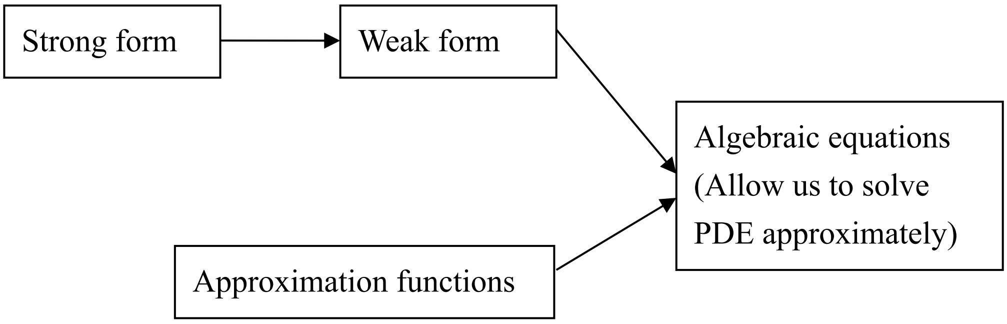
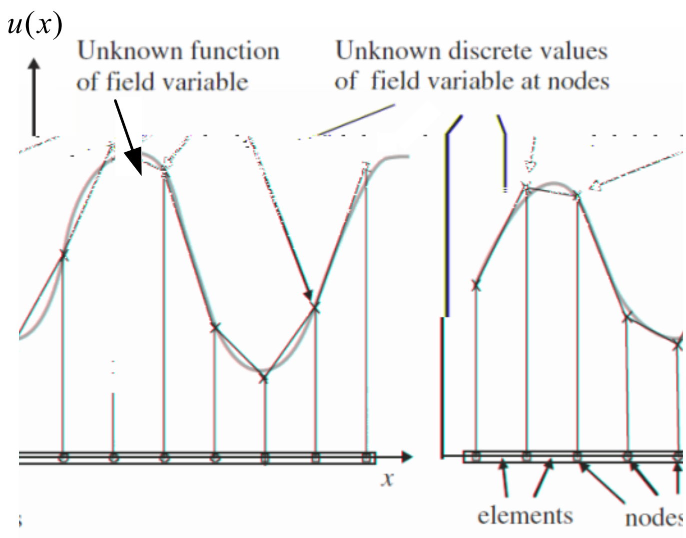
03/06/2019

Chapter 2: 1D Finite Element Formulation



2.1 Introduction



The finite element method treats a given domain as a collection of simple subdomains, called **finite elements**. Since the subdomain is relatively simple, it is possible to systematically construct the approximation functions needed in the weighted-residual method over **each element**.

The finite element differs from the classical Galerkin method in the manner in which the approximation solutions are constructed. This difference leads to three basic features of the finite element method:

1. **Division of whole domain into subdomains** that enables a systematic derivation of the approximation functions over a complex domain.
2. **Derivation of approximation functions over each element**. The approximation functions are often algebraic polynomials that are derived using interpolation theory.
3. **Assembly of elements** is based on enforcing **compatibility** to ensure the solution $u(x)$ at

common nodes are identical.

We will start introducing fundamental ideas that form the finite element method.

2.2 Basic Steps of Finite Element Analysis

The basic steps involved in the finite element analysis of a typical problem are:

Step1: Discretization of the given domain into a collection of preselected finite elements. We thus construct the **finite element** **mesh** of preselected elements (see below).

Step 2: Derivation of element equations for all typical elements in the mesh.

Step 3: Assembly of element equations to obtain equations of the whole problem.

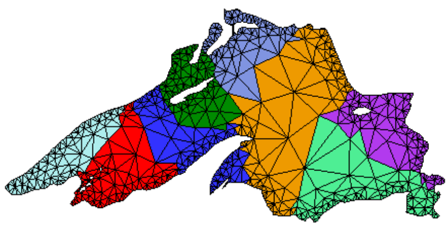
Step 4: Imposition of boundary conditions of the problem.

Step 5: Solution of the assembled equations.

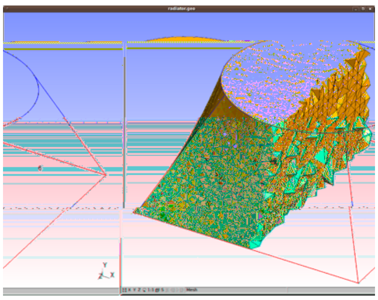
Step 6: Postprocessing of the results.

Remarks:

1. Discretization of a given domain into a collection of preselected finite elements is called **meshing**, triangulation or tessellation. It is an active research topic in the field of Computational Geometry and FEM has been served as the most important application.



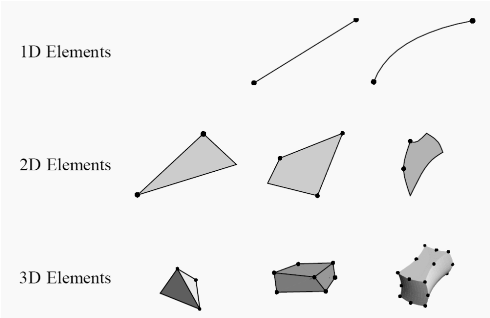
Triangle: http://www.cs.cmu.edu/~quake/triangle.html



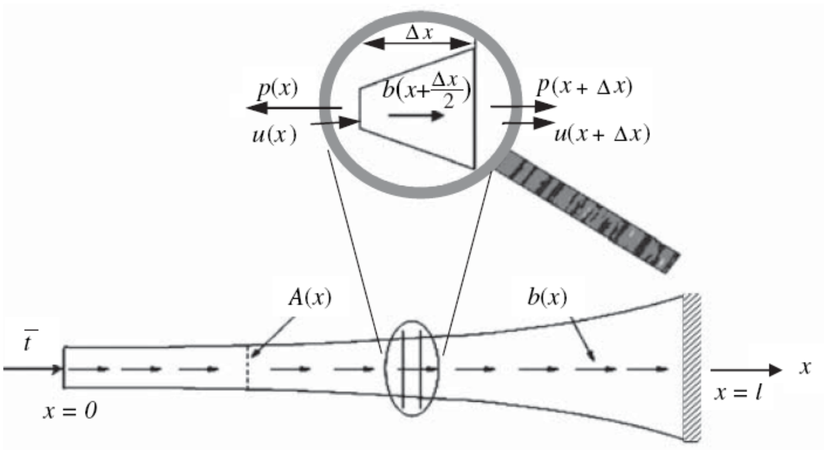
Clipped view of Tetrahedral mesh; generated and viewed in Gmsh: http://geuz.org/gmsh/

2. FEM meshing decomposes the domain into nodes and elements. Typical elements in 1D,

2D and 3D are listed below:



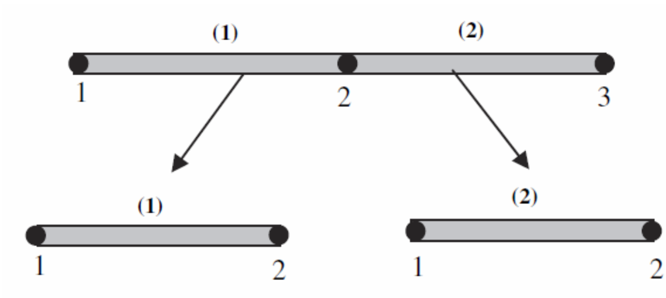
We will go through these **basic 6 steps** to **highlight fundamental ideas** behind the finite element method. A simple 1D elastic bar example will be used to demonstrate these ideas.



