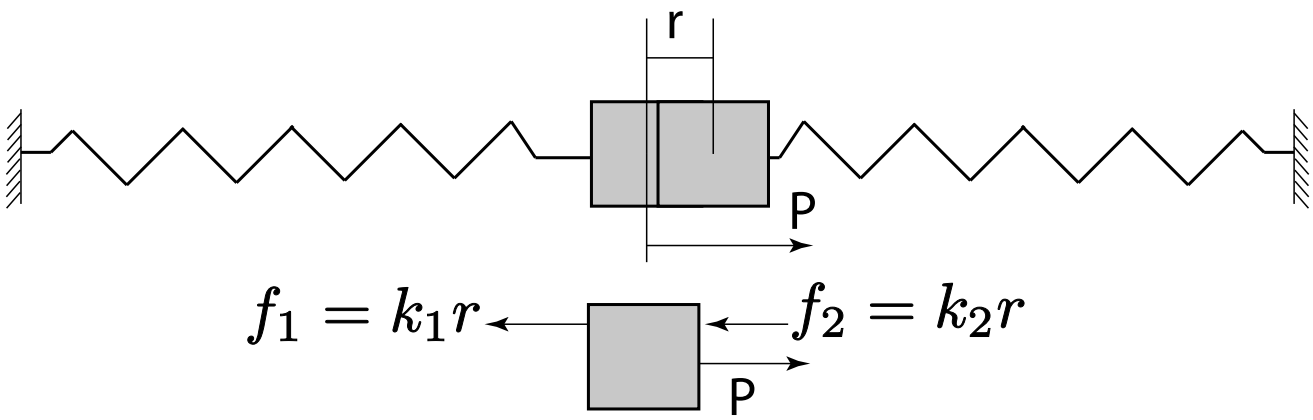
Balance laws

Motivation

- Why start with a balance law?
  - They are the actual physics laws.
  - They contain more *information* than their corresponding PDEs.
  - Larger solution space than the PDEs.
- Can we directly start the FE formulation from a PDE?
  - Yes, FE formulation starts from a differential equation.
  - A PDE may not be derived from a balance law.

Balance law for discrete systems (Statics)

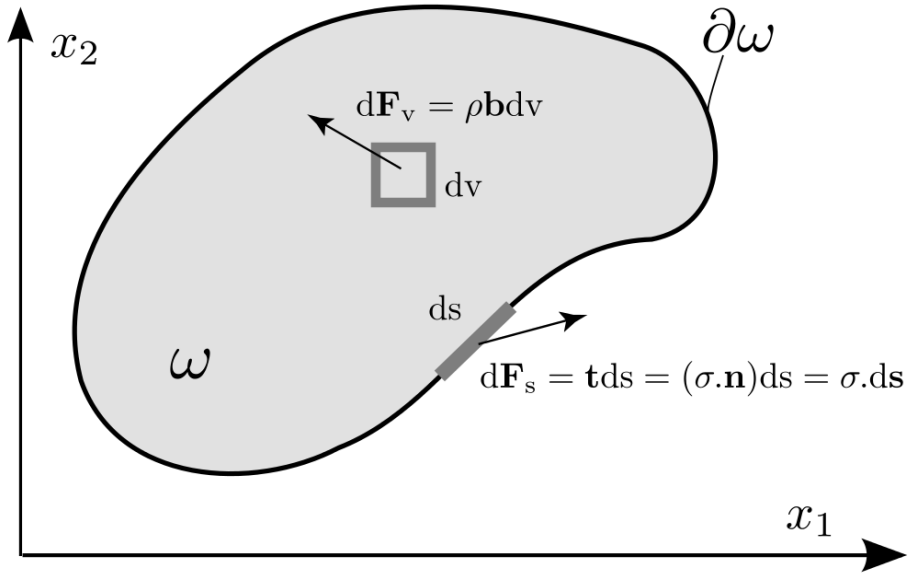


Newton's law:

$$F = \Sigma f_i = -f_1 - f_2 + P = 0 \quad \Rightarrow \quad r = \frac{P}{k_1 + k_2} \tag{1}$$

The unknowns of discrete point(s) can be obtained from the solution of one or a system of equations.

Balance law for continuum(Statics for solids)



$$\mathbf{F} = 0 \quad \Rightarrow \tag{2}$$

$$\mathbf{F} = \mathbf{F}_s + \mathbf{F}_v = \int_{\partial\omega} \mathrm{d}\mathbf{F}_\mathrm{s} + \int_{\omega} \mathrm{d}\mathbf{F}_\mathrm{v} = 0, \quad \text{where}$$

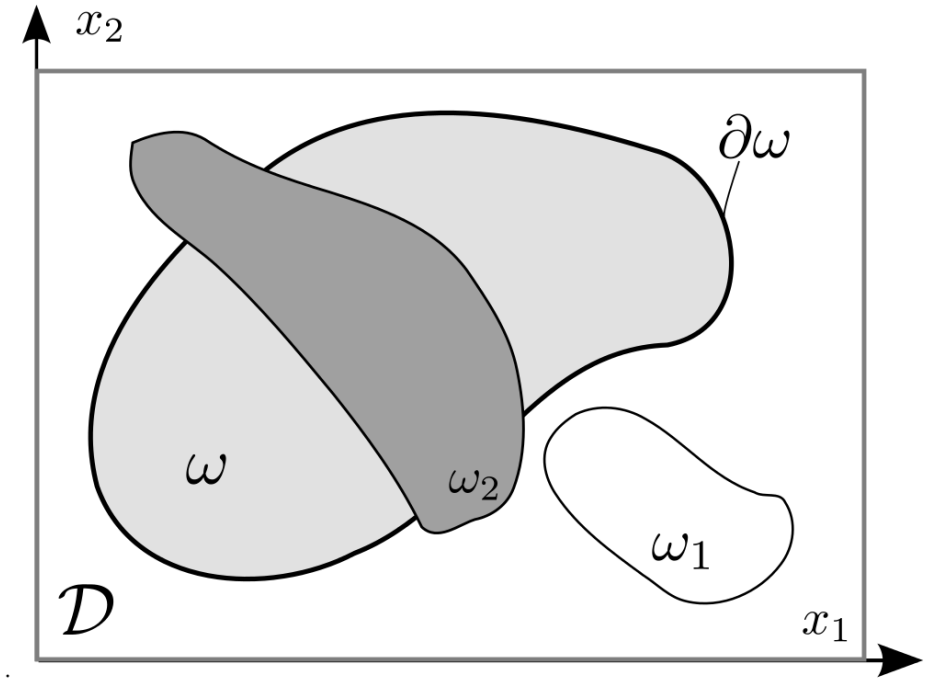
$\mathbf{F}_s =$ Sum of forces from tractions $\mathbf{t}$ (boundary flux) on the boundary of ω

$\mathbf{F}_v =$ Sum of forces from body force $\mathbf{b}$ (source term) inside ω

$\mathrm{d}\mathbf{F}_\mathrm{s} = \mathbf{t}\mathrm{ds} = (\sigma.\mathbf{n})\mathrm{ds} = \sigma.\mathrm{ds}$

$\mathrm{d}\mathbf{F}_\mathrm{v} = \rho\mathbf{b}\mathrm{dv}$ $\qquad$ $\partial\omega =$ boundary of ω

Balance law for continuum(Statics for solids)



The balance law is valid for arbitrary domains ω:

$$\forall \omega \subset \mathcal{D} : \int_{\partial\omega} \mathbf{t}\, \mathrm{ds} + \int_{\omega} \rho \mathbf{b}\, \mathrm{dv} = 0 \tag{3}$$

$$\text{that is} \quad \int_{\partial\omega} \sigma.\mathrm{d}\mathbf{s} + \int_{\omega} \rho \mathbf{b}\, \mathrm{dv} = 0$$

Note that $\mathrm{d}\mathbf{s} = \mathbf{n}\mathrm{ds}$ (boldface is used for tensor quantities)

Balance law for discrete systems (Dynamics)

$$\left.\begin{aligned} \mathbf{F} &= \frac{\mathrm{d}\mathbf{P}}{\mathrm{d}t} \\ \mathbf{P} &= m\mathbf{v} = \text{linear momentum} \end{aligned}\right\} \quad \Rightarrow \quad \mathbf{F} = m\frac{\mathrm{d}\mathbf{v}}{\mathrm{d}t} = ma \tag{4}$$

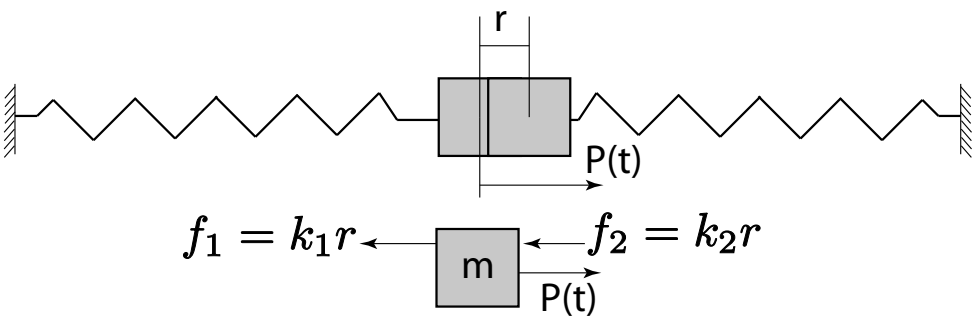
Another representation is using the "Impulse" J for the force $\mathbf{F}$ over the time interval Δt:

$$J = \int_{\Delta t} \mathbf{F}\mathrm{d}t \tag{5}$$

then Newton's second law reads as

$$J = \Delta\mathbf{P} \tag{6}$$

Advantage: $\mathbf{P}$ does not need to be differentiable

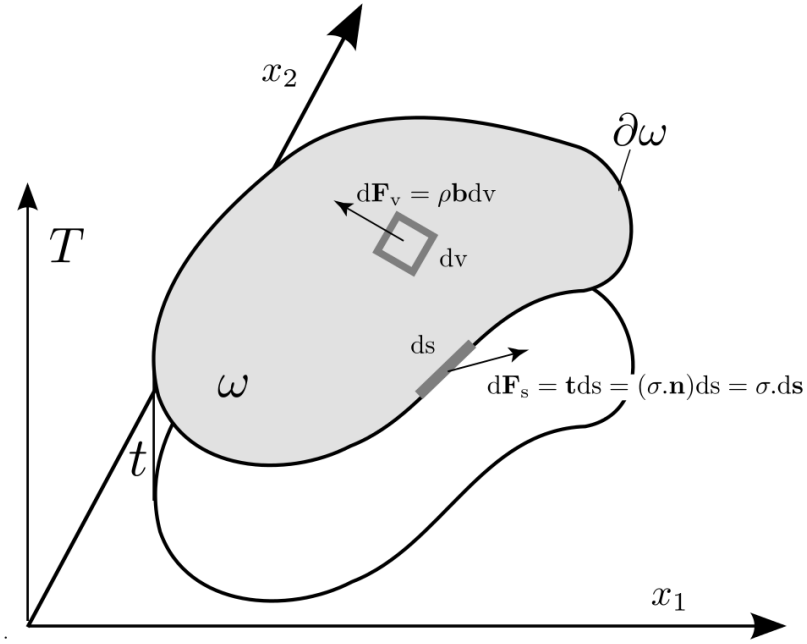


$$\mathbf{F} = P(t) - f_1 - f_2 = m\mathbf{a} \quad \Rightarrow \tag{7}$$

$$m\ddot{r} + (k_1 + k_2)r = P(t) \tag{8}$$

can solve the DE with FEM

Balance law for continuum(Dynamics for solids)



First approach (similar to $\mathbf{F} = \frac{\mathrm{d}\mathbf{P}}{\mathrm{d}t}$ for discrete systems):

