Lesson 02

Basic Mechanics, Discrete Systems

Minh-Chien Trinh, PhD

Division of Mechanical System Engineering
Jeonbuk National University

March 15, 2023

Content

1. Continuum Mechanics – Gradient, Divergence and Curl

$\mathbf{x} = \vec{x} = x_i \mathbf{e_i} = x_i \vec{e}_i$ is a position vector (spatial coordinates, rank-1 tensor).

$\phi(\mathbf{x})$ is a scalar field (rank-0 tensor).

$\mathbf{u}(\mathbf{x}) = \vec{u}(\mathbf{x})$ is a vector field (rank-1 tensor).

$\mathbf{A} = A_{ij} \vec{e}_i \otimes \vec{e}_j$ is a matrix field (rank-2 tensor).

- Gradient is considered a vector $\nabla = \dfrac{\partial}{\partial \vec{x}} = \vec{e}_i \dfrac{\partial}{\partial x_i}$
- Gradient of a scalar field is a vector $\nabla \phi = \text{grad}\phi = \vec{e}_i \dfrac{\partial \phi}{\partial x_i} = \phi_{,i} \vec{e}_i$
- Gradient of a vector field is a rank-2 tensor $\nabla \mathbf{u} = \left(\vec{e}_i \dfrac{\partial}{\partial x_i} \right) \otimes u_j \vec{e}_j = \dfrac{\partial u_j}{\partial x_i} \vec{e}_i \otimes \vec{e}_j = u_{j,i} \vec{e}_i \otimes \vec{e}_j$
- Divergence of a vector field is a scalar $\nabla \cdot \mathbf{u} = \left(\vec{e}_i \dfrac{\partial}{\partial x_i} \right) \cdot u_j \vec{e}_j = \dfrac{\partial u_j}{\partial x_i} \delta_{ij} = \dfrac{\partial u_i}{\partial x_i} = \dfrac{\partial u_1}{\partial x_1} + \dfrac{\partial u_2}{\partial x_2} + \dfrac{\partial u_3}{\partial x_3}$
- Divergence of rank-2 tensor is a rank-1 tensor $\nabla \cdot \mathbf{A} = \left(\vec{e}_i \dfrac{\partial}{\partial x_i} \right) \cdot A_{jk} \vec{e}_j \otimes \vec{e}_k = \dfrac{\partial A_{jk}}{\partial x_i} \delta_{ij} \vec{e}_k = \dfrac{\partial A_{jk}}{\partial x_j} \vec{e}_k$
- Laplace is the inner product two gradients (δ_{ij} is the Kronecker delta)

$$\nabla^2 = \nabla \cdot \nabla = \left(\vec{e}_i \frac{\partial}{\partial x_i} \right) \cdot \left(\vec{e}_i \frac{\partial}{\partial x_i} \right) = \frac{\partial}{\partial x_i} \frac{\partial}{\partial x_j} \delta_{ij} = \frac{\partial}{\partial x_j} \frac{\partial}{\partial x_j} = \frac{\partial^2}{\partial x_1^2} + \frac{\partial^2}{\partial x_2^2} + \frac{\partial^2}{\partial x_3^2}$$

- Curl of a vector (ε_{ijk} is the permutation symbol) $\nabla \times \mathbf{u} = \left(\vec{e}_i \dfrac{\partial}{\partial x_i} \right) \times u_j \vec{e}_j = \dfrac{\partial u_j}{\partial x_i} \varepsilon_{ijk} \vec{e}_k$

1. Continuum Mechanics – Integral Theorems, Integration-by-Parts

Integral Theorems

Ω	is a volume domain	$\mathbf{A}$	is a tensor
Γ	is a closed surface of the domain	$\mathbf{n}$	is a unit outward normal vector

- Gradient Theorem $\int_\Omega \nabla\phi\, d\Omega = \oint_\Gamma \mathbf{n}\,\phi\, ds$
- Divergence Theorem $\int_\Omega \nabla\cdot\mathbf{A}\, d\Omega = \oint_\Gamma \mathbf{n}\cdot\mathbf{A}\, ds$
- Curl Theorem $\int_\Omega \nabla\times\mathbf{A}\, d\Omega = \oint_\Gamma \mathbf{n}\times\mathbf{A}\, ds$

Integration-by-Parts

$u(x)$ and $v(x)$ are continuously differentiable functions.