Notations: $u(x)$: displacement, $b(x)$: body force or distributed loading (force/length), $p(x)$: internal force, $A(x)$: the cross-sectional area, $\bar{t}$: traction (load prescribed at the end node)
Goal: Solve $u(x)$. From $u(x)$, we find strain $\varepsilon(x)$ and stress $\sigma(x)$.

Step 1: Discretization of the given domain into a collection of preselected finite elements.

We simply consider a mesh with 2 elements illustrated below. Notice the global and local node numbers for the mesh. The local node number of a bar element are always numbered 1, 2 in the positive x-direction. The global node numbers are arbitrary.



Step 2: Derivation of element equations for all typical elements in the mesh.

This step is in heart of finite element formulation and it involves three sub-steps: (a) construct the weak form of the given differential equation over an element, (b) express the approximation solution $u(x)$ for a particular element e (will be denoted by $u^e(x)$) in the form of **shape functions**, and (c) substitute ( ) into the weak form to obtain element equations.

<u>**Substep 2.a** Construct the weak form of the given differential equation over an element</u>

The weak form for the example has been developed previously (Eq. (1.9)) and is rewritten below:

Find among the smooth functions that satisfy such that

(2.1)

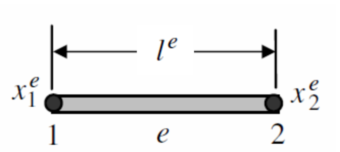
Notice that we have taken the **transpose** of the weight functions; as is a scalar, this does not matter, but it is necessary for consistency when we later substitute matrix expressions for and its derivative.

Without loss of generality, we can replace the weak form of integral over the entire domain with a sum of integrals over an individual element domain :

(2.2)

in which n_{el} is number of elements (in our simple case, $n_{el} =$).

Substep 2.b Express the approximation solution $u^e(x)$ for an element *e* in the form of **shape functions**.



For the moment, let us consider a very simple two-node linear element. As in the classical Galerkin method, the approximation solution $u^e(x)$ must fulfill certain conditions to ensure convergence to the actual solution as the number of elements is increased. They are:

1. It should be continuous over the element and differentiable, **as required by the weak form**.
2. It should be **complete** polynomial, i.e., including all lower-order terms up to the highest order used.

We thus can express $u^e(x)$ in terms of linear polynomial as we have done in the classical methods:

$$u^e(x) = \alpha_0^e + \alpha_1^e x = [1 \quad x]\begin{bmatrix}\alpha_0^e \\ \alpha_1^e\end{bmatrix} = \mathbf{p}(x)\boldsymbol{\alpha}^e \tag{2.3}$$

To ease the compatibility requirement to ensure the solution $u(x)$ at common nodes are identical, we express the *unknown* coefficients α_0^e and α_1^e in terms of the *unknown* **nodal values** of the approximation. That is, nodal displacements at nodes 1 and 2 in our case:

$$u^e(x_1^e) \equiv u_1^e = \alpha_0^e + \alpha_1^e x_1^e \qquad u^e(x_2^e) \equiv u_2^e = \alpha_0^e + \alpha_1^e x_2^e \qquad \Rightarrow \tag{2.4}$$

$$\begin{bmatrix} u_1^e \\ u_2^e \end{bmatrix} = \begin{bmatrix} 1 & x_1^e \\ 1 & x_2^e \end{bmatrix}\begin{bmatrix} \alpha_0^e \\ \alpha_1^e \end{bmatrix} \qquad \Rightarrow \qquad \mathbf{d}^e = \mathbf{M}^e \boldsymbol{\alpha}^e$$

$$\mathbf{d}^e \equiv \begin{bmatrix} u_1^e \\ u_2^e \end{bmatrix}$$

Q: How to resolve the unknown coefficients $\boldsymbol{\alpha}^e$ if we have solved the nodal displacements $\mathbf{d}^e$?

A:

(2.5)

Substituting (2.5) into (2.3), we then have:

(2.6) $\quad (\ ) = \mathbf{N}\ (\ )\mathbf{d} \qquad \text{where} \quad \mathbf{N}\ (\ ) = \mathbf{p}(\ )\left(\mathbf{M}\ \right)^{-1}$

(Derivation Exercise): find the element shape function of Eq. (2.6).

$\mathbf{p}(x) = ?$

$\left(\mathbf{M}^e\right)^{-1} = ?$

$(\ ) = \mathbf{N}\ (\ )\mathbf{d} \qquad \text{where} \quad \mathbf{N}\ (\ ) = \mathbf{p}(\ )\left(\mathbf{M}\ \right)^{-1}$

(Answer)

Remarks:

1. The row matrix $\mathbf{N}^e(x) = \begin{bmatrix} N_1^e(x) & N_2^e(x) \end{bmatrix}$ is called the **element shape function** matrix.

   We will see that **shape functions** play a central role in the finite element method.
2. Shape functions of various polynomial orders enable the finite element method to solve problems of many types with varying degrees of accuracy. We will show a more systematic way through Lagrange interpolants to construct shape functions for any polynomial order.
3. $N_1^e(x)$ and $N_2^e(x)$ are the element shape functions for nodes 1 and 2.

Q: What will $N_1^e(x)$ and $N_2^e(x)$ look like if we plot their values vs. x?

A:

Note that they are nonzero only **in** the element e. **This is a very important feature in FEM.**

<u>The Interpolation Property of Shape Functions</u>

The shape functions have the following properties:

$N_1^e(x_1^e) = \_\_\_$ and $N_1^e(x_2^e) = \_\_\_$

$N_2^e(x_1^e) = \_\_\_$ and $N_2^e(x_2^e) = \_\_\_$

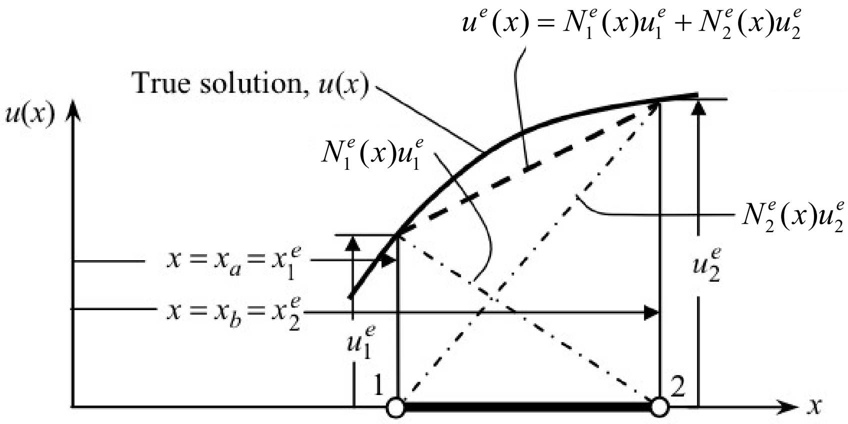
Or in a concise notation:

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

where δ_{IJ} **the Kronecker delta** and is given by

$$\delta_{IJ} = \begin{cases} 1 & \text{if } I = J \\ 0 & \text{if } I \neq J \end{cases}$$

Eq. (2.7) is known as the **Kronecker delta property** and is related to a fundamental property of shape functions called the interpolation property. **Interpolants** are functions that pass exactly through the data. If you think of nodal values as data, then shape functions are **interpolants of the nodal data**.