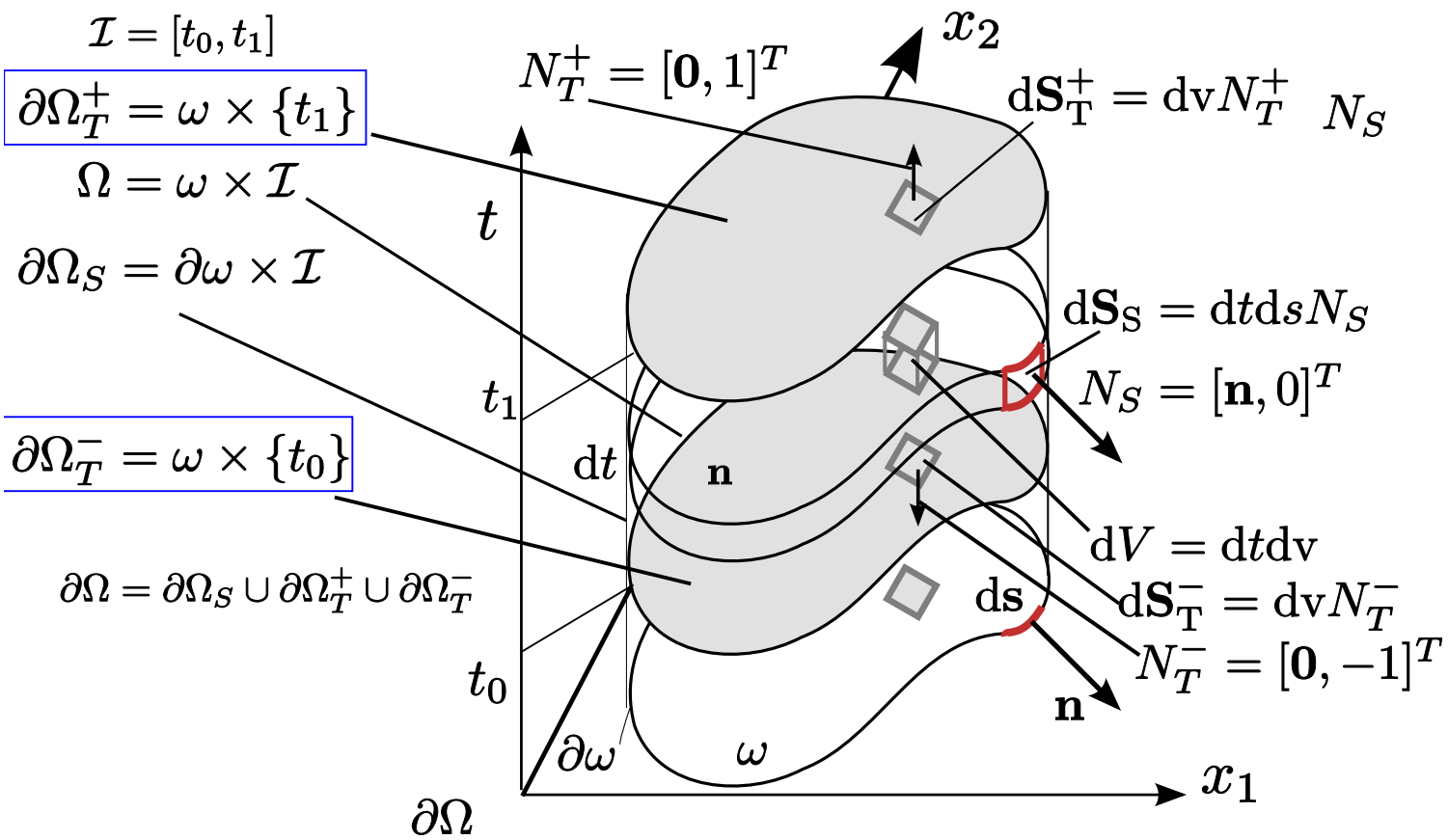
$$\forall \omega \subset \mathcal{D} : \mathbf{F} = \frac{\mathrm{d}\mathbf{P}}{\mathrm{d}t} \quad \Rightarrow$$

$$\int_{\partial\omega} \sigma.\mathrm{ds} + \int_{\omega} \rho\mathbf{b}\,\mathrm{dv} = \frac{\mathrm{d}}{\mathrm{d}t}\int_{\omega} \mathbf{p}\,\mathrm{dv} \tag{9}$$

note that $\mathbf{P} = \int_\omega \mathbf{p}\,\mathrm{dv}$ where $\mathbf{p}$ is the *linear momentum density* defined by:

$$\mathbf{p} = \frac{\mathrm{d}\mathbf{P}}{\mathrm{dV}} = \frac{(\mathrm{d}m)\mathbf{v}}{\mathrm{d}V} = \frac{(\rho\mathrm{d}V)\mathbf{v}}{\mathrm{d}V} = \rho\mathbf{v} \tag{10}$$

Balance law for continuum(Dynamics for solids)

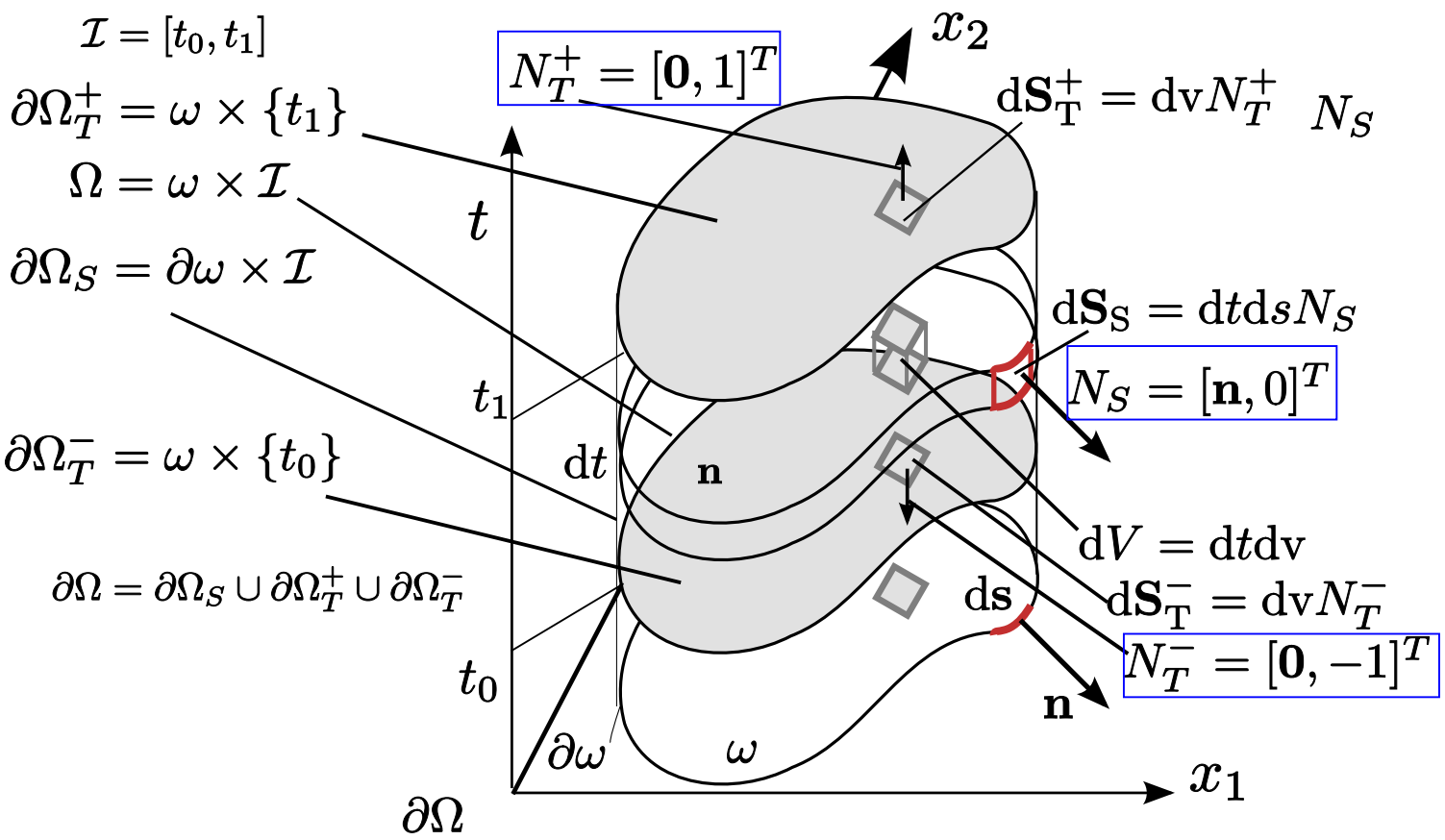


Second approach (similar to $J = \int_{\Delta t} \mathbf{F}\,\mathrm{d}t = \Delta\mathbf{P}$ for discrete systems):

$$\forall \Omega = \omega \times \mathcal{I} \subset \mathcal{D} : J = \int_{t_0}^{t_1} \mathbf{F} = \mathbf{P}(t_1) - \mathbf{P}(t_0) \quad \Rightarrow$$

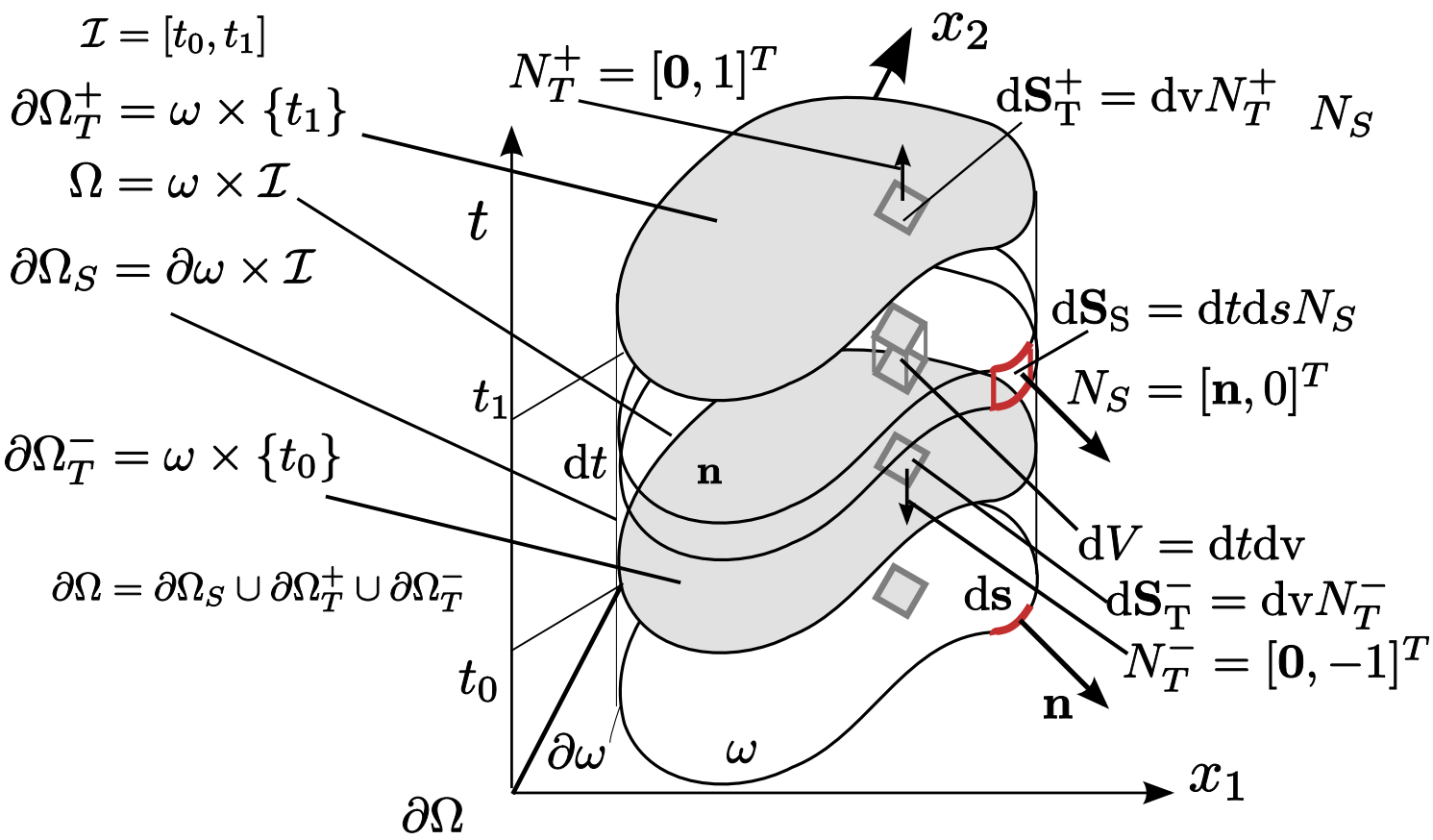
$$\int_{t_0}^{t_1} \left(\int_{\partial\omega} \sigma.\mathrm{d}\mathbf{s} \right) \mathrm{d}t + \int_{t_0}^{t_1} \left(\int_{\omega} \rho\mathbf{b}\,\mathrm{d}v \right) \mathrm{d}t = \int_{\partial\Omega_T^+} \mathbf{p}\,\mathrm{d}v - \int_{\partial\Omega_T^-} \mathbf{p}\,\mathrm{d}v$$

second approach continued . . .



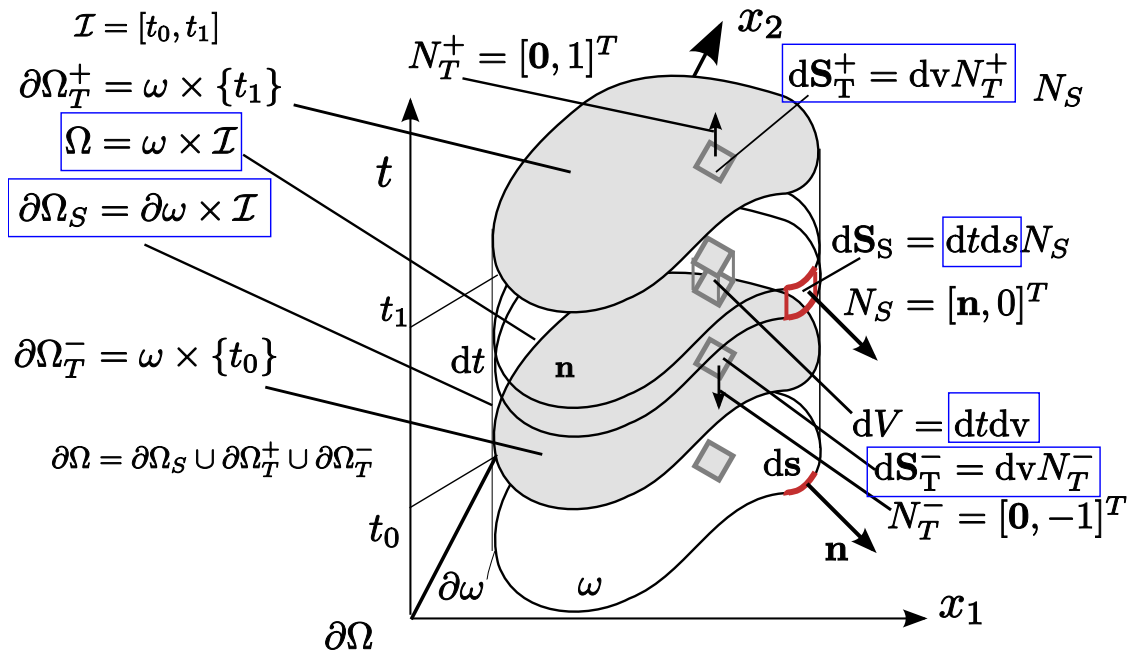
$$\int_{\partial\Omega_S} \sigma . (\mathrm{d}\mathbf{s}\mathrm{d}t) + \int_{\Omega} \rho\mathbf{b} \ (\mathrm{dv}\mathrm{d}t) = \int_{\partial\Omega_T^+} \mathbf{p} \, \mathrm{dS}_T^+ - \int_{\partial\Omega_T^-} \mathbf{p} \, \mathrm{dS}_T^-$$

second approach continued . . .