- 1D: $\int_a^b u(x)v'(x)\,dx = \left[u(x)v(x)\right]_a^b - \int_a^b u'(x)v(x)\,dx$
- 2D, 3D: $\int_\Omega \frac{\partial u}{\partial x_i} v\,\mathrm{d}\Omega = \int_\Gamma u\,v\,n_i\,\mathrm{d}\Gamma - \int_\Omega u\frac{\partial v}{\partial x_i}\mathrm{d}\Omega$
- For a vector field **v**(x): $\int_\Omega \nabla u\cdot\mathbf{v}\,\mathrm{d}\Omega = \int_\Gamma u(\mathbf{v}\cdot\mathbf{n})\,\mathrm{d}\Gamma - \int_\Omega u\nabla\cdot\mathbf{v}\,\mathrm{d}\Omega$
- Green's identity: $\int_\Omega \nabla u\cdot\nabla v\,\mathrm{d}\Omega = \int_\Gamma u\nabla v\cdot\mathbf{n}\,\mathrm{d}\Gamma - \int_\Omega u\nabla^2 v\,\mathrm{d}\Omega$

1. Continuum Mechanics – Kinematics, Equilibrium, Constitutive Eq.

Kinematics

❖ Displacement vector $\mathbf{u} = \vec{u} = (u_1, u_2, u_3)$

❖ Strain tensor (symmetric) $\boldsymbol{\varepsilon} = \frac{1}{2}\left[\nabla\mathbf{u} + (\nabla\mathbf{u})^T\right]$

$$\varepsilon_{ij} = \frac{1}{2}\left(\frac{\partial u_i}{\partial x_j} + \frac{\partial u_j}{\partial x_i}\right) = \frac{1}{2}\left(u_{i,j} + u_{j,i}\right)$$ (6 equations) (9 unknowns)

Equilibrium Equations

❖ 2nd order stress tensor (symmetric) σ_{ij}

❖ Equilibrium equations $\sigma_{ij,j} + \rho f_i = 0$ (3 equations) (6 unknowns)

❖ Cauchy formula $t_i = \sigma_{ji} n_j$

Constitutive Equations

❖ Stress and strain relation $\sigma_{ij} = C_{ijkl}\varepsilon_{kl}$ (6 equations)

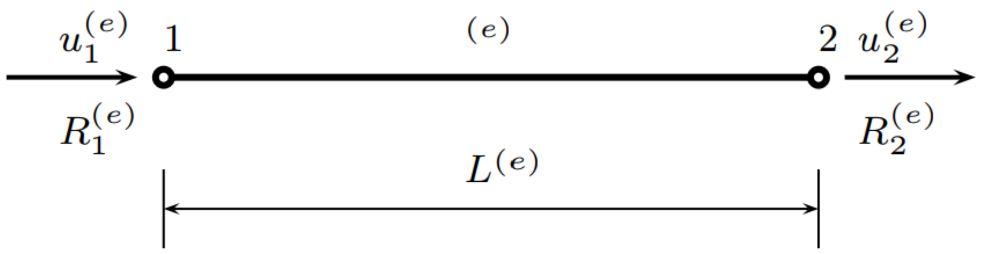
Total Equations in Elasticity

(15 equations) (15 unknowns)

2. Discrete Systems – Springs and Bars

❖ A bar (or spring) element with 2 nodes, 1 degree of freedom per node (axial displacement)

$u_1^{(e)}$	Axial displacement of node 1
$u_2^{(e)}$	Axial displacement of node 2
$L^{(e)}$	Length of an element
A	Constant cross-section
E	Modulus of elasticity



"2-node bar element under axial forces only"

