 12~13

- $H^k = W_2{}^k$
  - Hilbert space with the inner product

$$(u, v)_{W_2^k(\Omega)} := \sum_{|\alpha| \le k} (D^\alpha u, D^\alpha v)$$

- H^1

$$H^1(\Omega) \;=\; \left\{ u \in L_2(\Omega) : \frac{\partial u}{\partial x_j} \in L_2(\Omega),\;\; j = 1, \ldots, n \right\},$$

$$\|u\|_{H^1(\Omega)} \;=\; \left\{ \|u\|^2_{L_2(\Omega)} + \sum_{j=1}^{n} \|\frac{\partial u}{\partial x_j}\|^2_{L_2(\Omega)} \right\}^{1/2}$$

$$|u|_{H^1(\Omega)} \;=\; \left\{ \sum_{j=1}^{n} \|\frac{\partial u}{\partial x_j}\|^2_{L_2(\Omega)} \right\}^{1/2}$$

Hilbertian Sobolev Spaces

– Süli pp. 12~13

- $H_0{}^1$

  $H_0{}^1(\Omega) = \{u \in H^1(\Omega) \mid u = 0 \text{ on } \partial\Omega\}$

- H^2

$$H^2(\Omega) = \left\{u \in L_2(\Omega) : \frac{\partial u}{\partial x_j} \in L_2(\Omega), \quad j = 1, \ldots, n, \right.$$
$$\left. \frac{\partial^2 u}{\partial x_i \partial x_j} \in L_2(\Omega), \quad i, j = 1, \ldots, n \right\},$$

$$\|u\|_{H^2(\Omega)} = \left\{ \|u\|^2_{L_2(\Omega)} + \sum_{j=1}^{n} \left\| \frac{\partial u}{\partial x_j} \right\|^2_{L_2(\Omega)} + \sum_{i,j=1}^{n} \left\| \frac{\partial^2 u}{\partial x_i \partial x_j} \right\|^2_{L_2(\Omega)} \right\}^{1/2}$$

$$|u|_{H^2(\Omega)} = \left\{ \sum_{i,j=1}^{n} \left\| \frac{\partial^2 u}{\partial x_i \partial x_j} \right\|^2_{L_2(\Omega)} \right\}^{1/2}$$

Weight Integral Type Weak Form

- (Galerkin) finite dimensional weak form
  - Let all functions in the integral equation be linear combinations of basis functions, then the integral equation becomes a system of linear algebraic equations.
  - Let $W = \sum c_i N_i$. Trying to make all c_i's, keeping the solution admissible, satisfy the integral equation gives the Galerkin equation $\mathbf{R}(u) = 0$ where $\mathbf{R}$ is the assembly of residuals for all elements.

Weight Integral Type Weak Form

- (Galerkin) finite dimensional weak form
  - Ex.)

    Solution $u_h \in \{u^1 \in H^1(0,L) \mid u^1(0) = u_0\}$

    and $W_h \in \{w^1 \in H^1(0,L) \mid w^1(0) = 0\}$ and

$$\int_0^L W_{h,x}\,\sigma_h A\,dx = \int_0^L W_h F A\,dx + W_h(L)P_0$$

where $\sigma_h = E\,u_{h,x}$

Weight Integral Type Weak Form

- (Galerkin) finite dimensional weak form
  - Ex.)

    Let $u_h = \sum c_i N_i$ and $W_h = \sum d_j N_j$ (N_i : basis function, usually simple polynomials. $i, j = 1, 2, \ldots\ldots, n$. n: number of elements), and then

$$\int_0^L W_{h,x}\, \sigma_h A dx = \sum_{ij} c_i d_j \int_0^L N_{j,x} N_{i,x} E A dx$$

$$\int_0^L W_h F A dx = \sum_j d_j \int_0^L N_j F A dx$$

$$W_h(L) P_0 \rightarrow d_n P_0 N_n(L)$$

Weight Integral Type Weak Form

- (Galerkin) finite dimensional weak form
  - Ex.) $$\int_0^L W_{h,x}\,\sigma_h A dx = \int_0^L W_h F A dx + W_h(L) P_0$$

    will give a matrix equation of the form $\mathbf{w}^T\mathbf{A}\mathbf{u} = \mathbf{w}^T\mathbf{f}$ ($\mathbf{u}$: vector of c_i's & $\mathbf{w}$: vector of d_j's). You can erase $\mathbf{w}^T$ since W_h can be arbitrary. Therefore, $\mathbf{A}\mathbf{u} = \mathbf{f}$.