In the weak form (2.2), we need to evaluate the derivatives of the trial functions and weight functions.

Q: What is the derivative $\dfrac{du^e(x)}{dx}$ for the two-node linear element?

A:

Remark: How about weight function $w^e(x)$?

It is not required that the weight functions be approximated by the same interpolants (i.e., the same shape functions) that are used for the trial solutions approximation. However, for most problems it is advantageous to use the same approximation for the trial solutions and the weight functions. The resulting method is called the **Galerkin FEM**.

In summary, the trial solution and weight function and their derivatives in each linear element can be written:

(2.8) $$u^e(x) \quad \mathbf{N}^e(x)\mathbf{d}^e \quad \begin{bmatrix} N_1^e(x) & N_2^e(x) \end{bmatrix} \begin{bmatrix} u_1^e \\ u_2^e \end{bmatrix}$$

(2.9) $$\frac{du^e(x)}{dx} = \mathbf{B}^e\mathbf{d}^e = \frac{1}{l^e}\begin{bmatrix} -1 & 1 \end{bmatrix} \begin{bmatrix} u_1^e \\ u_2^e \end{bmatrix}$$

(2.10) $$w^e(x) \quad \mathbf{N}^e(x)\mathbf{w}^e \quad \begin{bmatrix} N_1^e(x) & N_2^e(x) \end{bmatrix} \begin{bmatrix} w_1^e \\ w_2^e \end{bmatrix}$$

(2.11) $$\frac{dw^e(x)}{dx} = \mathbf{B}^e\mathbf{w}^e = \frac{1}{l^e}\begin{bmatrix} -1 & 1 \end{bmatrix} \begin{bmatrix} w_1^e \\ w_2^e \end{bmatrix}$$

Substep (c). Substitute the trial solution and weight function and their derivatives in each element into the weak form to obtain element equations.

Substituting these equations into the weak form (2.2), we have:

(2.12) $$\sum_{e=1}^{n_{el}} \mathbf{w}^{eT} \left\{ \int_{x_1^e}^{x_2^e} \mathbf{B}^{eT} A^e E^e \mathbf{B}^e dx \; \mathbf{d}^e - \int_{x_1^e}^{x_2^e} \mathbf{N}^{eT} b \, dx - (\mathbf{N}^{eT} A^e \bar{t})_{x=0} \right\} = 0$$

We now define two **element matrices** that are very useful in FEM from (2.12):

(i) element stiffness matrix

(2.13) $$\mathbf{K}^e = \int_{x_1^e}^{x_2^e} \mathbf{B}^{eT} A^e E^e \mathbf{B}^e dx$$

(ii) element external force matrix

(2.14) 2 – 0 1

where is the element external body force matrix and is the element external boundary force matrix.

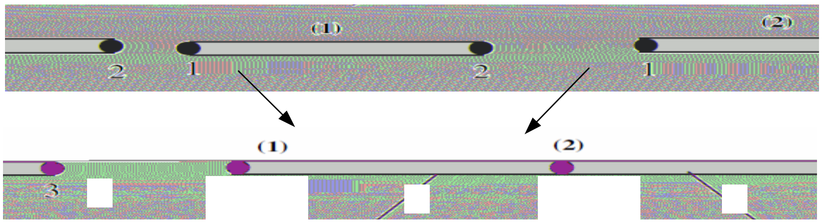
(Example) Consider a two-node linear element with a constant cross-sectional area and Young's modulus subjected to **linear distribution of body forces** as shown in the figure. Let us derive the element stiffness matrix and the external body force matrix for the element.

(Answer)

Remark: it can be seen that the sum of nodal forces acting on the element is $\frac{l^e(b_1+b_2)}{2}$, which is exactly the integral of the body force over the element domain.

Step 3: Assembly of element equations to obtain equations of the whole problem.

In deriving the element equations, we isolated a typical element e from the mesh and formulated the weak form and developed its element stiffness and force matrices. To obtain finite element equations of the total problem, we must put the elements and approximation solutions back into their original (global) view (i.e., reverse of **Step 1**).



Consider a mesh with n_{el} elements, the **global approximation** of trial solutions $u(x)$ and weight functions $w(x)$ can be obtained from **gathering** contributions from individual elements:

(2.15) $$u(x) = \sum_{e=1}^{n_{el}} \mathbf{N}^e(x)\mathbf{d}^e = \left(\sum_{e=1}^{n_{el}} \mathbf{N}^e(x)\mathbf{L}^e\right)\mathbf{d} = \mathbf{N}\mathbf{d}$$

(2.16) $$w\ x = \sum_{e=}^{n_{el}} \mathbf{N}^e\ x\ \mathbf{w}^e = \left(\sum_{e=}^{n_{el}} \mathbf{N}^e\ x\ \mathbf{L}^e\right)\mathbf{w} = \mathbf{N}\mathbf{w}$$

in which $\mathbf{L}^e$ is called **the gather matrix** $\mathbf{d}^e = \mathbf{L}^e\mathbf{d}$ (gather the nodal information from elements). $\mathbf{d}$ are **global** nodal values. Same thing holds for weight functions in which $\mathbf{w}$ are the **global** nodal weight values.

(2.17) $$\mathbf{d}^e = \mathbf{L}^e\mathbf{d} \quad \text{and} \quad \mathbf{w}^e \quad \mathbf{L}^e\mathbf{w}$$

Finite Element Method

03/06/2019

Q: For the two element problems, what are the gather matrix for element 1 ($\mathbf{L}^{(1)}$) and element 2 ($\mathbf{L}$)?

A:

Remarks:

1. Notice $\mathbf{d}^{(1)}$ $\mathbf{L}^{(1)}\mathbf{d}$ states that the element displacement at a node is the same as the corresponding global displacement. Since each global displacement is a unique and single value at a node, $\mathbf{d}^{(1)}$ $\mathbf{L}^{(1)}\mathbf{d}$ is equivalent to compatibility enforcement to ensure the solution $u(x)$ at the common node between elements is identical.
2. The gather matrix $\mathbf{L}^e$ contains only zeros and ones and is a Boolean matrix. They play an important role in developing matrix expressions relating element to global matrices. We do not perform "real" matrix operation for the gather matrix; we just do scatter and add.