$$\int_{\partial\Omega_S} (\sigma.\mathbf{n}) \underbrace{\mathrm{dS}_S}_{\mathrm{ds}\mathrm{d}t} + \int_\Omega \rho\mathbf{b} \underbrace{\mathrm{dV}}_{\mathrm{dv}\mathrm{d}t} = \int_{\partial\Omega_T^+} \mathbf{p}\, \mathrm{dS}_T^+ + \int_{\partial\Omega_T^-} \mathbf{p}\, (-\mathrm{dS}_T^-)$$

second approach continued . . .



$$\int_{\partial\Omega_S} \underbrace{-\left[\ -\sigma \ \middle|\ \mathbf{p}\ \right] . \overbrace{\begin{bmatrix}\mathbf{n}\\ 0\end{bmatrix}}^{\mathbf{N}_S}}_{\sigma.\mathbf{n}} \mathrm{dS}_S + \int_\Omega \rho\mathbf{b}\,\mathrm{dV} =$$

$$\int_{\partial\Omega_T^+} \underbrace{\left[\ -\sigma \ \middle|\ \mathbf{p}\ \right] . \overbrace{\begin{bmatrix}\mathbf{0}\\ 1\end{bmatrix}}^{\mathbf{N}_T^+}}_{\mathbf{p}} \mathrm{dS}_T^+ + \int_{\partial\Omega_T^-} \underbrace{\left[\ -\sigma \ \middle|\ \mathbf{p}\ \right] . \overbrace{\begin{bmatrix}\mathbf{0}\\ -1\end{bmatrix}}^{\mathbf{N}_T^-}}_{-\mathbf{p}} \mathrm{dS}_T^-$$

second approach continued . . .

Let us define the spacetime flux $\mathbf{M}$ by:

$$\mathbf{M} = \left[\ -\sigma \ \middle|\ \mathbf{p}\ \right] \tag{11a}$$

$$\mathbf{p} = \text{linear momentum} = \text{temporal flux} = \text{conserved quantity} \tag{11b}$$

$$-\sigma = -\text{stress} = \text{total } \textit{outward} \text{ spatial flux} \tag{11c}$$

then

$$\left.\begin{aligned} \int_{\partial\Omega_T^+} \mathbf{M}.\mathbf{N}_T^+ \,\mathrm{dS}_T^+ + \int_{\partial\Omega_T^-} \mathbf{M}.\mathbf{N}_T^- \,\mathrm{dS}_T^- & \\ + \int_{\partial\Omega_S} \mathbf{M}.\mathbf{N}_S \,\mathrm{dS}_S \quad &= \int_\Omega \rho\mathbf{b} \,\mathrm{dV} \\ \partial\Omega_T^+ \cup \partial\Omega_T^- \cup \partial\Omega_S \quad &= \partial\Omega \quad \text{disjoint union} \end{aligned}\right\} \quad \Rightarrow$$

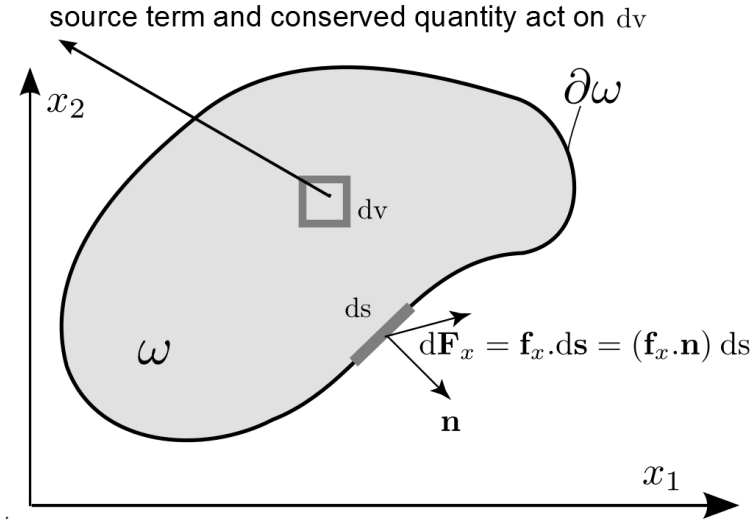
$$\boxed{\int_{\partial\Omega} \mathbf{M}.\mathrm{d}\mathbf{S} = \int_{\partial\Omega} \mathbf{M}.\mathbf{N} \,\mathrm{dS} = \int_\Omega \rho\mathbf{b} \,\mathrm{dV}} \tag{12}$$

General form of balance laws

For a general conservation law let:

- $\mathbf{f}_t$: conserved quantity = temporal flux
- $\mathbf{f}_x$: total outward spatial flux
- $\mathbf{r}$: source term

then the balance law for dynamics reads:



$$\forall \omega \subset \mathcal{D} \wedge \forall t : \int_\omega \mathbf{r}\,\mathrm{dv} - \int_{\partial\omega} \mathbf{f}_x.\mathrm{ds} = \int_\omega \mathbf{r}\,\mathrm{dv} - \int_{\partial\omega} (\mathbf{f}_x.\mathbf{n})\,\mathrm{ds} = \frac{\mathrm{d}}{\mathrm{d}t}\int_\omega \mathbf{f}_t\,\mathrm{dv} \tag{13}$$

For static case the RHS is zero (*i.e.*, the quantity $\int_\omega \mathbf{f}_t\,\mathrm{dv}$ remains constant). The static balance law reads:

$$\forall \omega \subset \mathcal{D} : \int_\omega \mathbf{r}\,\mathrm{dv} - \int_{\partial\omega} \mathbf{f}_x.\mathrm{ds} = \int_\omega \mathbf{r}\,\mathrm{dv} - \int_{\partial\omega} (\mathbf{f}_x.\mathbf{n})\,\mathrm{ds} = \mathbf{0} \tag{14}$$

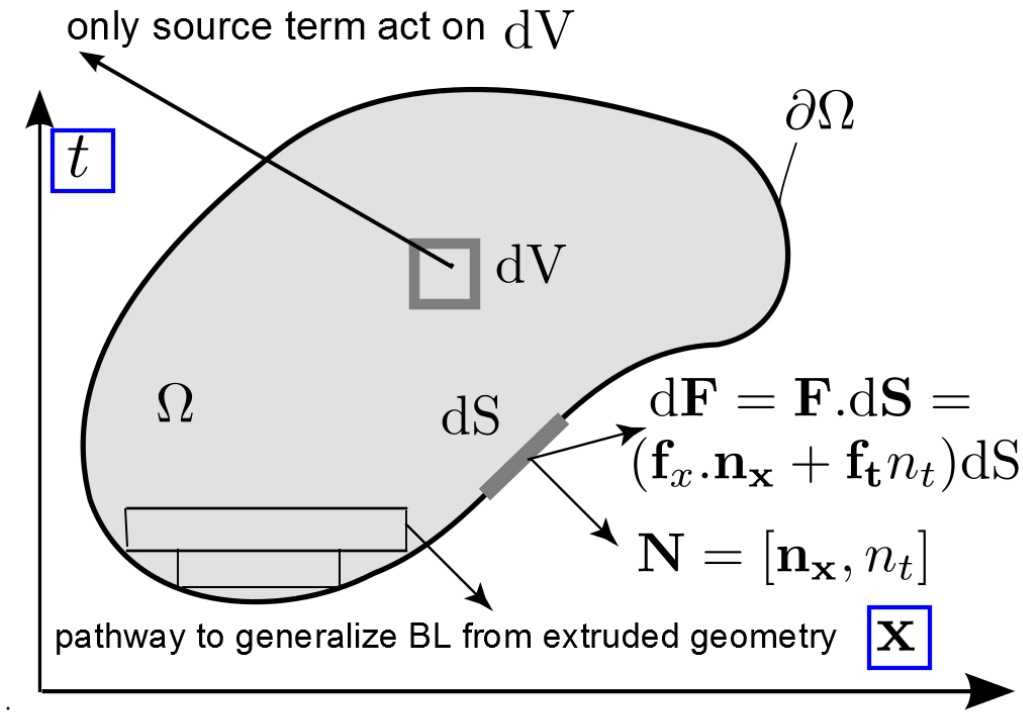
These can be directly compared to $\mathbf{F} = \mathrm{d}\mathbf{P}/\mathrm{d}t$ and $\mathbf{F} = 0$ in previous discrete examples.

General form of balance laws using spacetime flux

Using the same definitions from previous page we define the spacetime flux by

$$\mathbf{F} = [\mathbf{f}_x | \mathbf{f}_t] \tag{15}$$

then the balance law for dynamics reads:



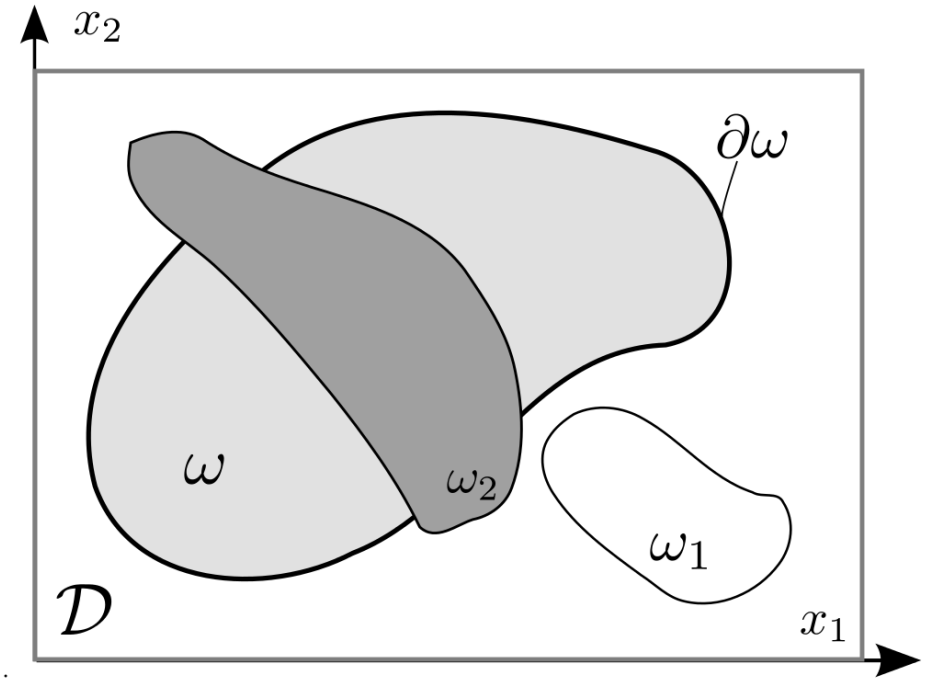
$$\boxed{\forall \Omega \subset \mathcal{D} : \int_{\partial\Omega} \mathbf{F}.\mathrm{dS} - \int_{\Omega} \mathbf{r}\,\mathrm{dV} = \int_{\partial\Omega} (\mathbf{f}_x.\mathbf{n_x} + \mathbf{f_t} n_t)\mathrm{dS} - \int_{\Omega} \mathbf{r}\,\mathrm{dV} = \mathbf{0}} \tag{16}$$

This can be directly compared to $\int_{\Delta t} \mathbf{F} = \Delta\mathbf{P}$ in previous discrete examples.

Strong Form
Boundary Value Problem

Path to obtain a point-wise equation (strong form)

While balance law hold for arbitrary volumes ω:



$$\forall \omega \subset \mathcal{D} : \int_{\partial\omega} \mathbf{F}.\mathrm{dS} - \int_{\omega} \mathbf{r}\,\mathrm{dV} = \mathbf{0}$$

the fact that they are always in integral form makes it practically difficult to obtain the solution to a problem. We systematically derive a point-wise equation (*i.e.*, a differential equation) which has a more limited solution space. We discuss the additional conditions that make the two approaches equivalent.

Transfer of boundary to interior integral in 1D

- First, we attempt to change the surface integral in the balance law to an interior integration.
- Most often, we do the opposite and transfer "volume integration" to "surface integration" (using the fundamental theorem of calculus):

$$\int_{[a,b]} f(x)\,\mathrm{d}x = F(b) - F(a), \text{ for } f \text{ continuous on } [a,b] \text{ and } \forall x \in [a,b]\ F'(x) = f(x)$$

- The reverse direction (2nd fundamental theorem) states that:

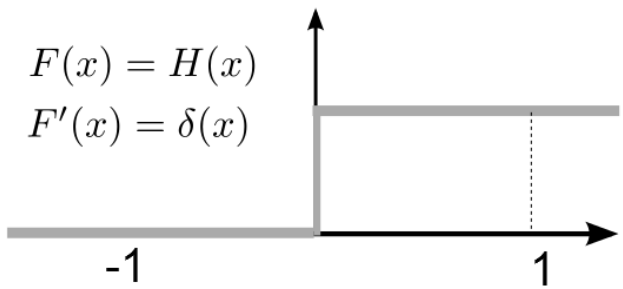
$$F(b) - F(a) = \int_{[a,b]} f(x)\,\mathrm{d}x, \quad \text{for} \tag{17}$$

1. For all $x \in [a,b]:\ F'(x) = f(x)$.
2. $f(x)$ is integrable.

- The divergence theorem is the generalization of this theorem.