$$\mathrm{A}_{ij} = \int_0^L N_{j,x}\, N_{i,x}\, E A dx \qquad f_j = \int_0^L N_j F A dx$$

for $j \neq n$

Properties of Shape Functions

- Shape functions (= basis functions) have

1. Linear independence
2. Delta function property

   $N_i(x_j) = \delta_{ij}$

   – x_j : coordinate of *j*th node

   – 1 for the designated node; 0 for the others

3. Partition of unity

   $\sum N_i(x) = 1$

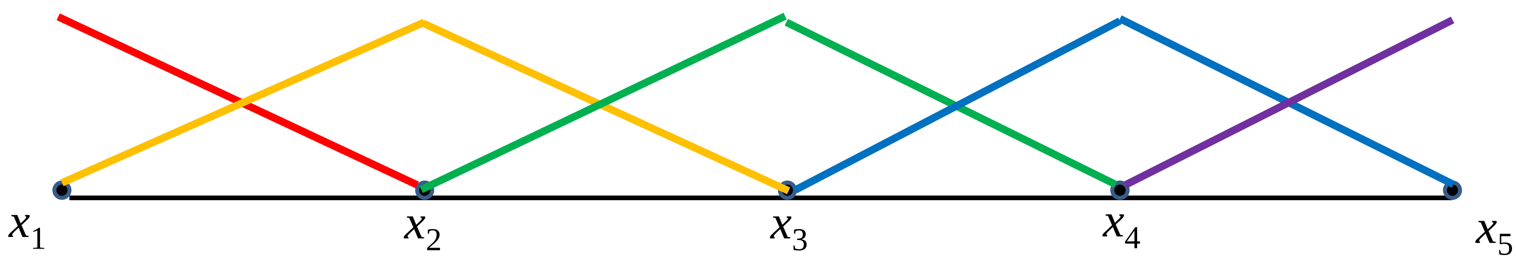
   – Sum of all shape functions is always 1.

1-D Shape Functions

- Linear shape functions (2 functions in 1 element)

$$N_i(x) = (x - x_{i-1})/l_{i-1} \quad , x_{i-1} \leq x \leq x_i$$
$$= 1 - (x - x_i)/l_i \quad , x_i \leq x \leq x_{i+1}$$
$$= 0 \,, \qquad \text{otherwise}$$

l_i : length of ith element
(= x_{i+1} - x_i)



– In the 1-D mechanical problem, $u_h = \sum c_i N_i$
  - Ex.) For the element 1 ($0 \leq x \leq l_1$),

$$u_{h,1} = c_1 N_1 + c_2 N_2 = c_1 + \frac{c_2 - c_1}{l_1} x$$

✓ $u_h(x_i) = c_i \equiv U_i$

1-D Shape Functions

- Natural coordinate $\xi = \frac{2(x - x_c)}{l_e}$
  - x_c: center of element, l_e: element length

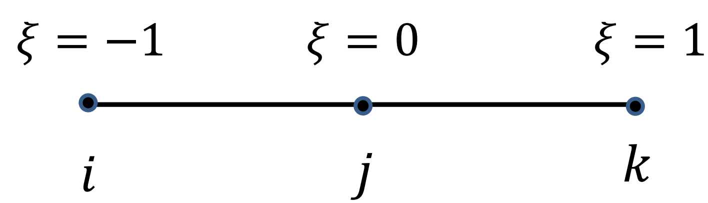
❖ Higher order shape functions can be derived from Lagrange interpolation.

- Quadratic shape functions (3 functions in 1 element)

$$N_i(\xi) = -\tfrac{1}{2}\xi(1-\xi)$$

$$N_j(\xi) = (1+\xi)(1-\xi)$$

$$N_k(\xi) = \tfrac{1}{2}\xi(1+\xi)$$



1-D Shape Functions

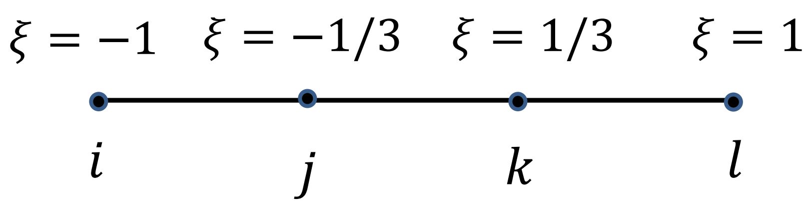
- Cubic shape functions (4 functions in 1 element)