❖ The deformation in the bar

$$\varepsilon = \frac{u_2^{(e)} - u_1^{(e)}}{L^{(e)}}$$

❖ The stress in the bar is computed by Hooke's law

$$\sigma = E^{(e)}\varepsilon = E^{(e)}\frac{u_2^{(e)} - u_1^{(e)}}{L^{(e)}}$$

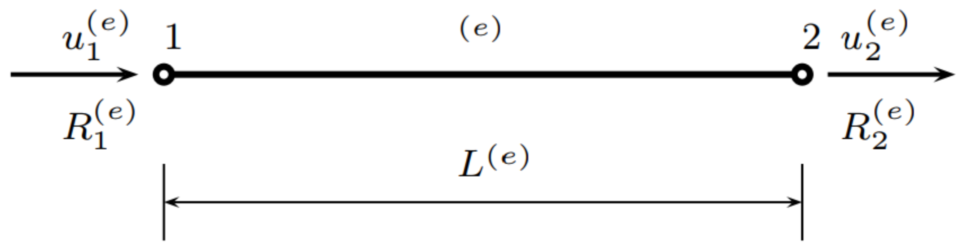
❖ The axial resultant force is obtained by integration of stresses on the cross-section area

$$N = A^{(e)}\sigma = A^{(e)}\left(E^{(e)}\varepsilon\right) = \left(EA\right)^{(e)}\frac{u_2^{(e)} - u_1^{(e)}}{L^{(e)}}$$

❖ Static equilibrium of the axial forces

$$R_2^{(e)} = -R_1^{(e)} = N = \left(\frac{EA}{L}\right)^{(e)}\left(u_2^{(e)} - u_1^{(e)}\right) = k^{(e)}\left(u_2^{(e)} - u_1^{(e)}\right)$$

2. Discrete Systems – Springs and Bars



❖ Rewrite the static equilibrium as

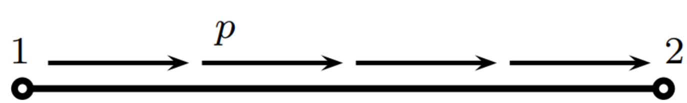
$$\mathbf{q}^{(e)} = \begin{Bmatrix} R_1^{(e)} \\ R_2^{(e)} \end{Bmatrix} = k^{(e)} \begin{bmatrix} 1 & -1 \\ -1 & 1 \end{bmatrix} \begin{Bmatrix} u_1^{(e)} \\ u_2^{(e)} \end{Bmatrix} = \mathbf{K}^{(e)} \mathbf{u}^{(e)}$$

$\mathbf{K}^{(e)}$ is the stiffness matrix of the bar element

$\mathbf{u}^{(e)}$ is the displacement vector of the bar element

$\mathbf{q}^{(e)}$ is the vector of nodal forces

❖ If uniformly distributed forces exist, we need to transform those forces into nodal forces as



