The Finite Element Method for the Analysis of Non-Linear and Dynamic Systems

Prof. Dr. Eleni Chatzi
Numerical Recipes
Basic Formulas

**Galerkin’s** method assumes that the approximate (or *trial*) solution, \( u \), can be expressed as a linear combination of the nodal point displacements \( u_i \), where \( i \) refers to the corresponding node number.

\[
u(x) \approx u^h(x) = \sum_i N_i(x)u_i = N(x)u
\]

The same applies for the weighting function \( w \):

\[
w(x) \approx w^h(x) = \sum_i N_i(x)w_i = N(x)w
\]

The shape functions correspond to each degree of freedom (dof), and they have to satisfy the condition of being equal to 1 at the node of the corresponding dof and 0 everywhere else.
Basic Formulas-Shape Functions

The order of the shape function is chosen based on the problem’s strong form. Moreover, the shape functions are usually expressed in the iso-parametric coordinate system.

- Bar Element

For the standard bar element, the strong form is $AE \frac{d^2u}{dx^2} + f(x) = 0$. The homogeneous form is $AE \frac{d^2u}{dx^2} = 0$ which is a 2nd order ODE with known solution: $u(x) = C_1x + C_2$, i.e., a 1st degree polynomial.

The shape functions $N(\xi)$ are therefore selected as 1st degree polynomials:

$$N(\xi) = \begin{bmatrix} \frac{1}{2}(1 - \xi) \\ \frac{1}{2}(1 + \xi) \end{bmatrix} \rightarrow u_1 \quad \rightarrow u_2$$
Basic Formulas-Shape Functions

**Beam Element**

For the standard beam element, the strong form is

\[ \frac{d^2}{dx^2} \left( EI \frac{d^2 \nu}{dx^2} \right) + f(x) = 0. \]

The homogeneous form is

\[ \frac{d^2}{dx^2} \left( EI \frac{d^2 \nu}{dx^2} \right) = 0 \]

which is a 4th order ODE with known solution:

\[ u(x) = C_1x^3 + C_2x^2 + C_3x + C_4, \]

i.e., a 3rd degree polynomial.

The shape functions \( H(\xi) \) are therefore selected as 3rd degree polynomials:

\[
H(\xi) = \begin{bmatrix}
\frac{1}{4} (1 - \xi)^2 (2 + \xi) \\
\frac{1}{4} (1 - \xi)^2 (1 + \xi) \\
\frac{1}{4} (1 + \xi)^2 (2 + \xi) \\
\frac{1}{4} (1 + \xi)^2 (\xi - 1)
\end{bmatrix}
\]

\[ \rightarrow u_1 \quad \rightarrow \theta_1 \quad \rightarrow u_2 \quad \rightarrow \theta_2 \]
Basic Formulas-Stiffness Matrices

Going from Strong to Weak form yields the expressions of the equivalent stiffness matrix for each element:

- **Bar Element**

\[
K_{ij} = \int_0^L \frac{dN_j(x)}{dx} AE \frac{dN_i(x)}{dx} \, dx
\]

Using matrix notation, the definition of the strain-displacement matrix \( B = \frac{dN(x)}{dx} \), and the definition of the Jacobian (since we are looking for derivatives in term of \( x \) and not \( \xi \) ultimately):

\[
J = \frac{dx}{d\xi} = \frac{x^e_2 - x^e_1}{2} = \frac{h}{2}
\]

we obtain:

\[
K^e = \int_0^L B^T E A B \, dx = \frac{EA}{h} \begin{bmatrix}
1 & -1 \\
-1 & 1
\end{bmatrix}
\]

where \( h \) is the length of the element.
Basic Formulas-Stiffness Matrices

Going from Strong to Weak form yields the expressions of the equivalent stiffness matrix for each element:

- **Beam Element**

\[ K_{ij} = \int_0^L d^2H_j(x) \frac{EI}{dx^2} d^2H_i(x) \frac{dx}{dx^2} dx \]

Using matrix notation, the definition of the strain-displacement matrix \( B = \frac{d^2H(x)}{dx^2} \), and the definition of the Jacobian (since we are looking for derivatives in terms of \( x \) and not \( \xi \) ultimately): \( J = \frac{dx}{d\xi} = \frac{l}{2} \) we obtain:

\[ K^e = \int_0^L B^T EI B dx = \frac{EI}{l^3} \begin{bmatrix} 12 & 6l & -12 & 6l \\ 6l & 4l^2 & -6l & 2l^2 \\ -12 & -6l & 12 & -6l \\ 6l & 2l^2 & -6l & 4l^2 \end{bmatrix} \]

where \( l \) is the length of the element.
Note: Gaussian Quadrature

A number of integrals need be evaluated for the formulation of both the stiffness matrices as well as the load vectors. How may we numerically calculate integrals of general functions using an automated way?

First note that when using an iso-parametric formulation these integrals are typically expressed within the range of $[-1, 1]$. Gaussian Quadrature enables calculation of such integrals using the following approximation:

$$
\int_{-1}^{1} f(x) dx \approx \sum_{i=1}^{n} w_i f(x_i)
$$

This rule succeeds in delivering an exact result for polynomials of degree $2n - 1$ or less by a suitable choice of the points $x_i$ and weights $w_i$ for $i = 1, ..., n$. 
Note: Gaussian Quadrature

Depending on the order of the polynomial to be approximated, the corresponding order (number of point) for the quadrature are selected along with the corresponding weights from the following table:

<table>
<thead>
<tr>
<th># Points</th>
<th>$x_i$</th>
<th>$w_i$</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>0</td>
<td>2</td>
</tr>
<tr>
<td>2</td>
<td>${\pm \sqrt{1/3}}$</td>
<td>1</td>
</tr>
<tr>
<td>3</td>
<td>${0, \pm \sqrt{3/5}}$</td>
<td>${8/9, 5/9}$</td>
</tr>
<tr>
<td>4</td>
<td>${\pm \sqrt{3/7 - 2/7 \sqrt{6/5}},$</td>
<td>${(18 + \sqrt{30})/36,$</td>
</tr>
<tr>
<td></td>
<td>$\pm \sqrt{3/7 + 2/7 \sqrt{6/5}}}$</td>
<td>$18 - \sqrt{30}/36}$</td>
</tr>
</tbody>
</table>

The scheme can be extended to 2 or 3 dimensions in a similar manner.