$$N_i(\xi) = -\frac{1}{16}(1-\xi)(1-9\xi^2)$$

$$N_j(\xi) = \frac{9}{16}(1-3\xi)(1-\xi^2)$$

$$N_k(\xi) = \frac{9}{16}(1+3\xi)(1-\xi^2)$$

$$N_l(\xi) = -\frac{1}{16}(1+\xi)(1-9\xi^2)$$

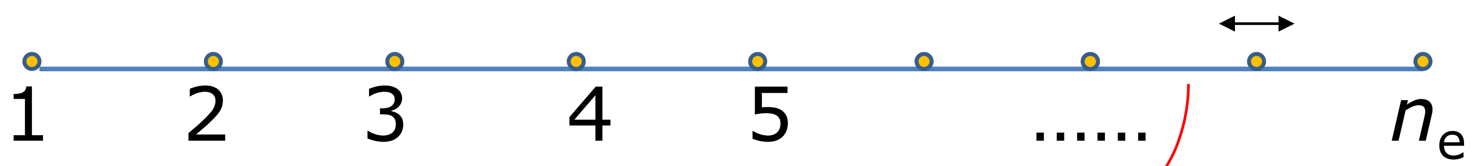


FEM Procedure

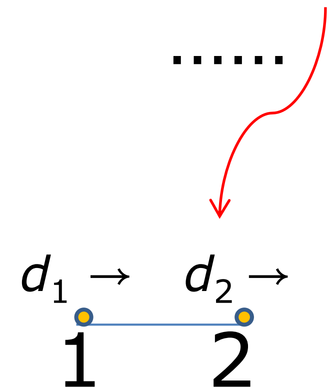
1. Mesh generation (domain discretization)
2. Interpolation of state variables (ex.: displacement)
   - Setting degrees of freedom
   - Selecting or constructing shape functions
3. Constructing element equations in local coordinate system
4. Coordinate transformation
5. Assembly in the global equation
6. Imposing boundary conditions
7. Solving the global equation

Truss Model 1

- Global coordinates



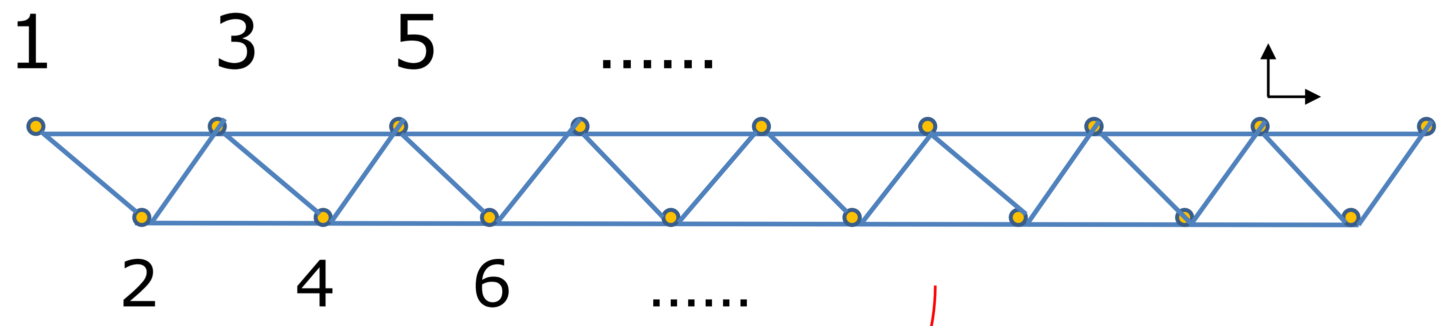
- Local coordinates
  - d_i : displacement



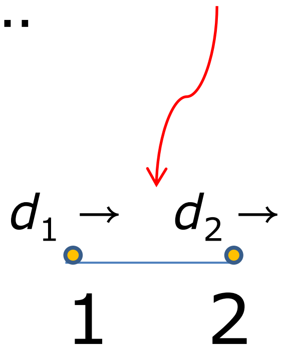
❖ (degree of freedom per node) = 1

Truss Model 2

- Global coordinates



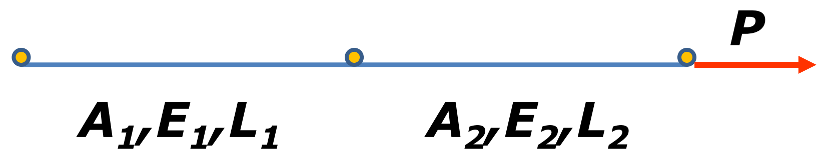
- Local coordinates
  - d_i : displacement



❖ (degree of freedom per node) = 2

Simple Example

- Two elements



$$u_h = \sum U_i N_i$$

- For element 1

$$\frac{A_1E_1}{L_1}\begin{bmatrix} 1 & -1 & 0 \\ -1 & 1 & 0 \\ 0 & 0 & 0 \end{bmatrix}\begin{Bmatrix} U_1 \\ U_2 \\ U_3 \end{Bmatrix} = \begin{Bmatrix} P_1^{(1)} \\ P_2^{(1)} \\ 0 \end{Bmatrix}$$