Comment on condition $F'(x) = f(x)$ for all points

- $F(x)$ should be differentiable at all points. Consider the functions $F(x) = H(x)$ and $f(x) = 0$. $F'(x) = f(x)$ everywhere except at 0.



$$\left.\begin{aligned} \int_{-1}^{1} f(x) = \int_{-1}^{1} 0 \quad &= 0 \\ F(1) - F(-1) \quad &= 1 \end{aligned}\right\}$$

$$\Rightarrow \int_{-1}^{1} f(x) \neq F(1) - F(0)$$

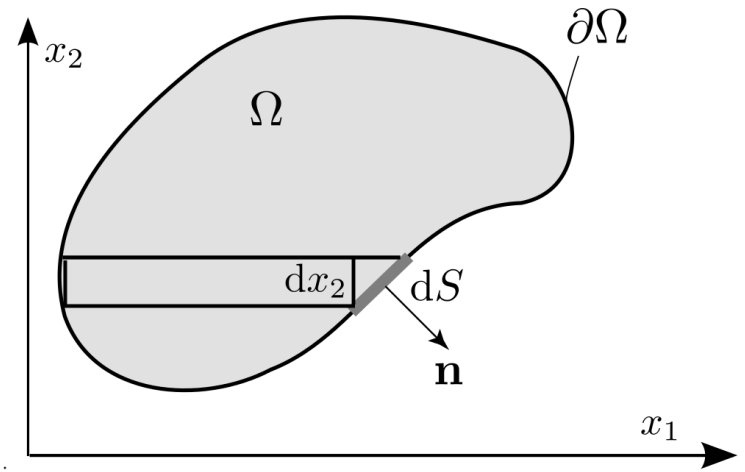
- The problem stems from the fact that F is not differentiable at zero.
- Loosely speaking $F'(x) = \delta(x)$, the so-called delta Dirac "function". Then,

$$\int_{-1}^{1} F'(x) = \int_{-1}^{1} \delta(x) = 1 = F(1) - F(0)$$

- Delta dirac is in fact not a function. It's a distribution (generalization of a function), which is infinitely differentiable.
- Physically, jump conditions similar to this example correspond to regions where the strong form would not hold.

Transfer of boundary to interior integral higher dimensions

- Ω is compact and closed.
- $\partial\Omega$ is piecewise smooth.
- Normal vector $\mathbf{n}$ is defined almost everywhere (a.e.) and is pointing outward.
- tensor field (scalar, vector, matrix, ...):
  - $\mathbf{F}_{,i} = \partial\mathbf{F}/\partial x_i$ exists everywhere and
  - is continuous.



$$\boxed{\int_{\partial\Omega} \mathbf{F}.n_i \,\mathrm{d}S = \int_{\Omega} F_{,i} \,\mathrm{d}V} \tag{18}$$

This is the generalization of the 1D version:

$$1.F(b) + (-1).F(a) = F(b) - F(a) = \int_{[a,b]} F'(x) \,\mathrm{d}x$$

Divergence Theorem

Let tensor field $\mathbf{F}$ have continuous partial derivatives, $F_{,i}$, everywhere for all i, Ω be closed, compacted, and has piecewise smooth boundary with well-defined normal vector a.e. Then,

$$\boxed{\int_{\partial\Omega} \mathbf{F}.\mathbf{n}\,\mathrm{d}S = \int_{\Omega} \mathrm{div}\mathbf{F}\,\mathrm{d}V} \tag{19}$$

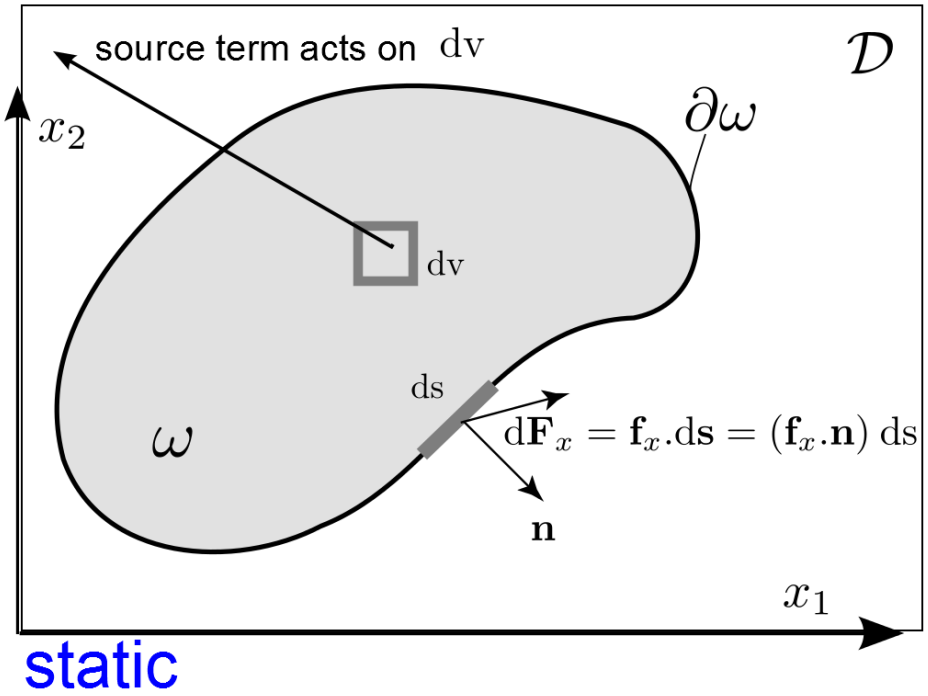
where $(\mathrm{div}\mathbf{F})_\alpha = (\nabla.\mathbf{F})_\alpha = \sum_{i=1}^{\mathrm{d}} F_{\alpha i,i}$, *e.g.*, for a vector $\mathbf{F}$, $\mathrm{div}\mathbf{F} = F_{i,i}$
Comments on having a continuous derivative for $\mathbf{F}$:

- This condition can be relaxed, *i.e.*, $\mathbf{F}$ still having partial derivatives everywhere, but not necessarily continuous.
- For all practical applications and function spaces for our approximation solutions, if $F_{,i}$ exists it is also continuous.
- In general, existence of $F_{,i}$ does not ensure that it is also continuous:

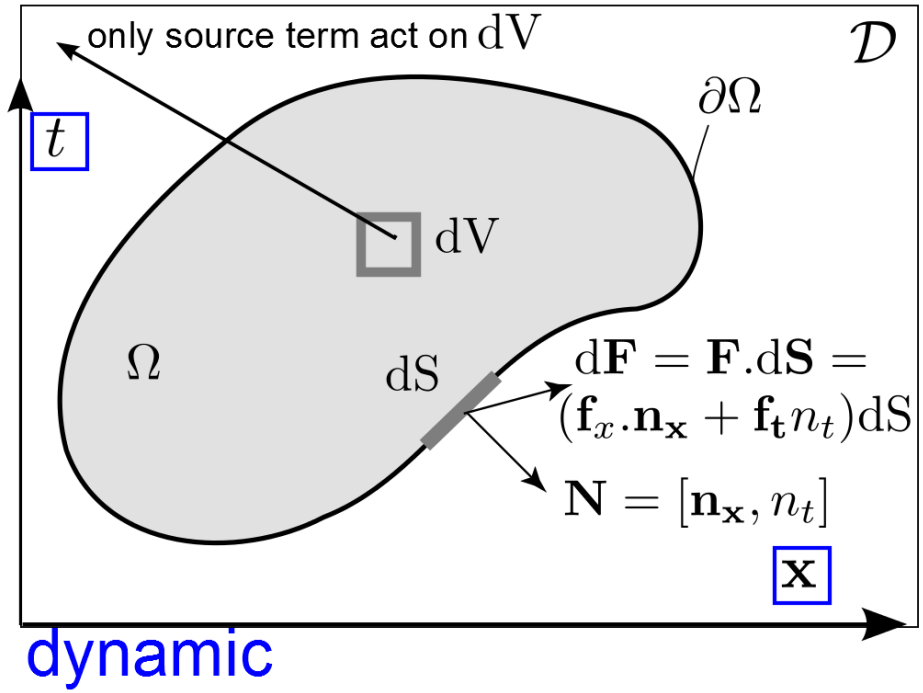
$$F(x) = \begin{cases} x^2 \sin\frac{1}{x} & x \neq 0 \\ 0 & x = 0 \end{cases}$$

$F'(x)$ exists everywhere but is not continuous at 0 (proof).

General form of balance laws



static



dynamic

$$\forall\omega \subset \mathcal{D} : \int_{\partial\omega} (\mathbf{f}_x.\mathbf{n})\,\mathrm{ds} - \int_{\omega} \mathbf{r}\,\mathrm{dv} = \mathbf{0} \qquad \forall\Omega \subset \mathcal{D} : \int_{\partial\Omega} (\mathbf{F}.\mathbf{N})\mathrm{dS} - \int_{\Omega} \mathbf{r}\,\mathrm{dV} = \mathbf{0}$$

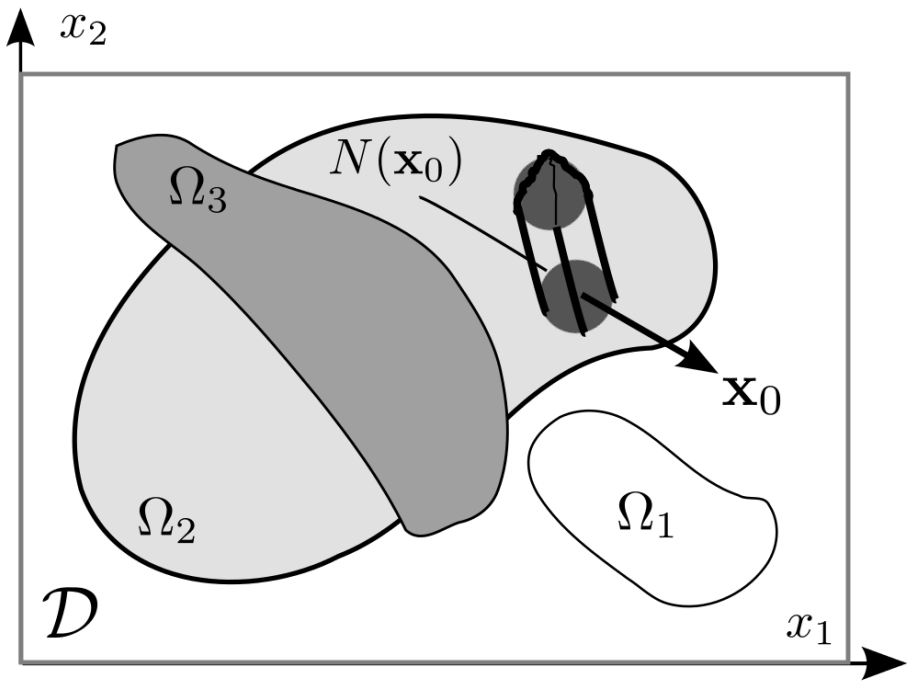
where $\mathbf{F} = [\mathbf{f}_x|\mathbf{f}_t]$ (*cf.* (15) and (16)). So, in either case a balance law takes the general form of:

$$\boxed{\forall\Omega \subset \mathcal{D} : \int_{\partial\Omega} (\mathbf{f}.\mathbf{n})\mathrm{ds} - \int_{\Omega} \mathbf{r}\,\mathrm{dv} = \mathbf{0}} \tag{20}$$

Localization theorem

Localization theorem states that if the integral of a continuus function is zero for all subsets of $\mathcal{D}$, then the function is zero:

$$\boxed{\forall \Omega \subset \mathcal{D} : \int_{\Omega} \mathbf{g}(\mathbf{x})\,\mathrm{dv} = \mathbf{0} \quad \Rightarrow \quad \forall x \in \mathcal{D} : \; g(\mathbf{x}) = 0} \tag{21}$$



Let's assume $g(x_0) \neq 0$ (*e.g.*, $g(x_0) > 0$). Since $g(\mathbf{x})$ is continuus, there is a neighborhood of $\mathbf{x}_0$ ($N(\mathbf{x}_0)$) that $g(x) > 0$. We choose an Ω that is only nonzero inside $N(\mathbf{x}_0)$. Then, $\int_{\Omega} g(\mathbf{x})\,\mathrm{d}V > 0$. Thus, $g(\mathbf{x}_0)$ cannot be nonzero and the function g is identically zero.