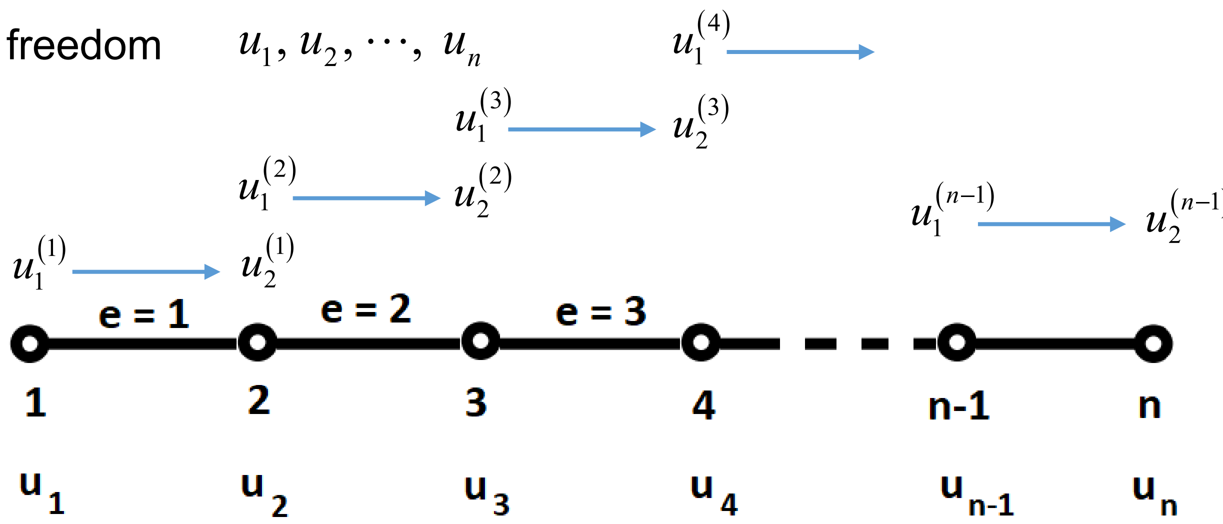
$$\mathbf{q}^{(e)} = k^{(e)} \begin{bmatrix} 1 & -1 \\ -1 & 1 \end{bmatrix} \begin{Bmatrix} u_1^{(e)} \\ u_2^{(e)} \end{Bmatrix} - \frac{(pL)^{(e)}}{2} \begin{Bmatrix} 1 \\ 1 \end{Bmatrix} = \mathbf{K}^{(e)} \mathbf{u}^{(e)} - \mathbf{f}^{(e)}$$

$\mathbf{f}^{(e)}$ is the vector of nodal forces equivalent to distributed forces p

2. Discrete Systems – Equilibrium at Nodes

- The structure should also be in equilibrium. Thus, we need to assemble all elements to obtain a global system of equations.

- System degrees of freedom $u_1, u_2, \cdots, u_n$



- Example: Assembly element (3) stiffness into the system stiffness matrix.

$$\mathbf{K}^{(3)} = \begin{array}{cc} u_3 & u_4 \\ \end{array}$$
$$\mathbf{K}^{(3)} = \begin{array}{c c} \begin{bmatrix} k_{11}^{(3)} & k_{12}^{(3)} \\ k_{21}^{(3)} & k_{22}^{(3)} \end{bmatrix} & \begin{array}{c} u_3 \\ u_4 \end{array} \end{array} \Rightarrow \mathbf{K}^{(3)} = \begin{bmatrix} K_{33}^{(3)} & K_{34}^{(3)} \\ K_{43}^{(3)} & K_{44}^{(3)} \end{bmatrix} \Rightarrow \mathbf{K} = \begin{array}{cc} \begin{bmatrix} \cdots & \cdots & \cdots & \cdots & \cdots \\ \cdots & \cdots & \cdots & \cdots & \cdots \\ \cdots & \cdots & K_{33}^{(3)} + & K_{34}^{(3)} + & \cdots \\ \cdots & \cdots & K_{43}^{(3)} + & K_{44}^{(3)} + & \cdots \\ \cdots & \cdots & \cdots & \cdots & \cdots \end{bmatrix} & \begin{array}{c} \\ \\ u_3 \\ u_4 \\ \\ \end{array} \end{array}$$

(Column labels: element matrix columns u_3, u_4; system matrix third and fourth columns u_3, u_4.)

< Element index >　< System index >　< System matrix >

2. Discrete Systems – Equilibrium at Nodes

- At the node j, the sum of all forces arising from various adjacent elements equals the applied load at that node j.