- For element 2

$$\frac{A_2E_2}{L_2}\begin{bmatrix} 0 & 0 & 0 \\ 0 & 1 & -1 \\ 0 & -1 & 1 \end{bmatrix}\begin{Bmatrix} U_1 \\ U_2 \\ U_3 \end{Bmatrix} = \begin{Bmatrix} 0 \\ P_2^{(2)} \\ P_3^{(2)} \end{Bmatrix}$$

- Assembled global equation

$$\begin{bmatrix} \frac{A_1E_1}{L_1} & -\frac{A_1E_1}{L_1} & 0 \\ -\frac{A_1E_1}{L_1} & \frac{A_1E_1}{L_1}+\frac{A_2E_2}{L_2} & -\frac{A_2E_2}{L_2} \\ 0 & -\frac{A_2E_2}{L_2} & \frac{A_2E_2}{L_2} \end{bmatrix}\begin{Bmatrix} U_1 \\ U_2 \\ U_3 \end{Bmatrix} = \begin{Bmatrix} P_1^{(1)} \\ P_2^{(1)}+P_2^{(2)} \\ P_3^{(2)} \end{Bmatrix} = \begin{Bmatrix} P_1 \\ P_2 \\ P_3 \end{Bmatrix}$$

Simple Example

- Boundary conditions

$$U_1 = 0$$
$$P_2 = 0$$
$$P_3 = P$$

- Assembled global equation

$$\begin{bmatrix} \frac{A_1E_1}{L_1} & -\frac{A_1E_1}{L_1} & 0 \\ -\frac{A_1E_1}{L_1} & \frac{A_1E_1}{L_1}+\frac{A_2E_2}{L_2} & -\frac{A_2E_2}{L_2} \\ 0 & -\frac{A_2E_2}{L_2} & \frac{A_2E_2}{L_2} \end{bmatrix} \begin{Bmatrix} 0 \\ U_2 \\ U_3 \end{Bmatrix} = \begin{Bmatrix} P_1 \\ 0 \\ P \end{Bmatrix}$$

$$\begin{bmatrix} \frac{A_1E_1}{L_1}+\frac{A_2E_2}{L_2} & -\frac{A_2E_2}{L_2} \\ -\frac{A_2E_2}{L_2} & \frac{A_2E_2}{L_2} \end{bmatrix} \begin{Bmatrix} U_2 \\ U_3 \end{Bmatrix} = \begin{Bmatrix} 0 \\ P \end{Bmatrix}$$

What if $U_1 \neq 0$?

Do It Yourself

- Develop the finite element equation for this model.

$\boldsymbol{E_1}=\boldsymbol{E_3}=100$, $\boldsymbol{E_2}=50$, $\boldsymbol{A}=1$, $\boldsymbol{L}=1$, $\boldsymbol{P}=10$

– Units are omitted.

- Compute nodal displacements and forces.

Key Properties of the FEM

- Best approximation property
    - a. k. a. Reproduction property
  - **Guarantee** to choose the best possible solution that can be produced *by the shape functions*.

- Convergence property
  - Rate of convergence: $O(h^{p+1})$
    - *p* : polynomial order, *h* : element size
  - The error of the FEM decreases for finer elements (*h*-convergence) or higher order of shape functions (*p*-convergence)

Code Structure

1. Setting elements and nodes
    1. Mesh generation
    2. Setting degrees of freedom
    3. Global numbering
2. Calculation in local coordinates
    1. Estimation of shape functions and their gradients
    2. Integration (quadrature)
    3. Mapping
3. Assembly in global coordinates
    1. Composing matrices and vectors
    2. Imposing boundary conditions (or initial conditions)
4. Solving the linear algebra equation
5. Output

❖ You can exclude some steps from your code and do them by hand.

Error Sources

- Approximation of domain
  - Discretization error
- Approximation of solution
  - Interpolation error
- Numerical error

參考: Error Estimation

- A priori error estimates
  - Related to the best approximation property
  - Typical one is

    $$\| e \|_{H^1} \leq C\, h^p \,| u |_{H^{p+1}} \qquad (C\text{: const.})$$

    - A value of the constant C can be found in the chapter 3 of the E. Süli's Lecture note.
  - Another one is

    $$\| e' \|_{L^2} \leq C\, h^p \,\| u'' \|_{L^{p+1}} \qquad (C\text{: const.})$$

  - Some use energy norms instead of H-norms or L-norms. See Oden & Reddy or other references.