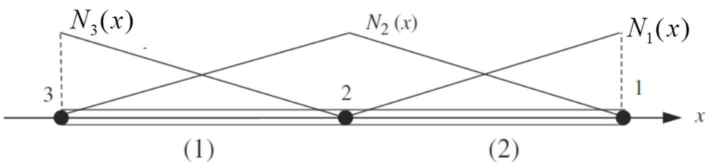
The **global shape functions** are defined as:

$$(2.18) \quad \mathbf{N} = \sum_{e=1}^{n_{el}} \mathbf{N}^e(x)\mathbf{L}^e$$

For a mesh with two elements, the global shape functions are:

$$\mathbf{N} = \mathbf{N}^{(1)}\mathbf{L}^{(1)} + \mathbf{N}^{(2)}\mathbf{L}^{(2)} \quad \text{or} \quad \begin{bmatrix} N_1 & N_2 & N_3 \end{bmatrix} = \begin{bmatrix} N_2^{(2)} & N_1^{(2)} + N_2^{(1)} & N_1^{(1)} \end{bmatrix}$$

Or view graphically,



The number of global shape functions is equal to the number of nodes.

Finite Element Method

03/06/2019

Q: What are finite element trial solutions for this simple case?

A:

(2.19)

Q: What is the **global nodal** value u_1? What are the **global nodal** values u_2 and u_3?

A:

Q: What are finite element weight functions for this simple case?

A:

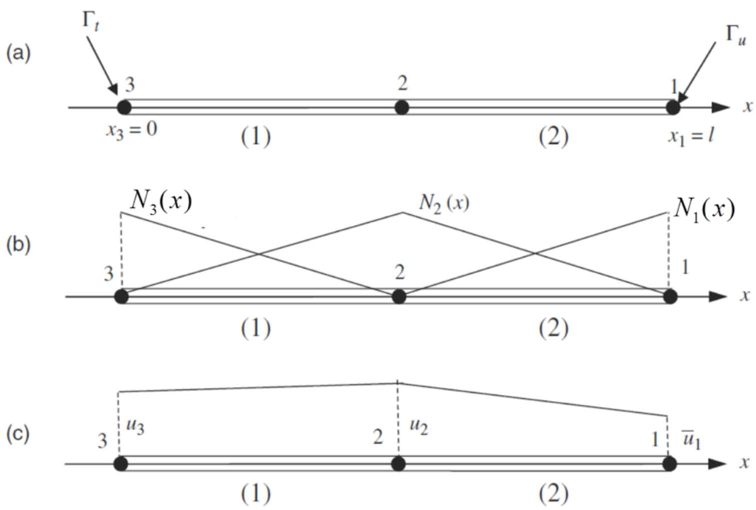
(2.20)

Q: What is the **global nodal** value w_1? What are the **global nodal** values w and w_3?

A:

Remark: it should be clear by now that instead of using polynomial basis with unknown coefficients to construct the approximation solution in the classical methods, finite element method uses **global interpolation functions constructed from element shape functions** $\mathbf{N} = \sum_{e=1}^{n_{el}} \mathbf{N}^e(x)\mathbf{L}^e$ and unknown nodal values to construct the approximation solution.

The classical way	The FEM way
$u(x) \approx \alpha_0 + \alpha_1 x + \alpha_2 x^2 ...$	$u(x) \approx \mathbf{N}(x)\mathbf{d}$
$w \ x \quad + \quad x + \quad x$	$w(x) \approx \mathbf{N}(x)\mathbf{w}$



Substituting (2.13) and (2.14) into (2.12) and using the relation from (2.17) gives:

$$\sum_{e=1}^{n_{el}} \mathbf{w}^T \mathbf{L}^{eT} \mathbf{K}^e \mathbf{L}^e \mathbf{d} \qquad \sum_{e=1}^{n_{el}} \mathbf{w}^T \mathbf{L}^{eT} \mathbf{f}^e \quad = 0$$

Q: Can we take $\mathbf{w}$ outside the summation? Why or why not?

$$\text{(2.21)} \qquad \mathbf{w}^T \left(\sum_{e=1}^{n_{el}} \mathbf{L}^{eT} \mathbf{K}^e \mathbf{L}^e \mathbf{d} - \sum_{e=1}^{n_{el}} \mathbf{L}^{eT} \mathbf{f}^e \right) \quad 0$$

A:

Moreover, in deriving (2.21), we have taken gather matrices $\mathbf{L}^e$ out of the integral since they are not function of x but they are element dependent.

We now define the system matrix **K** for differential equation (or stiffness matrix for stress analysis):

$$\text{(2.22)} \qquad \mathbf{K} = \sum_{e=1}^{n_{el}} \mathbf{L}^{eT} \mathbf{K}^e \mathbf{L}^e$$

Recall the gather matrix $\mathbf{L}^e$ only gathers the element displacements at nodes of each element from global displacement matrix. $\mathbf{L}^e$ contains only zeros and ones and is a Boolean matrix. Thus, the $\mathbf{K}$ matrix is assembled by the process of *matrix scatter and add*; you do not need to do triple matrix multiplication. Similarly, the assembled external force matrix is:

$$\text{(2.23)} \qquad \mathbf{f} = \sum_{e=1}^{n_{el}} \mathbf{L}^{eT} \mathbf{f}^e$$

The matrix is assembled by the process of **column** *matrix scatter and add* which is even easier than matrix assembly. We will see the example later and MATLAB codes to illustrate and perform these assembling processes.

Substituting (2.22) and (2.23) into (2.21), the weak form finally yields:

$$\text{(2.24)} \qquad \mathbf{w}^T \left(\mathbf{K}\mathbf{d} - \mathbf{f} \right) = 0 \qquad \forall \mathbf{w} \text{ except } w_1 = 0$$

Remark: (2.22) and (2.23) can be obtained with the same procedures from Direct Stiffness Method as we have introduced in *Advanced Structural Analysis*. You can consult Chapter 2

(2.1 and 2.2) of F&B for details.

Step 4: Imposition of boundary conditions of the problem.

Let

(2.25) $\mathbf{r} = \mathbf{Kd} - \mathbf{f}$

where **r** is the residual. We then have

(2.26) $w_2 r_2 + w_3 r_3 = 0$

where the first term has dropped out because $w_1 = 0$.

Q: From (2.26), what can we say about r r?

A:

Q: How about r_1?

A:

Eq. (2.25) is now:

$$(2.27) \quad \begin{bmatrix} K_{11} & K_{12} & K_{13} \\ K_{21} & K_{22} & K_{23} \\ K_{31} & K_{32} & K_{33} \end{bmatrix} \begin{bmatrix} \bar{u}_1 \\ u_2 \\ u_3 \end{bmatrix} = \begin{bmatrix} f_1 + r_1 \\ f_2 \\ f_3 \end{bmatrix}$$

Q: What are the unknowns to be solved in Eq. (2.27)?
A:

Step 5: Solution of the assembled equations.