$$\sum_{e=1}^{n_e} R^{(e)} = f_j$$

$n^{(e)}$ is the number of elements in the structure

- A global system of equations

$$\begin{bmatrix} K_{11} & K_{12} & \dots & K_{1n} \\ K_{21} & K_{22} & \dots & K_{2n} \\ \dots & \dots & \dots & \dots \\ K_{n1} & K_{n2} & \dots & K_{nn} \end{bmatrix} \begin{Bmatrix} u_1 \\ u_2 \\ \dots \\ u_n \end{Bmatrix} = \begin{Bmatrix} f_1 \\ f_2 \\ \dots \\ f_n \end{Bmatrix}$$

or

$$\mathbf{Ku} = \mathbf{f}$$

$\mathbf{K}$ is the system (or structure) stiffness matrix

$\mathbf{u}$ is the system displacement vector

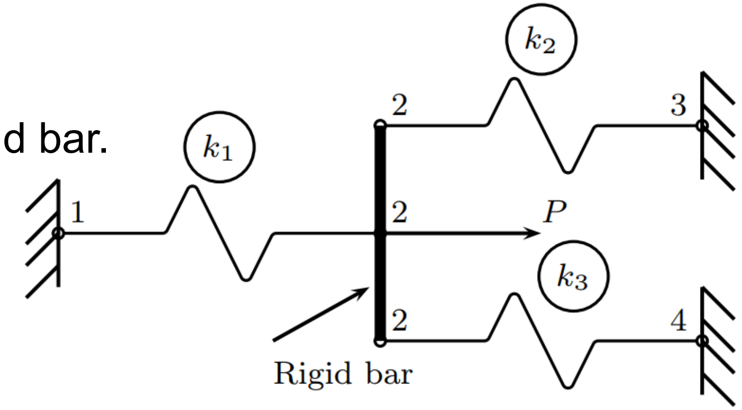
$\mathbf{f}$ is the system force vector

2. Discrete Systems – Basic Steps

Typical steps in any finite element problem:

- Define a set of elements connected at nodes.
- For each element, compute stiffness matrix $\mathbf{K}^{(e)}$, and force vector $\mathbf{f}^{(e)}$.
- Assemble the contribution of all elements into the global system $\mathbf{Ku} = \mathbf{f}$.
- Modify the global system by imposing essential (displacements) boundary conditions.
- Solve the global system and obtain the global displacements $\mathbf{u}$.
- For each element, evaluate the strains and stresses (post-processing).

3. Example 1 – A Spring System



- 03 spring finite elements are connected via a rigid bar.

$$u_1 = u_3 = u_4 = 0; \qquad k_i = 1; \qquad P = 10$$

- Local equilibrium equation for Spring 1, 2, 3:

$$\begin{Bmatrix} R_1^{(1)} \\ R_2^{(1)} \end{Bmatrix} = k^{(1)} \begin{bmatrix} 1 & -1 \\ -1 & 1 \end{bmatrix} \begin{Bmatrix} u_1^{(1)} \\ u_2^{(1)} \end{Bmatrix}; \quad \begin{Bmatrix} R_1^{(2)} \\ R_2^{(2)} \end{Bmatrix} = k^{(2)} \begin{bmatrix} 1 & -1 \\ -1 & 1 \end{bmatrix} \begin{Bmatrix} u_1^{(2)} \\ u_2^{(2)} \end{Bmatrix}; \quad \begin{Bmatrix} R_1^{(3)} \\ R_2^{(3)} \end{Bmatrix} = k^{(3)} \begin{bmatrix} 1 & -1 \\ -1 & 1 \end{bmatrix} \begin{Bmatrix} u_1^{(3)} \\ u_2^{(3)} \end{Bmatrix}$$

- Compatibility condition to relate local (element) and global (structure) displacements

$$u_1^{(1)} = u_1, \quad u_2^{(1)} = u_1^{(2)} = u_1^{(3)} = u_2, \quad u_2^{(2)} = u_3, \quad u_2^{(3)} = u_4$$