參考: Error Estimation

- A posteriori error estimate
  - Can be used to refine adaptive mesh
  - There are various kinds. One for a 1-D case (the chapter 4 of the E. Süli's Lecture note) is

$$\| \ e \ \|_{L_2} \leq K_0 \left(\sum_{i=1}^{N} h_i^4 \| R(u_h) \|_{L_2(x_{i-1},x_i)}^2 \right)^{1/2}$$

$R(u_h) = f - F(u_h)$ from a PDE $F(u) = -u'' + bu' + cu = f$

$K = 1 + \frac{1}{\sqrt{2}} \|b\|_{L_\infty(0,1)} + \frac{1}{2} \|c - b'\|_{L_\infty(0,1)}$ $K_0 = K/\pi^2$ ($\|\ \|_{L2}$: L_2 norm)

- See also T. Grätsch & K.-J. Bathe, *Comput. Struct.* **83**, 235 (2005).

參考: Error Estimation

- A posteriori error estimate
  - Other references
    - M. Ainsworth & J. T. Oden, “A Posteriori Error Estimation in Finite Element Analysis”, (2000).
    - I. Babuska & W. C. Rheinboldt, *SIAM J. Numer. Anal.* **15**, 736 (1978).
    - I. Babuska & W. C. Rheinboldt, *Internat. J. Numer. Methods Engng.* **12**, 1597 (1978).
    - R. E. Bank & A. Weiser, *Math. Comp.* **44**, 283 (1985).
    - O. C. Zienkiewicz and J. Z. Zhu, *Internat. J. Numer. Methods Engrg.* 24, 337 (1987).

參考: Adaptive Mesh

- h-adaptivity
  – Halving h (in 2D, 1 element → 4 elements)

- hp-adaptivity (hp-FEM)
  – Adding p-refinement
    - Higher polynomial degree
    - Increasing 1 or 2 order
  – Various algorithms
    - Various combination of h- & p- refinements
    - 1 error estimate per element isn't enough.

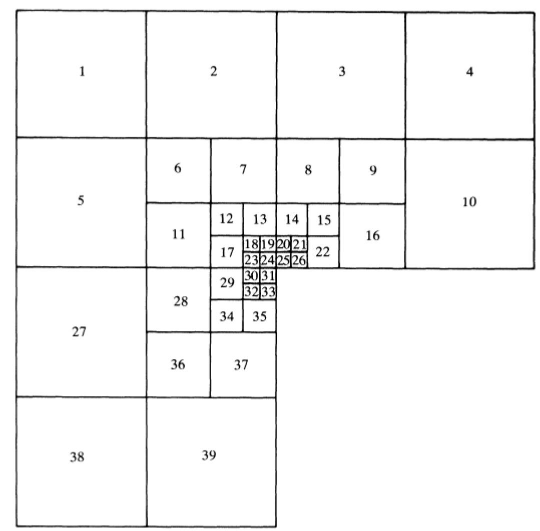


Figure from Babuska & Rheinboldt

參考: FEM Packages

- Abaqus
- ANSA Pre-processor
- COMSOL Multiphysics
- FEBio
- Z88 FEM software
- Elmer
- ……
- MATLAB (FEM toolbox), Mathematica

參考: FEM Libraries

- FEniCS (Python or C++)
- SfePy (Python)
- deal.II (C++)
- MFEM (C++)
- DUNE (C++)
  - http://dune-project.org/

- There are many others.

參考: Visualization

- Standard formats for computational data exchange and processing
  - VTK, HDF5, XDMF
- FEM visualization packages
  - ParaView
  - MayaVi

Investigation

- About the **Lagrangian functional** weak form

References

- Süli,
  "Lecture Notes on Finite Element Methods for Partial Differential Equations"

- Liu & Quek,
  The Finite Element Method (2nd Edition)

- Zienkiewicz & Taylor,
  The Finite Element Method (5th Ed.),
  Vol. 1: The Basis.

References

- Oden & Reddy,
  An Introduction to the Mathematical Theory of Finite Elements.

- Reddy,
  An Introduction to the Finite Element Method

- Strang & Fix,
  An Analysis of the Finite Element Method (2nd Ed., Wellesley-Cambridge Press)

References

- Larson & Bengzon,
  The Finite Element Method
  - Theory, Implementation and Applications
- Šlesinger, “Tutorial: The Finite Element Method”
- Sadd, “Introduction to Finite Element Methods”
- I. Babuska & W. C. Rheinboldt, *SIAM J. Numer. Anal.* **15**, 736 (1978).
- Wikipedia