Strong form

We apply the divergence theorem (19) on the balance law from (20) to obtain:

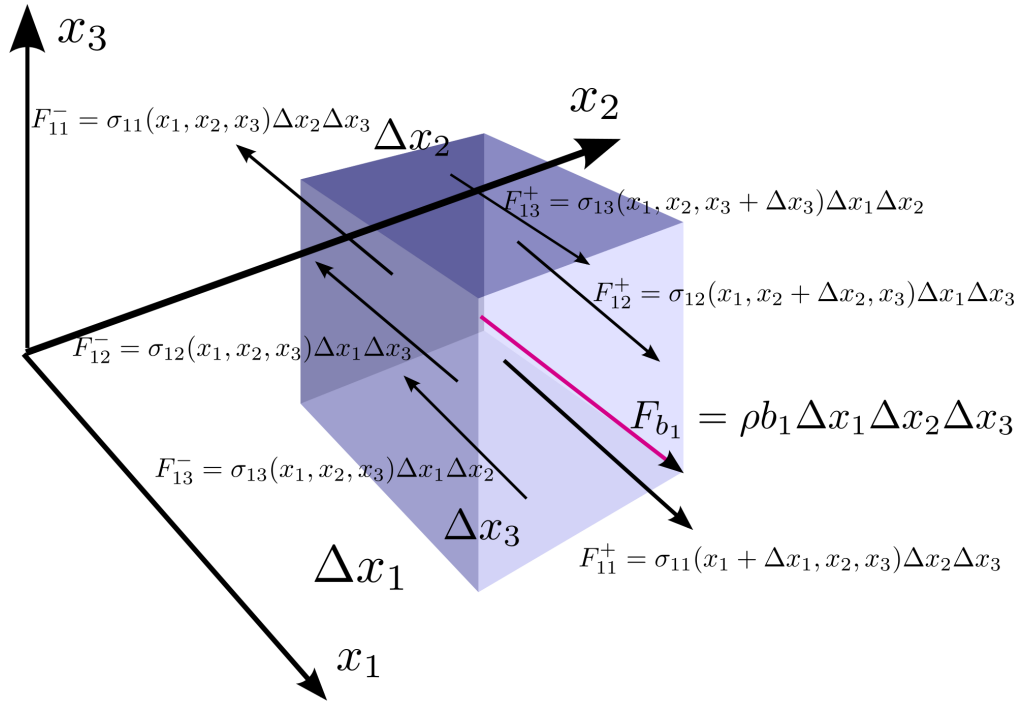
$$\forall \Omega \subset \mathcal{D} : \int_{\Omega} (\nabla . \mathbf{f} - \mathbf{r}) \, \mathrm{dv} = \mathbf{0} \tag{22}$$

Assuming the continuity of the source term, the localization theorem (21) implies:

$$\boxed{\forall \mathbf{x} \in \mathcal{D} : \nabla . \mathbf{f} - \mathbf{r} = \mathbf{0} \quad \text{Strong Form}} \tag{23}$$

Second approach to derive the strong form

Since the integration domain in (20) is arbitrary, we can choose a cube small enough about a point $\mathbf{x}$ and use the existence and continuity of the derivative to directly obtain the strong form. This is basically the approach generally used in undergraduate courses to derive differential equations for bars, beams, and so forth. This is just a formal proof (for solid mechanics only x_1 dir.), as the actual proof needs the continuity of partial derivative.



$$\sum F_{x_1} = (F_{11}^+ - F_{11}^-) + (F_{12}^+ - F_{12}^-) + (F_{13}^+ - F_{13}^-) + F_{b_1} = \mathrm{dm}\,\dot{p}_1 \Rightarrow$$

$$\Delta x_1 \Delta x_2 \Delta x_3 \left\{ \frac{\sigma_{11}(x_1 + \Delta x_1, x_2, x_3) - \sigma_{11}(x_1, x_2, x_3)}{\Delta x_1} + \right.$$

$$\frac{\sigma_{12}(x_1, x_2 + \Delta x_2, x_3) - \sigma_{12}(x_1, x_2, x_3)}{\Delta x_2} +$$

$$\left. \frac{\sigma_{13}(x_1, x_2, x_3 + \Delta x_3) - \sigma_{13}(x_1, x_2, x_3)}{\Delta x_3} + \rho b_1 - \rho \dot{p}_1 \right\} = 0 \Rightarrow$$

$$(\nabla . \sigma)_1 + \rho b_1 = \rho \dot{p}_1$$

Closing the system of equations (Statics)

Strong form (23) of balance of linear momentum for statics is:

$$\nabla.(-\sigma) - \rho\mathbf{b} = \mathbf{0}, \quad \Rightarrow \nabla.\sigma + \rho\mathbf{b} = \mathbf{0} \quad \Rightarrow \quad \sigma_{ij,j} + \rho b_i = 0 \tag{24}$$

where $\mathbf{f} = -\sigma$, $\mathbf{r} = \rho\mathbf{b}$, and $\nabla(.) = \frac{\partial(.)}{\partial x_1} + \frac{\partial(.)}{\partial x_2} + \frac{\partial(.)}{\partial x_3}$.

Type	Equation	n_{e}	new unknowns	n_{u}	$N_{\mathrm{e}} - N_{\mathrm{u}}$
Balance law	$\sigma_{ij,j} + \rho b_i = 0$	3	$\sigma_{ij} = \sigma_{ji}$, $i,j \in \{1,2,3\}$	6	3
Constitutive equation	$\sigma_{ij} = C_{ijkl}E_{kl}$	6	$E_{kl} = E_{lk}$	6	3
kinematic compatibility	$E_{kl} = \frac{1}{2}(u_{k,l} + u_{l,k})$	6	u_k	3	0

n_{e} = number of new equations n_{u} = number of new unknowns
N_{e} = total number of equations N_{u} = total number of unknowns

- We need other equations (constitutive equations and kinematic compatibility equations) to balance the number of unknowns and equations.

Closing the system of equations (Dynamics)

Strong form (23) of balance of linear momentum for dynamics is:

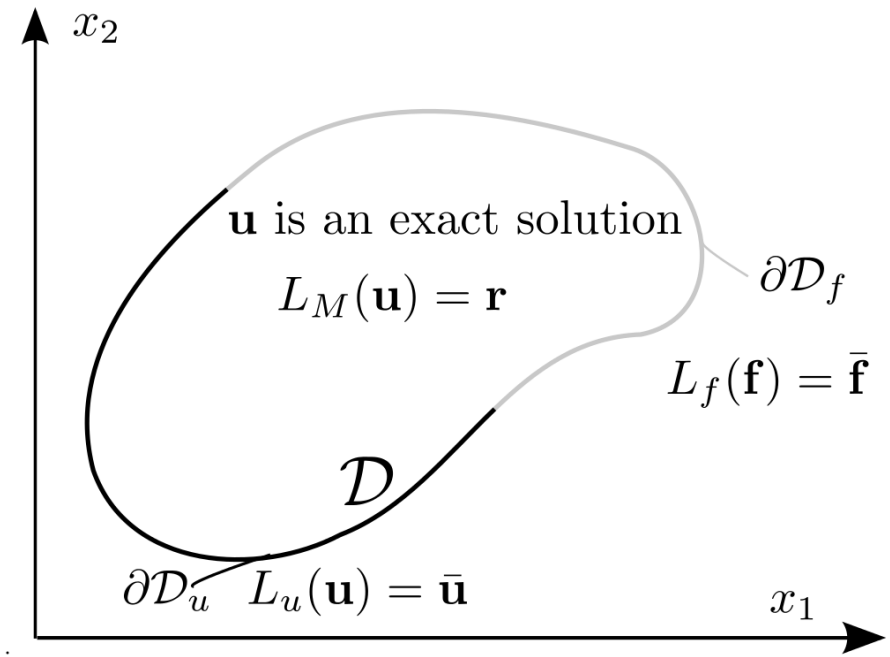
$$\nabla.[-\sigma|\mathbf{p}] - \rho\mathbf{b} = \mathbf{0}, \quad \Rightarrow \nabla.\sigma - \frac{\partial \mathbf{p}}{\partial t} + \rho\mathbf{b} = \mathbf{0} \quad \Rightarrow \quad \sigma_{ij,j} + \rho b_i = \dot{p}_i \qquad (25)$$

where $\mathbf{f} = [-\sigma|\mathbf{p}]$, $\mathbf{r} = \rho\mathbf{b}$, and $\nabla(.) = [\nabla_x(.)|\nabla_t(.)] = (\frac{\partial(.)}{\partial x_1} + \frac{\partial(.)}{\partial x_2} + \frac{\partial(.)}{\partial x_3}) + \frac{\partial(.)}{\partial t}$.

Type	Equation	n_e	new unknowns	n_u	$N_e - N_u$
Balance law	$\sigma_{ij,j} + \rho b_i = \dot{\mathbf{p}}$	3	$\sigma_{ij} = \sigma_{ji}$, $i,j \in \{1,2,3\}$	6	6
			p_j	3	
Constitutive equation	$\sigma_{ij} = C_{ijkl}E_{kl}$	6	$E_{kl} = E_{lk}$	6	6
	$p_j = \rho v_j$	3	v_j	3	
kinematic compatibility	$E_{kl} = \frac{1}{2}(u_{k,l} + u_{l,k})$	6	u_k	3	0
	$v_j = \dot{u}_j$	3			

Different types of spatial boundary conditions (BC)

- $L_M(\mathbf{u}) = \mathbf{r}$ is the strong form after incorporating the "constitutive" and "compatibility" conditions.
- $L_u(\mathbf{u}) = \bar{\mathbf{u}}$: Dirichlet BC, order M_u.
- $L_f(\mathbf{u}) = \bar{\mathbf{f}}$: Neumann BC, order M_f.
- $\mathbf{u}$ is a primary field, (*e.g.*, displacement for solid mechanics; temperature for heat conduction)
- M is typically even (*e.g.*, $M = 2m$)



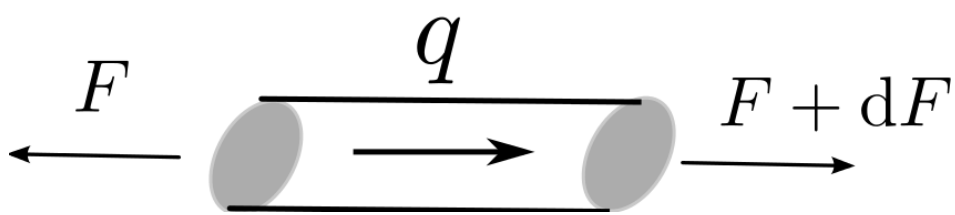
$\partial\mathcal{D}_u$	$\partial\mathcal{D}_f$
Dirichlet BC	Neumann BC
Essential BC (typically strongly enforced)	Natural BC ("naturally" derived from balance law fluxes)
"primary" or "kinematic" BC	"flux" or "force" BC

Sample Boundary value problems: 1D static bar

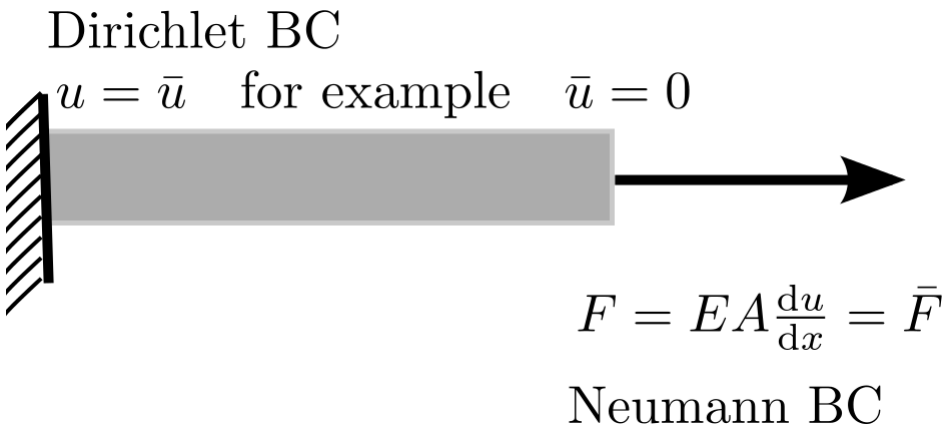
$$\left.\begin{aligned} \nabla.\mathbf{F} - \mathbf{r} &= 0 && \text{Balance law} \\ \mathbf{F} &= F && \text{Flux} \\ \mathbf{r} &= -q && \text{Source term} \\ F &= A\sigma = AE\epsilon && \text{Constitutive equation} \\ \epsilon &= \frac{\mathrm{d}u}{\mathrm{d}x} && \text{Kinematic compatibility} \end{aligned}\right\} \Rightarrow \boxed{\frac{\mathrm{d}}{\mathrm{d}x}\left(EA\frac{\mathrm{d}u}{\mathrm{d}x}\right) = -q} \tag{26}$$

where

F = Axial Force
q = Axial body force
σ = Stress
ϵ = Strain
E = Elastic modulus
u = Displacement



1D static bar: BCs



Operator	Sample	1D bar	operator order
$L_{2m}(\mathbf{u}) = \mathbf{r}$	$\frac{\mathrm{d}}{\mathrm{d}x}\left(EA\frac{\mathrm{d}u}{\mathrm{d}x}\right) = -q$	$L_{2m} = \frac{\mathrm{d}}{\mathrm{d}x}\left(EA\frac{\mathrm{d}(.)}{\mathrm{d}x}\right)$	$m = 1 (M = 2)$
$L_u(\mathbf{u}) = \bar{\mathbf{u}}$	$u = \bar{u}$	$L_u = (.)$	$M_u = 0$
$L_f(\mathbf{u}) = \bar{\mathbf{f}}$	$EA\frac{\mathrm{d}u}{\mathrm{d}x} = \bar{f}$	$L_f = EA\frac{\mathrm{d}(.)}{\mathrm{d}x}$	$M_f = 1$

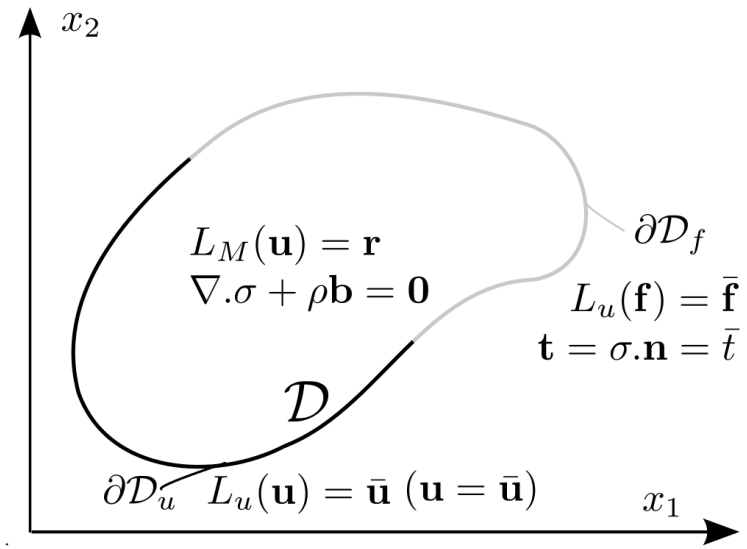
- Can Dirichlet and Neumann BCs be enforced simultaneously at the same point?
- Do Neumann boundary conditions correspond to the flux terms?
- How many boundary conditions do we have at each end point? How is it related to m?
- Verify $M_u + M_f = M - 1$.