We usually solve the unknown displacements first then back substitution to find the unknown reactions. Thus we find the unknown nodal displacements u_2, u_3 by:

(2.28) $$\begin{bmatrix} K_{22} & K_{23} \\ K_{32} & K_{33} \end{bmatrix} \begin{bmatrix} u_2 \\ u_3 \end{bmatrix} = \begin{bmatrix} f_2 - K_{21}\bar{u}_1 \\ f_3 - K_{31}\bar{u}_1 \end{bmatrix} \quad \Rightarrow \quad \begin{bmatrix} u_2 \\ u_3 \end{bmatrix} = \begin{bmatrix} K_{22} & K_{23} \\ K_{32} & K_{33} \end{bmatrix}^{-1} \begin{bmatrix} f_2 - K_{21}\bar{u}_1 \\ f_3 - K_{31}\bar{u}_1 \end{bmatrix}$$

and the unknown reaction follows:

(2.29) $$r_1 = \begin{bmatrix} K_{11} & K_{12} & K_{13} \end{bmatrix} \begin{bmatrix} \bar{u}_1 \\ u_2 \\ u_3 \end{bmatrix} - f_1$$

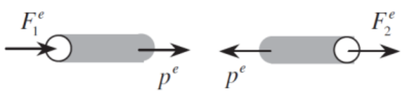
Step 6: Postprocessing of the results.

For purposes of postprocessing, the displacements and stresses are computed in **each element** using (2.8), (2.9) and the stress–strain law:

(2.30) $$u^e(x) = \mathbf{N}^e(x)\mathbf{d}^e \qquad \sigma^e(x) = E^e(x)\mathbf{B}^e\mathbf{d}^e$$

in which the element nodal values $\mathbf{d}^e$ are obtained by the gather operator using $\mathbf{d}^e = \mathbf{L}^e\mathbf{d}$.

For one-dimensional problems, we are often interested in finding member forces in each element which can be easily resolved from the stresses:



$$\sigma^e(x) = E^e(x)\mathbf{B}^e\mathbf{d}^e$$

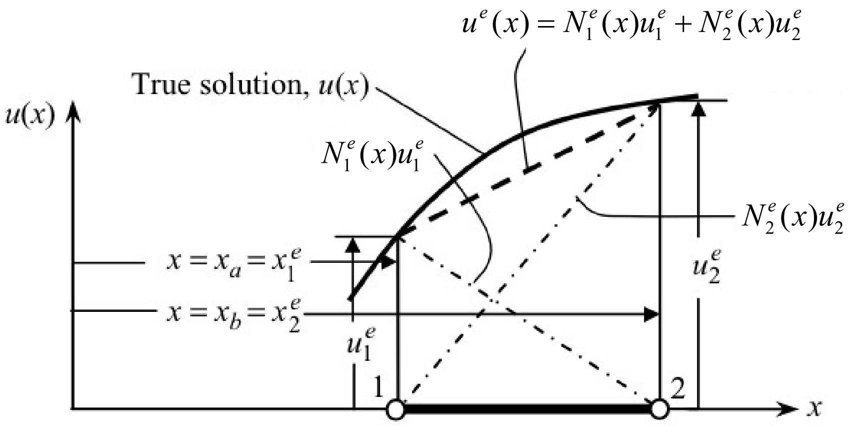
$$p^e = \sigma^e A^e = E^e A^e \mathbf{B}^e \mathbf{d}^e = \frac{E^e A^e}{l^e}\left(-u_1^e + u_2^e\right)$$

$$\begin{bmatrix} \ _1 \\ \ _2 \end{bmatrix} = \frac{\quad}{\quad} \begin{bmatrix} 1 & -1 \\ -1 & 1 \end{bmatrix} \begin{bmatrix} \ _1 \\ \ _2 \end{bmatrix}$$

$$\mathbf{F}^e \quad \mathbf{K}^e\mathbf{d}^e$$

Reviews, Highlights and Checkpoints

1. The finite element method treats a given domain as a collection of simple subdomains, called **finite elements**. Since the subdomain is relatively simple, it is possible to systematically construct the approximation functions needed in the weighted-residual or variational methods over **each element**.
2. Shape functions with the **Kronecker delta property** are used to construct the elementwise approximation functions $u^e(x)$.



3. Finite element method uses **global interpolation functions constructed from element shape functions** $\mathbf{N} = \sum_{e=1}^{n_{el}} \mathbf{N}^e(x)\mathbf{L}^e$ and unknown nodal values to construct the approximation solution. This differs from the classical way in which polynomial basis with unknown coefficients is used to construct the approximation solution

The classical way	The FEM way
$u(x) \approx \alpha_0 + \alpha_1 x + \alpha_2 x^2 \ldots$	$u(x) \approx \mathbf{N}(x)\mathbf{d}$
$w \;\; x \qquad + \;\; x + \;\; x$	$w(x) \approx \mathbf{N}(x)\mathbf{w}$

2.3 Formulation of Discrete Finite Element Equations: 1D Elastic Bar with Arbitrary Boundary Conditions

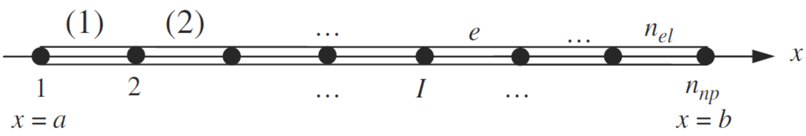
We will now consider the weak form of stress analysis in 1D with arbitrary boundary conditions and finite elements with general discretization (mesh). Recall Eq. (1.11):

(1.11)

Find $u(x)$ among the smooth functions that satisfy $u = \bar{u}$ on Γ_u such that

$$\int_{\Omega} \left(\frac{dw}{dx}\right)^T AE \frac{du}{dx} dx - \int_{\Omega} w^T b \, dx - (w^T A \bar{t})\big|_{\Gamma_t} = 0 \qquad w = 0 \text{ on } \Gamma_u$$

And consider the following finite element mesh:



We express the weak form from an integral over the entire domain Ω to a sum of integrals over element domains:

(2.31) $$\sum_{e=}^{n_{el}} \left\{ \int_{\Omega^e} \left(\frac{dw^e}{dx}\right)^T A^e E^e \left(\frac{du^e}{dx}\right) dx - \int_{\Omega^e} w^{eT} b \, dx - \; w^{eT} A^e \bar{t} \;\; {}_{\Gamma_t^e} \right\} =$$

Recall (2.8) – (2.11):

(2.8) $$u^e(x) \quad \mathbf{N}^e(x)\mathbf{d}^e \quad \begin{bmatrix} N_1^e(x) & N_2^e(x) \end{bmatrix} \begin{bmatrix} u_1^e \\ u_2^e \end{bmatrix}$$

(2.9) $$\frac{du^e(x)}{dx} = \mathbf{B}^e \mathbf{d}^e = \frac{1}{l^e} \begin{bmatrix} -1 & 1 \end{bmatrix} \begin{bmatrix} u_1^e \\ u_2^e \end{bmatrix}$$

(2.10) $$w^e(x) \quad \mathbf{N}^e(x)\mathbf{w}^e \quad \begin{bmatrix} N_1^e(x) & N_2^e(x) \end{bmatrix} \begin{bmatrix} w_1^e \\ w_2^e \end{bmatrix}$$