- Contribution of each element stiffness to the global stiffness

$$\mathbf{K}^{(1)} = \begin{array}{c} \begin{array}{cccc} \mathbf{u_1} & \mathbf{u_2} & \mathbf{u_3} & \mathbf{u_4} \end{array} \\ \begin{bmatrix} k_1 & -k_1 & 0 & 0 \\ -k_1 & k_1 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \end{bmatrix} \end{array} \begin{array}{c} \mathbf{u_1} \\ \mathbf{u_2} \\ \mathbf{u_3} \\ \mathbf{u_4} \end{array} \quad \mathbf{K}^{(2)} = \begin{array}{c} \begin{array}{cccc} \mathbf{u_1} & \mathbf{u_2} & \mathbf{u_3} & \mathbf{u_4} \end{array} \\ \begin{bmatrix} 0 & 0 & 0 & 0 \\ 0 & k_2 & -k_2 & 0 \\ 0 & -k_2 & k_2 & 0 \\ 0 & 0 & 0 & 0 \end{bmatrix} \end{array} \begin{array}{c} \mathbf{u_1} \\ \mathbf{u_2} \\ \mathbf{u_3} \\ \mathbf{u_4} \end{array} \quad \mathbf{K}^{(3)} = \begin{array}{c} \begin{array}{cccc} \mathbf{u_1} & \mathbf{u_2} & \mathbf{u_3} & \mathbf{u_4} \end{array} \\ \begin{bmatrix} 0 & 0 & 0 & 0 \\ 0 & k_3 & 0 & -k_3 \\ 0 & 0 & 0 & 0 \\ 0 & -k_3 & 0 & k_3 \end{bmatrix} \end{array} \begin{array}{c} \mathbf{u_1} \\ \mathbf{u_2} \\ \mathbf{u_3} \\ \mathbf{u_4} \end{array}$$

3. Example 1 – A Spring System

- Equilibrium of force at node 1: $\sum_{e=1}^{3} R^{(e)} = F_1 \quad \leftrightarrow \quad R_1^{(1)} = F_1$
- Equilibrium of force at node 2: $\sum_{e=1}^{3} R^{(e)} = P \quad \leftrightarrow \quad R_2^{(1)} + R_1^{(2)} + R_1^{(3)} = P$
- Equilibrium of force at node 3: $\sum_{e=1}^{3} R^{(e)} = F_3 \quad \leftrightarrow \quad R_2^{(3)} = F_3$
- Equilibrium of force at node 4: $\sum_{e=1}^{3} R^{(e)} = F_4 \quad \leftrightarrow \quad R_2^{(4)} = F_4$
- The static global equilibrium equations

$$\begin{bmatrix} k_1 & -k_1 & 0 & 0 \\ -k_1 & k_1+k_2+k_3 & -k_2 & -k_3 \\ 0 & -k_2 & k_2 & 0 \\ 0 & -k_3 & 0 & k_3 \end{bmatrix} \begin{Bmatrix} u_1 \\ u_2 \\ u_3 \\ u_4 \end{Bmatrix} = \begin{Bmatrix} F_1 \\ P \\ F_3 \\ F_4 \end{Bmatrix}$$

- Apply the boundary conditions, we have

$$\begin{bmatrix} k_1 & -k_1 & 0 & 0 \\ -k_1 & k_1+k_2+k_3 & -k_2 & -k_3 \\ 0 & -k_2 & k_2 & 0 \\ 0 & -k_3 & 0 & k_3 \end{bmatrix} \begin{Bmatrix} 0 \\ u_2 \\ 0 \\ 0 \end{Bmatrix} = \begin{Bmatrix} F_1 \\ P \\ F_3 \\ F_4 \end{Bmatrix}$$

- Unknown displacement is solved: $(k_1+k_2+k_3)u_2 = P \quad \rightarrow \quad u_2 = P/(k_1+k_2+k_3)$
- Unknown reactions are found: $-k_1u_2 = F_1; \quad -k_2u_2 = F_3; \quad -k_3u_2 = F_4$

3. Example 1 – Python Code

❖ "problem_1_spring.py" for the example "A Spring System" is available at:

https://github.com/mctrinh/fem-class.git

Content

1. Continuum Mechanics
    - Gradient, Divergence, Curl Theorems
    - Kinematics
    - Equilibrium Equations, Equations of Elasticity
2. Discrete Systems
    - Springs and Bars
    - Equilibrium at Node
    - Basic Steps
3. Example 1
    - A Spring System
    - Python Code

thank you