Sample Boundary value problems: Elastostatics

$$
\left.
\begin{aligned}
\nabla.\mathbf{F} - \mathbf{r} &= 0 && \text{Balance law} \\
\mathbf{F} &= \sigma && \text{Flux} \\
\mathbf{r} &= -\rho\mathbf{b} && \text{Source term} \\
\sigma &= C\epsilon && \text{or} \\
\sigma_{ij} &= C_{ijkl}\epsilon_{kl} && \text{Constitutive equation} \\
\epsilon &= \tfrac{1}{2}\left(\nabla\mathbf{u} + \nabla\mathbf{u}^{\mathrm{T}}\right) && \text{or} \\
\epsilon_{kl} &= \tfrac{1}{2}\left(\frac{\partial u_k}{\partial x_l} + \frac{\partial u_l}{\partial x_k}\right) && \text{Kinematic compatibility}
\end{aligned}
\right\} \Rightarrow \boxed{\begin{aligned} \nabla.\sigma &= -\rho\mathbf{b} \\ \frac{\partial}{\partial x_j}\left(C_{ijkl}\frac{\partial u_k}{\partial x_l}\right) &= -\rho b_i \end{aligned}} \tag{27}
$$

where

ρ = Mass density
$\mathbf{b}$ = Body force per unit mass
σ = Stress
ϵ = Strain
C = 4th order elasticity tensor
u = Displacement



Elastostatics: BCs

Operator	Sample	elastostatics	operator order
$L_{2m}(\mathbf{u}) = \mathbf{r}$	$\frac{\partial}{\partial x_j}\left(C_{ijkl}\frac{\partial u_k}{\partial x_l}\right) = -\rho b_i$	$(L_{2m})_i = \frac{\partial}{\partial x_j}\left(C_{ijkl}\frac{\partial (.)_k}{\partial x_l}\right)$	$m = 1 (M = 2)$
$L_u(\mathbf{u}) = \bar{\mathbf{u}}$	$u = \bar{u}$	$L_u = (.)$	$M_u = 0$
$L_f(\mathbf{u}) = \bar{\mathbf{f}}$	$C_{ijkl}\frac{\mathrm{d}u_k}{\mathrm{d}x_l} = \bar{t_i}$	$L_f = C_{ijkl}\frac{\mathrm{d}(.)_k}{\mathrm{d}x_l}$	$M_f = 1$

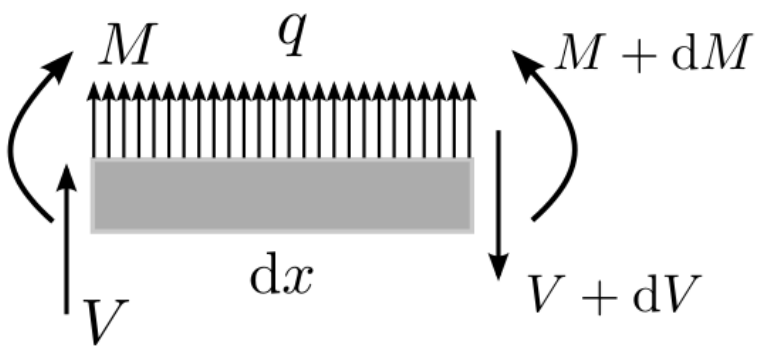
- Can Dirichlet and Neumann BCs be enforced simultaneously at the same point (mixed boundary condition)?
- How about simultaneous enforcement of both BCs at the same point in the same direction (may need local coordinate)?
- Do Neumann boundary conditions correspond to the flux terms?
- Do Neumann boundary conditions fall in the upper half of derivatives ($[m, 2m-1]$)?
- How many boundary conditions do we have at each end point? How is it related to m?
- Verify $M_u + M_f = M - 1$.

Sample Boundary value problems: Euler Bernoulli beam

$$
\left.
\begin{aligned}
\nabla.\mathbf{F} - \mathbf{r} &= 0 && \text{Balance law} \\
\mathbf{F} &= \begin{bmatrix} M \\ V \end{bmatrix} && \text{Flux} \\
\mathbf{r} &= \begin{bmatrix} V \\ q \end{bmatrix} && \text{Source term} \\
M &= EI\kappa && \text{Constitutive equation} \\
\kappa &= \frac{\mathrm{d}^2 y}{\mathrm{d}x^2} && \text{Kinematic compatibility}
\end{aligned}
\right\} \Rightarrow
\begin{aligned}
&\boxed{\begin{aligned} \tfrac{\mathrm{d}M}{\mathrm{d}x} - V &= 0 \\ \tfrac{\mathrm{d}V}{\mathrm{d}x} - q &= 0 \end{aligned}} \\
&\boxed{\frac{\mathrm{d}^2 EI}{\mathrm{d}x^2}\left(\frac{\mathrm{d}^2 y}{\mathrm{d}x^2}\right) - q = 0}
\end{aligned}
\tag{28}
$$

where

M = Momentum
V = Shear force
q = Distributed load
E = Elastic modulus
I = Second moment of area
κ = Curvature
y = Vertical displacement



Euler Bernoulli beam: BCs

Operator	Sample	elastostatics	operator order
$L_{2m}(\mathbf{u}) = \mathbf{r}$	$\frac{\mathrm{d}^2 EI}{\mathrm{d}x^2}\left(\frac{\mathrm{d}^2 y}{\mathrm{d}x^2}\right) = q$	$\frac{\mathrm{d}^2 EI}{\mathrm{d}x^2}\left(\frac{\mathrm{d}^2 (.)}{\mathrm{d}x^2}\right) = q$	$m = 2 (M = 4)$
$L_u(\mathbf{u}) = \bar{\mathbf{u}}$	$u = \begin{bmatrix} \theta \\ y \end{bmatrix} = \begin{bmatrix} \frac{\mathrm{d}y}{\mathrm{d}x} \\ y \end{bmatrix} = \begin{bmatrix} \bar{\theta} \\ \bar{y} \end{bmatrix} = \bar{u}$	$L_u = \begin{bmatrix} \frac{\mathrm{d}(.)}{\mathrm{d}x} \\ (.) \end{bmatrix}$	$M_u = \begin{bmatrix} 1 \\ 0 \end{bmatrix}$
$L_f(\mathbf{u}) = \bar{\mathbf{f}}$	$\begin{bmatrix} EI\frac{\mathrm{d}^2 y}{\mathrm{d}x^2} \\ \frac{\mathrm{d}}{\mathrm{d}x}\left(EI\frac{\mathrm{d}^2 y}{\mathrm{d}x^2}\right) \end{bmatrix} = \begin{bmatrix} \bar{M} \\ \bar{V} \end{bmatrix}$	$L_f = \begin{bmatrix} EI\frac{\mathrm{d}^2 (.)}{\mathrm{d}x^2} \\ \frac{\mathrm{d}}{\mathrm{d}x}\left(EI\frac{\mathrm{d}^2 (.)}{\mathrm{d}x^2}\right) \end{bmatrix}$	$M_f = \begin{bmatrix} 2 \\ 3 \end{bmatrix}$

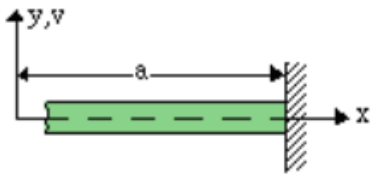
- One and only one of the pair M (Neumann) and θ (Dirichlet) is enforced at each end of the beam.
- One and only one of the pair V (Neumann) and y (Dirichlet) is enforced at each end of the beam.
- Neumann boundary conditions correspond to the flux terms (M and V).
- Neumann boundary conditions fall in the upper half of derivatives ($[m, 2m-1] = [2, 3]$).
- Dirichlet boundary conditions fall in the lower half of derivatives ($[0, m-1] = [0, 1]$).
- There are two boundary conditions at each end point (equal to $m = M/2$).
- $M_u + M_f = M - 1$.

Euler Bernoulli beam: pairs of boundary conditions

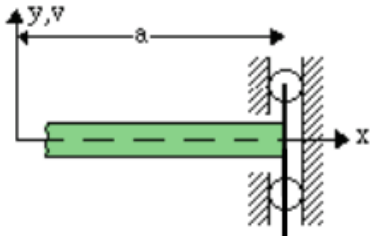
$y = \bar{y}, V = ?$
Essential y BC

$y = ?, V = \bar{V}$
Natural$V$ BC

$\theta = \bar{\theta}, M = ?$
Essential θ BC

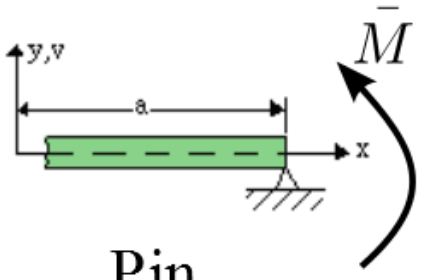


Fixed

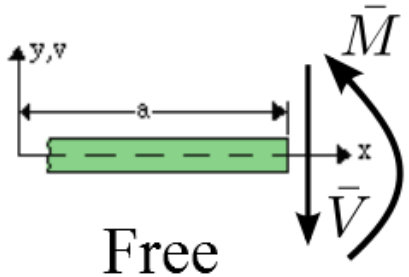


Glided

$\theta = ?, M = \bar{M}$
Natural$M$ BC



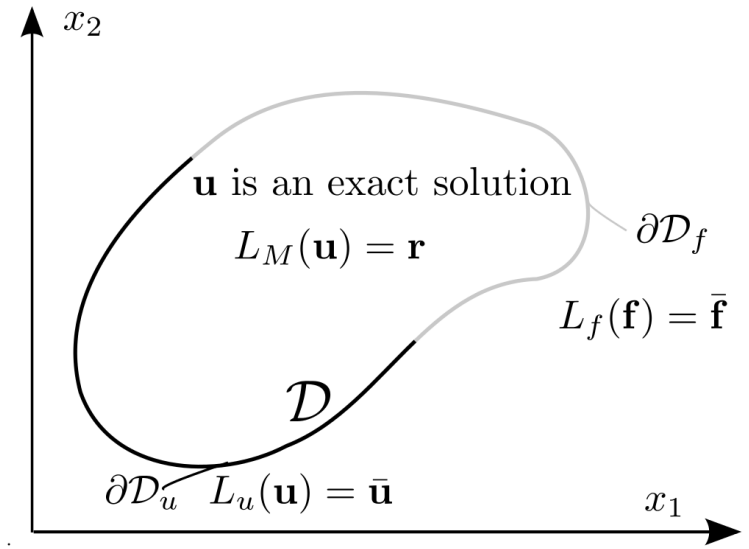
Pin



Free

- For each pair of Neumann and Dirichlet (Natural and Essential) boundary conditions, one and only one is specified.

(Extended) strong statement of the problem

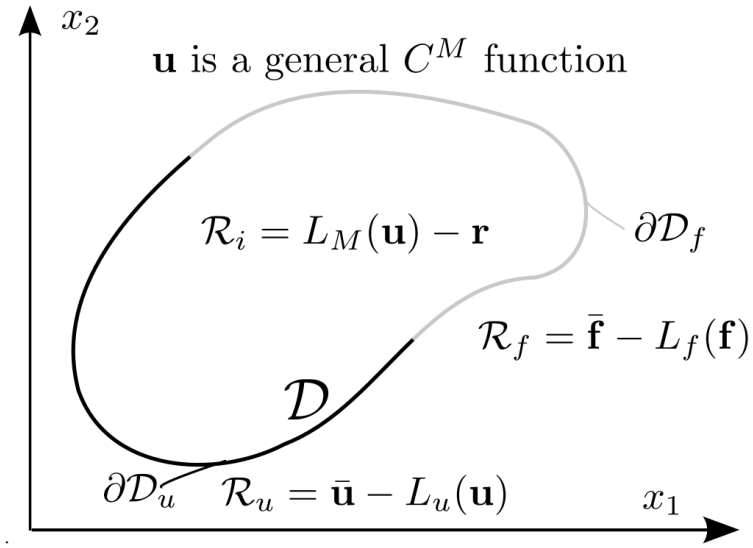


The function $\mathbf{u} \in C^M(\bar{\mathcal{D}})$ (function with M continuous derivatives on the closure of the set $\mathcal{D}$) is the strong solution to the boundary value problem, if it satisfies:

1. (Partial) differential equation: $\forall \mathbf{x} \in \mathcal{D}: \quad L_M(\mathbf{u}) = \mathbf{r}$.
2. Essential (Dirichlet) BC: $\forall \mathbf{x} \in \partial\mathcal{D}_u: \quad L_u(\mathbf{u}) = \bar{\mathbf{u}}$.
3. Natural (Neumann) BC: $\forall \mathbf{x} \in \partial\mathcal{D}_f: \quad L_f(\mathbf{f}) = \bar{\mathbf{f}}$.

Weak form

Weak solution: construction of residual



For the function $\mathbf{u} \in C^M(\bar{\mathcal{D}})$ the residuals for the boundary value problem are defined as,

1. (Partial) differential equation residual on $\mathcal{D}$: $\mathcal{R}_i(\mathbf{u}) = \quad L_M(\mathbf{u}) - \mathbf{r}$.
2. Essential (Dirichlet) BC residual on $\partial\mathcal{D}_u$: $\mathcal{R}_u = \bar{\mathbf{u}} - L_u(\mathbf{u})$.
3. Natural (Neumann) BC residual on $\partial\mathcal{D}_f$: $\mathcal{R}_f = \bar{\mathbf{f}} - L_f(\mathbf{u})$.