(2.11) $$\frac{dw^e(x)}{dx} = \mathbf{B}^e \mathbf{w}^e = \frac{1}{l^e} \begin{bmatrix} -1 & 1 \end{bmatrix} \begin{bmatrix} w_1^e \\ w_2^e \end{bmatrix}$$

Substituting them into (2.31) gives:

(2.32) $$\sum_{e=}^{n_{el}} \mathbf{w}^{eT} \left\{ \int_{\Omega^e} \mathbf{B}^{eT} A^e E^e \mathbf{B}^e dx \; \mathbf{d}^e - \int_{\Omega^e} \mathbf{N}^{eT} b \, dx - \; \mathbf{N}^{eT} A^e \bar{t} \;\; {}_{\Gamma_t^e} \right\} =$$

Recall (2.13), (2.14) and (2.17):

(2.13) $\mathbf{K}^e = \int_{x_1^e}^{x_2^e} \mathbf{B}^{eT} A^e E^e \mathbf{B}^e dx$

(2.14) $\mathbf{f}^e = \mathbf{f}_\Omega^e + \mathbf{f}_\Gamma^e = \int_{x_1^e}^{x_2^e} \mathbf{N}^{eT} b \, dx + (\mathbf{N}^{eT} A^e \overline{t})_{\Gamma_t^e}$

(2.17) $\mathbf{d}^e = \mathbf{L}^e \mathbf{d}$ and $\mathbf{w}^e \quad \mathbf{L}^e \mathbf{w}$

Substituting them into (2.32) gives:

(2.33) $\mathbf{w} \left(\sum_{=1} \mathbf{L} \quad \mathbf{K} \; \mathbf{L} \quad \mathbf{d} - \sum_{=1} \mathbf{L} \quad \mathbf{f} \quad \right) = 0$

(2.34) $\mathbf{w}^T \left(\mathbf{Kd} - \quad \right) =$

(2.35) $\mathbf{w}^T \mathbf{r} = 0$
where $\mathbf{r} = \mathbf{Kd} - \mathbf{f}$ as in (2.25).

We now partition the **global** solution and weight function matrices as:

$$\mathbf{d} = \begin{bmatrix} \overline{\mathbf{d}}_E \\ \mathbf{d}_F \end{bmatrix} \qquad\qquad \mathbf{w} = \begin{bmatrix} \mathbf{w}_E \\ \mathbf{w}_F \end{bmatrix} = \begin{bmatrix} \\ \mathbf{w}_F \end{bmatrix}$$

The part of the matrix denoted by the subscript '*E*' contains the **nodal values on the essential boundaries**. As indicated by the overbar on $\overline{\mathbf{d}}_E$, these values of the solution are set to satisfy the essential boundary conditions, so they are known values. The submatrices denoted by the subscript '*F*' contain **all the remaining nodal values: these entries are arbitrary for the weight function and unknown for the trial solution.**

Partitioning **r** in (2.35) congruent with **w** gives:

(2.36) $\mathbf{w}_E^T \mathbf{r}_E + \mathbf{w}_F^T \mathbf{r}_F = 0$

Q: What can we say on $\mathbf{w}_E$ and $\mathbf{w}_F$?

A:

Q: What can we say on $\mathbf{r}_E$ and $\mathbf{r}_F$?

A:

We thus have:

$$(2.37) \qquad \begin{bmatrix} \mathbf{K}_E & \mathbf{K}_{EF} \\ \mathbf{K}_{EF}^T & \mathbf{K}_F \end{bmatrix} \begin{bmatrix} \overline{\mathbf{d}}_E \\ \mathbf{d}_F \end{bmatrix} = \begin{bmatrix} \mathbf{f}_E + \mathbf{r}_E \\ \mathbf{f}_F \end{bmatrix}$$

Using the two-step approach discussed previously, we first solve for the unknown discrete solution ${}_F$ by using the second row in (2.37):

$$(2.38) \qquad \mathbf{K}_F \mathbf{d}_F = \mathbf{f}_F - \mathbf{K}_{EF}^T \overline{\mathbf{d}}_E \qquad \Rightarrow \qquad \mathbf{d}_F = \left(\mathbf{K}_F\right)^{-} \ \mathbf{f}_F - \mathbf{K}_{EF}^T \overline{\mathbf{d}}_E$$

Once ${}_F$ is known, the unknown reactions $\mathbf{r}_E$ can be computed from the first row of (2.37):

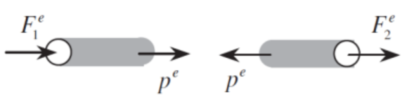
$$(2.39) \qquad \mathbf{r}_E = \mathbf{K}_E \overline{\mathbf{d}}_E + \mathbf{K}_{EF} \mathbf{d}_F - \mathbf{f}_E$$

Postprocessing follows (2.30):

$$(2.30) \qquad u^e(x) = \mathbf{N}^e(x)\mathbf{d}^e \qquad \sigma^e(x) = E^e(x)\mathbf{B}^e\mathbf{d}^e$$

in which the element nodal values $\mathbf{d}^e$ are obtained by the gather operator using $\mathbf{d}^e = \mathbf{L}^e\mathbf{d}$.

For one-dimensional problems, we are often interested in finding <u>member forces in each element</u> which can be easily resolved from the stresses:



$$\sigma^e(x) = E^e(x)\mathbf{B}^e\mathbf{d}^e$$

$$p^e = \sigma^e A^e = E^e A^e \mathbf{B}^e \mathbf{d}^e = \frac{E^e A^e}{l^e}\left(-u_1^e + u_2^e\right)$$

$$\begin{bmatrix} _1 \\ _2 \end{bmatrix} = \text{——} \begin{bmatrix} 1 & -1 \\ -1 & 1 \end{bmatrix} \begin{bmatrix} _1 \\ _2 \end{bmatrix}$$

$\mathbf{F}^e \quad \mathbf{K}^e \mathbf{d}^e$

Example: recall Examples 1 and 2(a) in Chapter 1, we have the strong form:

(a) $\frac{d}{dx} \quad AE\frac{du}{dx} \quad = - \quad Ax \qquad < x <$

(b) $u_{x=0} = u(0) = 10^{-4}$

(c) $\sigma_{\ =2} = \left(\quad \text{—} \right)_{=2} = 10$

The weak form

$$\int_0^2 AE\frac{dw}{dx}\frac{du}{dx}dx - \int_0^2 10wAxdx - 10\left(wA\right)_{x=2} = 0 \quad \forall w \text{ with } w(0) = 0$$

And we need to find $u(x)$ among the smooth functions that satisfy $u(0) = 10^{-4}$ such that the weak form holds for all smooth $w(x$ with $w(0) = 0$.

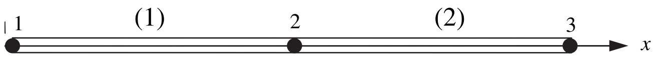
The classical way of approximation (let A be 1.0 and $E =$)

$u \ x \ = \qquad + \quad x \qquad \frac{du \ x}{dx} =$

$w \ x \ = \quad x \qquad\qquad \frac{dw \ x}{dx} =$

$$u(x) = 10^{-4}\left(1 + \frac{7}{3}x\right)$$

$$\sigma(\ ) = \quad \text{—} = 10^5 \times 10^{-4} \text{—} \left(1 + \frac{7}{3} \quad \right) = \frac{70}{3}$$

Perform FEM way of approximation using two two-node linear elements for the problem.

Ans:

03/06/2019

Step1: Discretization of the given domain into a collection of preselected finite elements.

Step 2: Derivation of element equations for all typical elements in the mesh. For each element, compute stiffness matrix $\mathbf{K}^e$, and force vector $\mathbf{f}^e$.

The two-node element stiffness matrix is

$$\mathbf{K}^e = \int_{x_1^e}^{x_2^e} \mathbf{B}^{eT} A^e E^e \mathbf{B}^e dx = \int_{x_1^e}^{x_2^e} \frac{1}{l^e}\begin{bmatrix}-1\\1\end{bmatrix} A^e E^e \frac{1}{l^e}\begin{bmatrix}-1 & 1\end{bmatrix} dx = \frac{A^e E^e}{\left(l^e\right)^2}\begin{bmatrix}-1\\1\end{bmatrix}\begin{bmatrix}-1 & 1\end{bmatrix}\int_{x_1^e}^{x_2^e} dx$$

$$= \frac{A^e E^e}{l^e}\begin{bmatrix}1 & -1\\-1 & 1\end{bmatrix}$$

For element 1

$$x_1^{(1)} = 0.0,\ x_2^{(1)} = 1.0,\ l^e = 1.0,\ A^e E^e = 10^5$$

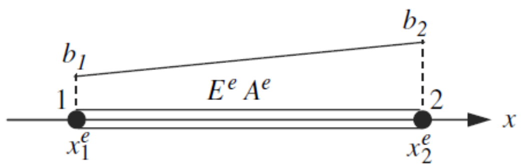
$$\mathbf{K}^{(1)} = 10^5\begin{bmatrix}1 & -1\\-1 & 1\end{bmatrix}$$

For element 2

$$x \quad = \qquad x \quad = \qquad l^e = \qquad A^e E^e =$$

$$\mathbf{K} \quad = \quad \begin{bmatrix} & -\\ - & \end{bmatrix}$$

Element external force matrix can be established by noting the linear distribution of the body force. Recall



$$b(x) = \mathbf{N}^e\mathbf{b}, \qquad \mathbf{b} = \begin{bmatrix}b_1\\b_2\end{bmatrix}$$

$$\mathbf{f}_\Omega^e = \int_{x_1^e}^{x_2^e} \mathbf{N}^{eT}\mathbf{N}dx\ \mathbf{b} = \frac{1}{\left(l^e\right)^2}\int_{x_1^e}^{x_2^e}\begin{bmatrix}(x_2^e - x)^2 & (x_2^e - x)(x - x_1^e)\\(x_2^e - x)(x - x_1^e) & (x - x_1^e)^2\end{bmatrix} dx\ \mathbf{b}$$

$$= \frac{l^e}{6}\begin{bmatrix}2 & 1\\1 & 2\end{bmatrix}\begin{bmatrix}b_1\\b_2\end{bmatrix}$$

$$= \frac{l^e}{6}\begin{bmatrix}2b_1 + b_2\\b_1 + 2b_2\end{bmatrix}$$

<u>For element 1</u>

$b_1 = 0,\ b_2 = 10,\ l^e = 1.0$

$$\mathbf{f}_\Omega^{(1)} = \frac{1}{6}\begin{bmatrix} 10 \\ 20 \end{bmatrix}$$

<u>For element 2</u>

$b_1 = 10,\ b_2 = 20,\ l^e = 1.0$

$$\mathbf{f}_\Omega^{(2)} = \frac{1}{6}\begin{bmatrix} 40 \\ 50 \end{bmatrix}$$

<u>**Step 3**: Assembly of element equations to obtain equations of the whole problem $\mathbf{Kd} = \mathbf{f}$.</u>

The global stiffness matrix is obtained by the matrix assembly operation:

$$\mathbf{K} = \sum_{e=1}^{n_{el}} \mathbf{L}^{eT}\mathbf{K}^e\mathbf{L}^e = \mathbf{L}^{(1)T}\mathbf{K}^{(1)}\mathbf{L}^{(1)} + \mathbf{L}^{(2)T}\mathbf{K}^{(2)}\mathbf{L}^{(2)}$$

We can use direct matrix assembly (as shown in sequel) to obtain it, but to show that the two procedures are identical we will first obtain the global stiffness matrix by the above equation.

The gather operator for the two elements are:

$$\mathbf{d}^{(1)} = \begin{bmatrix} u_1^{(1)} \\ u_2^{(1)} \end{bmatrix} = \begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \end{bmatrix}\begin{bmatrix} u_1 \\ u_2 \\ u_3 \end{bmatrix} \quad \Rightarrow \quad \mathbf{L}^{(1)} = \begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \end{bmatrix}$$

$$\mathbf{d}^{(2)} = \begin{bmatrix} u_1^{(2)} \\ u_2^{(2)} \end{bmatrix} = \begin{bmatrix} 0 & 1 & 0 \\ 0 & 0 & 1 \end{bmatrix}\begin{bmatrix} u_1 \\ u_2 \\ u_3 \end{bmatrix} \quad \Rightarrow \quad \mathbf{L}^{(2)} = \begin{bmatrix} 0 & 1 & 0 \\ 0 & 0 & 1 \end{bmatrix}$$

Thus, the global stiffness matrix is

$$\mathbf{K} = \mathbf{L}^{(1)T}\mathbf{K}^{(1)}\mathbf{L}^{(1)} + \mathbf{L}^{(2)T}\mathbf{K}^{(2)}\mathbf{L}^{(2)}$$

$$= \begin{bmatrix} 1 & 0 \\ 0 & 1 \\ 0 & 0 \end{bmatrix} 10^5 \begin{bmatrix} 1 & -1 \\ -1 & 1 \end{bmatrix} \begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \end{bmatrix} + \begin{bmatrix} 0 & 0 \\ 1 & 0 \\ 0 & 1 \end{bmatrix} 10^5 \begin{bmatrix} 1 & -1 \\ -1 & 1 \end{bmatrix} \begin{bmatrix} 0 & 1 & 0 \\ 0 & 0 & 1 \end{bmatrix}$$

$$= 10^5 \begin{bmatrix} 1 & -1 & 0 \\ -1 & 1 & 0 \\ 0 & 0 & 0 \end{bmatrix} + 10^5 \begin{bmatrix} 0 & 0 & 0 \\ 0 & 1 & -1 \\ 0 & -1 & 1 \end{bmatrix}$$

$$= 10^5 \begin{bmatrix} 1 & -1 & 0 \\ -1 & 2 & -1 \\ 0 & -1 & 1 \end{bmatrix}$$

In practice (and in MATLAB implementation), the above triple products are not performed and a direct assembly by id is employed. The process is shown below

$$\mathbf{K}^{(1)} = 10^5 \begin{bmatrix} 1 & -1 \\ -1 & 1 \end{bmatrix} \begin{matrix} [1] \\ [2] \end{matrix}$$
$$\qquad\qquad [1] \quad [2]$$

$$\mathbf{K}^{(2)} = 10^5 \begin{bmatrix} 1 & -1 \\ -1 & 1 \end{bmatrix} \begin{matrix} [2] \\ [3] \end{matrix}$$
$$\qquad\qquad [2] \quad [3]$$

$$\mathbf{K} = 10^5 \begin{bmatrix} 1 & -1 & 0 \\ -1 & 2 & -1 \\ 0 & -1 & 1 \end{bmatrix} \begin{matrix} [1] \\ [2] \\ [3] \end{matrix}$$
$$\qquad\qquad [1] \quad [2] \quad [3]$$