Clearly, $\mathbf{u}$ is a strong solution to the boundary value problem if all the residuals are identically zero.

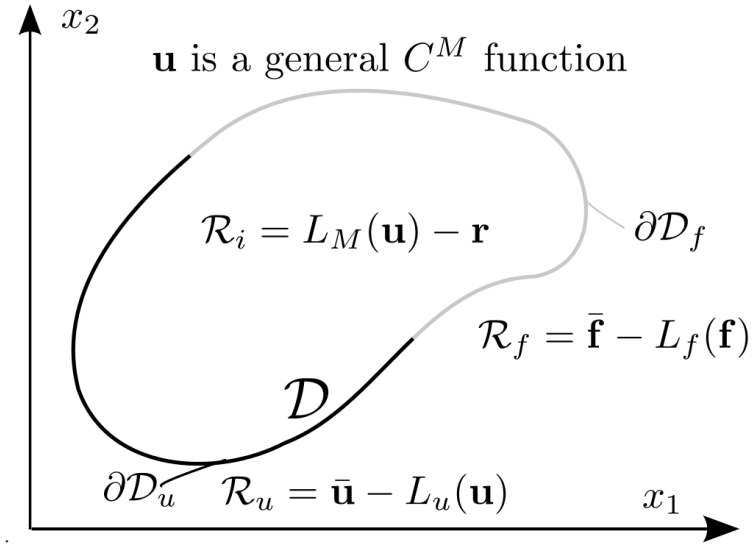
Weighted residual statement

First, we define the following function spaces,

$$\mathcal{W} = C^0(\bar{\mathcal{D}}). \tag{29a}$$

$$\mathcal{V} = C^M(\bar{\mathcal{D}}). \tag{29b}$$

We seek a weak solution, $\mathbf{u} \in \mathcal{V}$, that satisfies,



$$\begin{aligned} \forall \mathbf{w} \in \mathcal{W} \ : \int_{\mathcal{D}} \mathbf{w}.\mathcal{R}_i \,\mathrm{dv} + \int_{\partial\mathcal{D}_u} \mathbf{w}.\mathcal{R}_u \,\mathrm{ds} + \int_{\partial\mathcal{D}_f} \mathbf{w}.\mathcal{R}_f \,\mathrm{ds} &= \\ \int_{\mathcal{D}} \mathbf{w}.(L_M(\mathbf{u}) - \mathbf{r}) \,\mathrm{dv} + \int_{\partial\mathcal{D}_u} \mathbf{w}.(\bar{\mathbf{u}} - L_u(\mathbf{u})) \,\mathrm{ds} + \int_{\partial\mathcal{D}_f} \mathbf{w}.(\bar{\mathbf{f}} - L_f(\mathbf{u})) \,\mathrm{ds} &= 0 \end{aligned}$$

Basically, a weak solution "weakly" satisfies all the PDEs and BCs by requiring the residuals to be zero for arbitrary averaging schemes (through arbitrariness of weight functions $\mathbf{w}$). Next, we want to investigate the relationship between weak and strong solutions.

Equivalence of weak and strong solutions

We do not discuss the existence and uniqueness conditions for either system at the moment. Rather, show that weak and strong solutions are equivalent. That is, a strong solution is a weak solution and wise versa. From previous slides and the definition of a strong solution we have: Function $\mathbf{u} \in \mathcal{V}$ is a strong solution if,

$$\forall \mathbf{x} \in \mathcal{D}: \quad \mathcal{R}_i(\mathbf{u}) = L_M(\mathbf{u}) - \mathbf{r} = 0 \quad \text{PDE} \tag{30a}$$

$$\forall \mathbf{x} \in \partial\mathcal{D}_u: \quad \mathcal{R}_u(\mathbf{u}) = \bar{\mathbf{u}} - L_u(\mathbf{u}) = 0 \quad \text{Essential BC} \tag{30b}$$

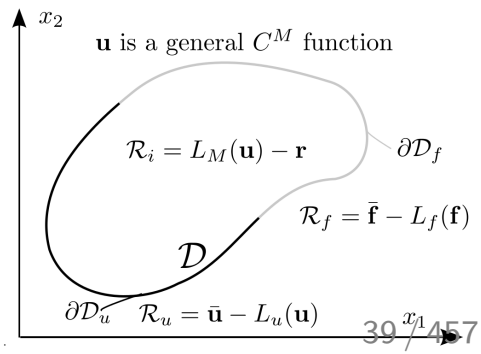
$$\forall \mathbf{x} \in \partial\mathcal{D}_f: \quad \mathcal{R}_f(\mathbf{u}) = \bar{\mathbf{f}} - L_f(\mathbf{f}) = 0 \quad \text{Natural BC} \tag{30c}$$

and from previous slide a weak solution $\mathbf{u} \in \mathcal{V}$ satisfies:

$$\begin{aligned} \forall \mathbf{w} \in \mathcal{W} \ : \int_{\mathcal{D}} \mathbf{w}.\mathcal{R}_i \, \mathrm{dv} + \int_{\partial\mathcal{D}_u} \mathbf{w}.\mathcal{R}_u \, \mathrm{ds} + \int_{\partial\mathcal{D}_f} \mathbf{w}.\mathcal{R}_f \, \mathrm{ds} \quad &= \\ \int_{\mathcal{D}} \mathbf{w}.(L_M(\mathbf{u}) - \mathbf{r}) \, \mathrm{dv} + \int_{\partial\mathcal{D}_u} \mathbf{w}.(\bar{\mathbf{u}} - L_u(\mathbf{u})) \, \mathrm{ds} + \int_{\partial\mathcal{D}_f} \mathbf{w}.(\bar{\mathbf{f}} - L_f(\mathbf{u})) \, \mathrm{ds} \quad &= 0 \end{aligned} \tag{31}$$

Proof:

- $\mathbf{u}$ is a strong solution $\Rightarrow$ $\mathbf{u}$ is a weak solution.
  The proof is trivial. Since all residuals $\mathcal{R}_i, \mathcal{R}_u$, and $\mathcal{R}_f$ are identically zero, the equation (31) is trivially satisfied.
- $\mathbf{u}$ is a weak solution $\Rightarrow$ $\mathbf{u}$ is a strong solution.
  This is the more difficult direction. What would be the proof process?



Lemma needed for the proof of equivalence of strong and weak forms

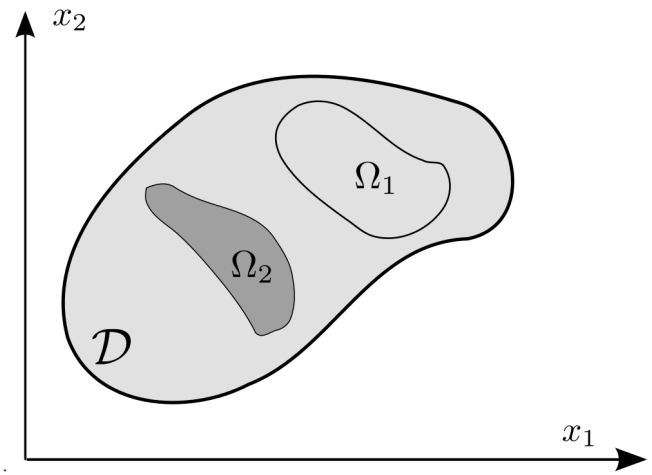
Before the proof, we need a theorem very similar to localization theorem which states for $g \in C^0(\mathcal{D})$:

$$\forall \Omega \subset \mathcal{D} : \int_{\Omega} \mathbf{g}(\mathbf{x})\, \mathrm{dv} = \mathbf{0} \quad \Rightarrow \quad \forall x \in \mathcal{D} : \; g(\mathbf{x}) = 0$$

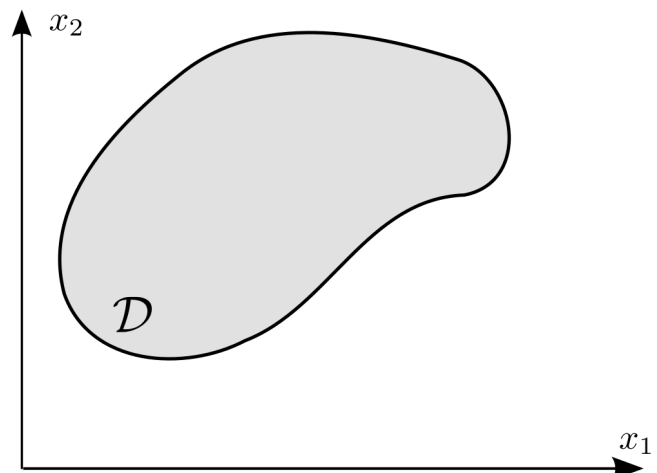
In localization theorem, the domain of integration was arbitrary, whereas in our case, the domain of integration is fixed ($\mathcal{D}, \partial\mathcal{D}_u$, and $\partial\mathcal{D}_f$), while the integrand is multiplied by an arbitrary continuous function $\mathbf{w}$. Basically, we want to prove:

$$\forall \mathbf{w} \in C^0(\mathcal{D}) : \int_{\mathcal{D}} w(x) g(\mathbf{x})\, \mathrm{dv} = \mathbf{0} \quad \Rightarrow \quad \forall x \in \mathcal{D} : \; g(\mathbf{x}) = 0 \tag{32}$$

Here $w(x)$ and $g(x)$ are scalar functions in $C^0(\bar{\mathcal{D}})$.



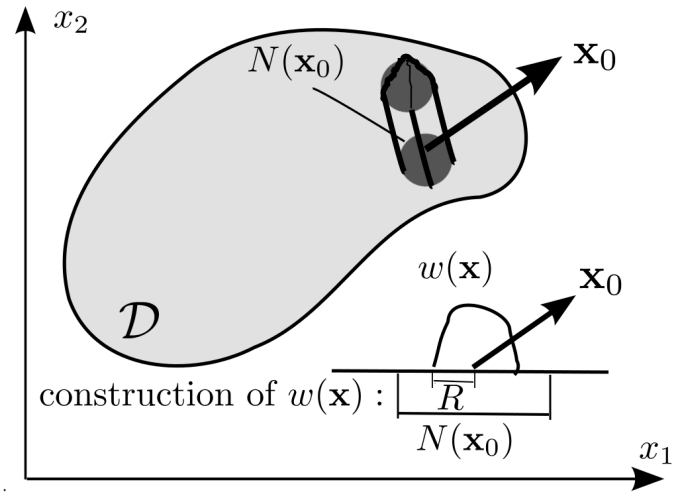
Localization Theorem: $\forall \Omega \subset \mathcal{D} : \int_{\Omega} g(\mathbf{x})\, \mathrm{dv} = 0 \Rightarrow \forall x \in \mathcal{D} : \; g(\mathbf{x}) = 0$



$\forall w \in C^0(\mathcal{D}) : \int_{\mathcal{D}} w(\mathbf{x}).g(\mathbf{x})\, \mathrm{dv} = 0 \Rightarrow \forall x \in \mathcal{D} : \; g(\mathbf{x}) = 0$

Lemma continued

Similar to previous proof, we assume the function is nonzero at $\mathbf{x}_0$. Without loss of generality we assume the function is positive at $\mathbf{x}_0$. Since, $g(\mathbf{x})$ is continuous, we can find a neighborhood of $\mathbf{x}_0$ such as $N(\mathbf{x}_0)$ where $\forall \mathbf{x} \in N(\mathbf{x}_0) : \; g(\mathbf{x}) > 0$. We construct a weight function $w(\mathbf{x})$ which is positive within $N(\mathbf{x}_0)$ and zero outside. For example, if a line/ circle / sphere of radius R fits within $N(\mathbf{x}_0)$, one candidate for $w(\mathbf{x})$ is:



$$w(\mathbf{x}) = \begin{cases} 1 - (\mathbf{x} - \mathbf{x}_0)^2 / R^2 & |\mathbf{x} - \mathbf{x}_0| < R \\ 0 & \text{other wise} \end{cases} \tag{33}$$

then

$$\int_{\mathcal{D}} w(x) g(\mathbf{x}) \, \mathrm{dv} = \int_{\mathcal{D} \setminus N(\mathbf{x}_0)} w(x) g(\mathbf{x}) \, \mathrm{dv} + \int_{N(\mathbf{x}_0)} w(x) g(\mathbf{x}) \, \mathrm{dv} = 0 + \int_{N(\mathbf{x}_0)} w(x) g(\mathbf{x}) \, \mathrm{dv} > 0 \tag{34}$$

because both $g(\mathbf{x})$ and $w(\mathbf{x})$ are positive within $N(\mathbf{x}_0)$. So $g(\mathbf{x}_0)$ cannot be nonzero. Since $\mathbf{x}_0$ is arbitrary we conclude that $\forall \mathbf{x} \in : \; g(\mathbf{x}) = 0$.