The load is prescribed at node 3. We can do direct assembly by adding the contribution from the boundary with those from body force:

$$\mathbf{f} = \begin{bmatrix} 0 \\ 0 \\ 10 \end{bmatrix} + \frac{1}{6} \begin{bmatrix} 10 \\ 60 \\ 50 \end{bmatrix} = \begin{bmatrix} \frac{10}{6} \\ 10 \\ \frac{110}{6} \end{bmatrix}$$

<u>**Step 4**: Imposition of essential boundary conditions of the problem.</u>

Step 5: Solution of the assembled equations and obtain the global displacements

The global system of equations is given by

$$10^5 \begin{bmatrix} 1 & -1 & 0 \\ -1 & 2 & -1 \\ 0 & -1 & 1 \end{bmatrix} \begin{bmatrix} 10^{-4} \\ u_2 \\ u_3 \end{bmatrix} = \begin{bmatrix} \frac{10}{6} \\ 10 \\ \frac{110}{6} \end{bmatrix} + \begin{bmatrix} r_1 \\ 0 \\ 0 \end{bmatrix} = \begin{bmatrix} \frac{10}{6} + r_1 \\ 10 \\ \frac{110}{6} \end{bmatrix}$$

Since node 1 is on the essential boundary, we partition after the first row, which gives

$$"\&^{\#} \begin{bmatrix} ! & -" \\ -" & " \end{bmatrix} \begin{bmatrix} u_{!} \\ u_{\%} \end{bmatrix} = \begin{bmatrix} \& - K_{!"}\bar{u} \\ "\& - K_{\%"}\bar{u} \end{bmatrix})* \begin{bmatrix} "\&+"\& \\ \frac{"\&\&}{+} \end{bmatrix}))))) \Rightarrow) \begin{bmatrix} u_{!} \\ u_{\%} \end{bmatrix} = "\&^{-\$} \begin{bmatrix} \%'(\%\% \\ \#'++, \end{bmatrix}$$

<u>Evaluation of solution quality</u>

The exact solution is given by:

$$u^{ex}(x) = 10^{-4}\left(1 + 3x - \frac{x^3}{6}\right)$$

The classical approximation with linear polynomial gives us

$$u^{L}(x) = 10^{-4}\left(1 + \frac{7}{3}x\right)$$

We can do a simple MATLAB plot to compare the difference between exact, classical linear approximation and FEM approximation with two linear elements

```
clf;clear All;
x=0:0.01:2;
u1=(1+7/3*x)/10000;
ue=(1+3*x-1/6*x.^3)/10000;

plot(x,u1,'k--');
hold on;

Kf = 1e5*[2 -1; -1 1];
F = [20 110/6]';
u = Kf\F;

femx = [0 1 2];
femy = [1e-4 u(1) u(2)];
plot(femx,femy,'ko-.');
hold on;

plot(x,ue,'k-');
hold on;

legend('u linear','u fem', 'u exact');
```

```
xlabel('x');ylabel('displacement, u');
```

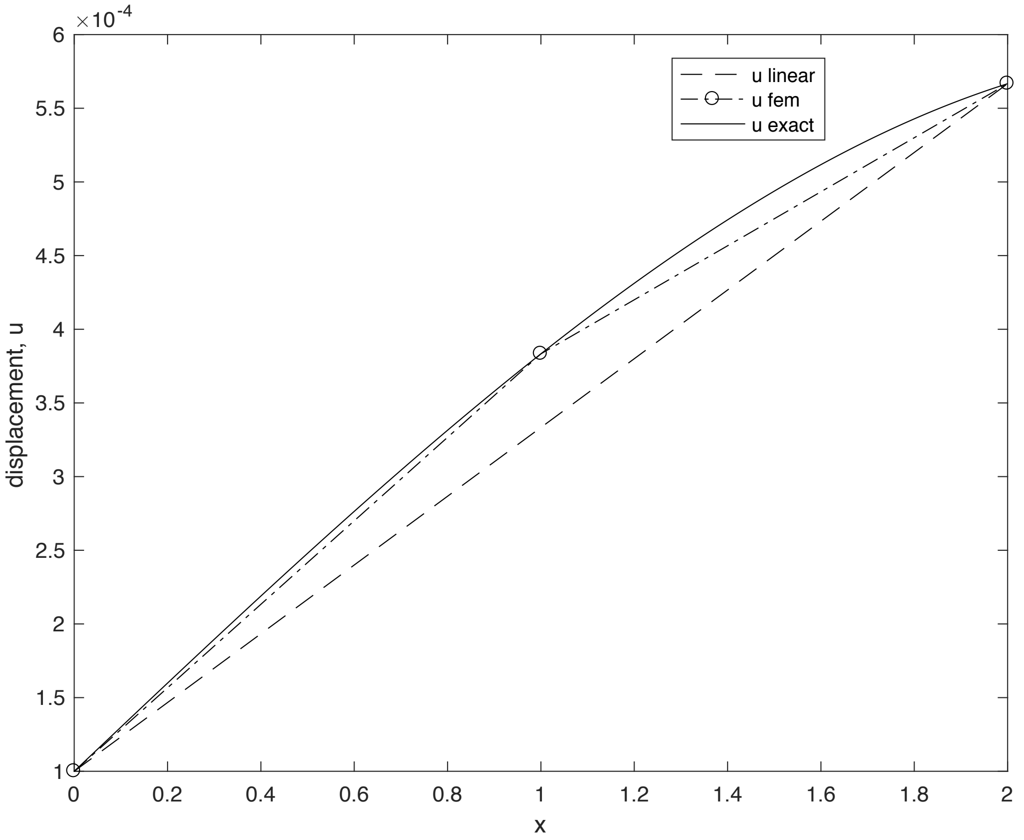


Figure above compares the FEM solution with the exact solution and classical linear approximation. It can be seen that the nodal displacements for the FEM solution are **exact**. This is an unusual anomaly of finite element solutions in one dimension and does not occur in multidimensional solutions. It is explained in Hughes (1987) Section 1.10 Mathematical Analysis, available from the course website. Note that the essential boundary condition is satisfied exactly. This is not surprising as the trial solution was constructed so as to satisfy the essential boundary condition. In finite element solutions, essential boundary conditions will always be satisfied exactly.