Equivalence of weak (weighted residual) and strong forms

For the second part of the proof we need to demonstrate that if for $\mathbf{u}$ the weighted residual (31) holds:

$$\forall \mathbf{w} \in \mathcal{W} \ : \int_{\mathcal{D}} \mathbf{w}.\mathcal{R}_i \, \mathrm{dv} + \int_{\partial \mathcal{D}_u} \mathbf{w}.\mathcal{R}_u \, \mathrm{ds} + \int_{\partial \mathcal{D}_f} \mathbf{w}.\mathcal{R}_f \, \mathrm{ds} = 0 \tag{35}$$

then it would be a strong solution. That is, all residual would be identically zero. The stages of proof are as follows:

- Restrict $\mathbf{w}$ to "bubble functions" that are zero on $\partial\mathcal{D}$. We define $\mathcal{W}_{00} = \{\mathbf{w} \mid \mathbf{w} \in \mathcal{W}, \ \mathbf{w}(\partial\mathcal{D}) = 0\}$. The we have,

$$\forall \mathbf{w} \in \mathcal{W}_{00} \ : \int_{\mathcal{D}} \mathbf{w}.\mathcal{R}_i \, \mathrm{dv} + \int_{\partial \mathcal{D}_u} \mathbf{w}.\mathcal{R}_u \, \mathrm{ds} + \int_{\partial \mathcal{D}_f} \mathbf{w}.\mathcal{R}_f \, \mathrm{ds} = \int_{\mathcal{D}} \mathbf{w}.\mathcal{R}_i \, \mathrm{dv} = 0$$

  Now, for all points in the interior of $\mathcal{D}$ we can employ the previous lemma to show that,

$$\forall \mathbf{w} \in \mathcal{W}_{00} \ : \int_{\mathcal{D}} \mathbf{w}.\mathcal{R}_i \, \mathrm{dv} = 0 \quad \Rightarrow \quad \boxed{\forall \mathbf{x} \in \mathcal{D} : \ \mathcal{R}_i = 0.} \tag{36}$$

  That is, the interior residual is identically equal to zero (strong form). Thus, the residual statement (35) simplifies to,

$$\forall \mathbf{w} \in \mathcal{W} \ : \int_{\partial \mathcal{D}_u} \mathbf{w}.\mathcal{R}_u \, \mathrm{ds} + \int_{\partial \mathcal{D}_f} \mathbf{w}.\mathcal{R}_f \, \mathrm{ds} = 0 \tag{37}$$

Equivalence of weak (weighted residual) and strong forms

- Similar to previous step, we want to focus only at one residual integral at a time. So, we define,

$$\boxed{\mathcal{W}_0 = \{\mathbf{w} \,|\mathbf{w} \in \mathcal{W},\ \mathbf{w}(\partial\mathcal{D}_u) = 0\}.} \tag{38}$$

  Then, from previous result we obtain,

$$\forall \mathbf{w} \in \mathcal{W}_0 \ : \int_{\partial\mathcal{D}_u} \mathbf{w}.\mathcal{R}_u \,\mathrm{ds} + \int_{\partial\mathcal{D}_f} \mathbf{w}.\mathcal{R}_f \,\mathrm{ds} = \int_{\partial\mathcal{D}_f} \mathbf{w}.\mathcal{R}_f \,\mathrm{ds} = 0$$

  Using the same line of proof used in the lemma (that is the integral of a function weighted with arbitrary functions on a set being zero implies that the function is identically zero) we observe:

$$\forall \mathbf{w} \in \mathcal{W}_0 \ : \int_{\partial\mathcal{D}_f} \mathbf{w}.\mathcal{R}_f \,\mathrm{ds} = 0 \quad \Rightarrow \boxed{\forall \mathbf{x} \in \partial\mathcal{D}_f \ :\ \mathcal{R}_f = 0} \tag{39}$$

- Finally we note that equations (37) and (39) imply:

$$\forall \mathbf{w} \in \mathcal{W} \ : \int_{\partial\mathcal{D}_u} \mathbf{w}.\mathcal{R}_u \,\mathrm{ds} = 0 \tag{40}$$

  As the final step, using the same line of proof in the lemma this equation yields:

$$\forall \mathbf{w} \in \mathcal{W} \ : \int_{\partial\mathcal{D}_u} \mathbf{w}.\mathcal{R}_u \,\mathrm{ds} = 0 \quad \Rightarrow \quad \boxed{\forall \mathbf{x} \in \partial\mathcal{D}_u \ :\ \mathcal{R}_u = 0} \tag{41}$$

To summarize, through equations (36), (39), (41), and (30) we complete the proof that the solution to the weighted residual statement (weak statement) is also a strong solution.

Motivation for different statement of the weighted residual

- While in weighted residual statement (31) all residuals $\mathcal{R}_i$, $\mathcal{R}_f$, and $\mathcal{R}_u$ are weighted simply by w in practice the weight functions are the "dual" quantities that are generated by successive integration by part of the interior integral. Examples of dual quantities are $\mathbf{u}$ and $\mathbf{t} = \sigma.\mathbf{n}$ for elastostatics, $F = EA\sigma$ and u in the bar example, and $[M, V]^{\mathrm{T}}$ and $[y', y]^{\mathrm{T}}$ for Euler Bernoulli beam. We will further discuss why this particular structure of weight function is beneficial when we attempt to derive the weak statement.
- Table below shows what the choices of the weight functions would be for Euler Bernoulli beam instead of simply using w for all residuals. We will observe how this particular form leads to a reasonable weak statement.

Term	Domain	Weight		Residual	
		function	order	function	order
Interior	$\partial\mathcal{D}$	w	0	$\mathcal{R}_i = \frac{\mathrm{d}^2}{\mathrm{d}x^2}\left(EI\frac{\mathrm{d}^2 y}{\mathrm{d}x^2}\right) - q$	4
Natural Boundary	$\partial\mathcal{D}_f$	$\mathbf{w}_f = \begin{bmatrix} -w' \\ w \end{bmatrix}$	$\begin{bmatrix} 1 \\ 0 \end{bmatrix}$	$\mathcal{R}_f = \begin{bmatrix} \bar{M} - M(y) \\ \bar{V} - V(y) \end{bmatrix}$	$\begin{bmatrix} 2 \\ 3 \end{bmatrix}$
Essential Boundary	$\partial\mathcal{D}_u$	$\mathbf{w}_u = \begin{bmatrix} -M(w) \\ V(w) \end{bmatrix}$	$\begin{bmatrix} 2 \\ 3 \end{bmatrix}$	$\mathcal{R}_u = \begin{bmatrix} \bar{y}' - y' \\ \bar{y} - y \end{bmatrix}$	$\begin{bmatrix} 1 \\ 0 \end{bmatrix}$

where $M(u) = EI\frac{\mathrm{d}^2 u}{\mathrm{d}x^2}$ and $V(u) = \frac{\mathrm{d}EI}{\mathrm{d}x}\left(\frac{\mathrm{d}^2 u}{\mathrm{d}x^2}\right)$.

Motivation for different statement of the weighted residual

Term	Domain	Weight		Residual	
		function	order	function	order
Interior	$\partial\mathcal{D}$	w	0	$\mathcal{R}_i = \frac{\mathrm{d}^2}{\mathrm{d}x^2}\left(EI\frac{\mathrm{d}^2 y}{\mathrm{d}x^2}\right) - q$	4
Natural Boundary	$\partial\mathcal{D}_f$	$\mathbf{w}_f = \begin{bmatrix} -w' \\ w \end{bmatrix}$	$\begin{bmatrix} 1 \\ 0 \end{bmatrix}$	$\mathcal{R}_f = \begin{bmatrix} \bar{M} - M(y) \\ \bar{V} - V(y) \end{bmatrix}$	$\begin{bmatrix} 2 \\ 3 \end{bmatrix}$
Essential Boundary	$\partial\mathcal{D}_u$	$\mathbf{w}_u = \begin{bmatrix} -M(w) \\ V(w) \end{bmatrix}$	$\begin{bmatrix} 2 \\ 3 \end{bmatrix}$	$\mathcal{R}_u = \begin{bmatrix} \bar{y}' - y' \\ \bar{y} - y \end{bmatrix}$	$\begin{bmatrix} 1 \\ 0 \end{bmatrix}$

- What is the maximum derivative order for trial functions (y)? 4 for the $\mathcal{R}_i$.
- What is the function space for trial functions? $C^4(\mathcal{D})$.
- What is the maximum derivative order for weight functions? 3 for $\mathbf{w}_u$.
- What is the function space for weight functions? $C^3(\mathcal{D})$.
- So for the current form of the weighted residual statement trial and weight function spaces are $y \in C^4(\mathcal{D})$ and $w \in C^3(\mathcal{D})$.

Dropping the WR term on Essential Boundary $\partial\mathcal{D}_u$.

We noticed that in the original statement $w \in C^3(\mathcal{D})$ because of the weighted residual term on the essential boundary $\partial\mathcal{D}_u$. Suppose we could modify the weighted residual to (dropping the integral on $\partial\mathcal{D}_u$).

$$\forall \mathbf{w} \in \mathcal{W}^{\mathrm{WRS}} \; : \int_{\mathcal{D}} \mathbf{w}.\mathcal{R}_i \, \mathrm{dv} + \int_{\partial\mathcal{D}_f} \mathbf{w}.\mathcal{R}_f \, \mathrm{ds} = 0 \tag{42}$$

We will shortly discuss what the space $\mathcal{W}^{\mathrm{WRS}}$ would be. The weight and residual functions are:

Term	Domain	Weight		Residual	
		function	order	function	order
Interior	$\partial\mathcal{D}$	w	0	$\mathcal{R}_i = \frac{\mathrm{d}^2}{\mathrm{d}x^2}\left(EI\frac{\mathrm{d}^2 y}{\mathrm{d}x^2}\right) - q$	4
Natural Boundary	$\partial\mathcal{D}_f$	$\mathbf{w}_f = \begin{bmatrix} -w' \\ w \end{bmatrix}$	$\begin{bmatrix} 1 \\ 0 \end{bmatrix}$	$\mathcal{R}_f = \begin{bmatrix} \bar{M} - M(y) \\ \bar{V} - V(y) \end{bmatrix}$	$\begin{bmatrix} 2 \\ 3 \end{bmatrix}$

- What is the maximum derivative order for trial functions? 4 for the $\mathcal{R}_i$.
- What is the function space for trial functions? $C^4(\mathcal{D})$.
- What is the maximum derivative order for weight functions? 1.
- What would be the function space for the weight functions? $C^1(\mathcal{D})$.
- Trial and weight function spaces are $y \in C^4(\mathcal{D})$ and $w \in C^1(\mathcal{D})$.

Dropping the WR term on Essential Boundary $\partial\mathcal{D}_u$.

Term	Domain	Weight		Residual	
		function	order	function	order
Interior	$\partial\mathcal{D}$	w	0	$\mathcal{R}_i = \frac{\mathrm{d}^2}{\mathrm{d}x^2}\left(EI\frac{\mathrm{d}^2y}{\mathrm{d}x^2}\right) - q$	4
Natural Boundary	$\partial\mathcal{D}_f$	$\mathbf{w}_f = \begin{bmatrix} -w' \\ w \end{bmatrix}$	$\begin{bmatrix} 1 \\ 0 \end{bmatrix}$	$\mathcal{R}_f = \begin{bmatrix} \bar{M} - M(y) \\ \bar{V} - V(y) \end{bmatrix}$	$\begin{bmatrix} 2 \\ 3 \end{bmatrix}$

As mentioned before, the differential equation $\frac{\mathrm{d}^2}{\mathrm{d}x^2}\left(EI\frac{\mathrm{d}^2y}{\mathrm{d}x^2}\right) - q$ is 4th order requiring the very strict requirement that $y \in C^4(\mathcal{D})$, while in reality the balance law (balance and linear and angular momentum in this case) only requires 2nd order derivatives ($M = EI\frac{\mathrm{d}^2y}{\mathrm{d}x^2}$). This is an undesirable consequence of going from a balance law to a strong statement, where we added additional derivative terms on y.
Is there anyway, to reverse this process and have larger solution space for y?
Yes, we notice that in the interior weighted residual expression $w.\mathcal{R}_i = w.\left[\frac{\mathrm{d}^2}{\mathrm{d}x^2}\left(EI\frac{\mathrm{d}^2y}{\mathrm{d}x^2}\right) - q\right]$, w appears only as a function value while y is a $4th$ order derivative. We can use **integration by part** to transfer derivatives from trial to weight function (y to w). For example in the following,

$$\int_a^b f'(x)g(x)\,\mathrm{d}x = f(b).g(b) - f(a).g(a) - \int_a^b f(x)g'(x)\,\mathrm{d}x \tag{43}$$

We transfer derivatives from $f'(x)$ to $g(x)$.

Summary: Variations of Weighted residual statement

As observed, strong satisfaction of the essential boundary conditions relaxes the differentiability constraints on the weight functions and as will be shown facilitates the derivation of the weak statement. Here we summarize three variations of the weighted residual statement:

- Weak enforcement of all conditions:

$$\begin{aligned} &\text{Find } \mathbf{u} \in \mathcal{V} = \{\mathbf{v} \,|\mathbf{v} \in C^M(\mathcal{D}\} \text{ such that} \\ &\forall \mathbf{w} \in \mathcal{W} \;: \int_{\mathcal{D}} \mathbf{w}.\mathcal{R}_i \,\mathrm{dv} + \int_{\partial\mathcal{D}_u} \mathbf{w}.\mathcal{R}_u \,\mathrm{ds} + \int_{\partial\mathcal{D}_f} \mathbf{w}.\mathcal{R}_f \,\mathrm{ds} \quad = \\ &\int_{\mathcal{D}} \mathbf{w}.(L_M(\mathbf{u}) - \mathbf{r}) \,\mathrm{dv} + \int_{\partial\mathcal{D}_u} \mathbf{w}.(\bar{\mathbf{u}} - L_u(\mathbf{u})) \,\mathrm{ds} + \int_{\partial\mathcal{D}_f} \mathbf{w}.(\bar{\mathbf{f}} - L_f(\mathbf{u})) \,\mathrm{ds} \quad = 0 \end{aligned} \tag{44}$$

- Weak enforcement of differential equation and natural BC, strong enforcement of essential BC (the most common form):

$$\begin{aligned} &\text{Find } \mathbf{u} \in \mathcal{V} = \{\mathbf{v} \,|\mathbf{v} \in C^M(\mathcal{D}, L_u(\mathbf{u}) = \bar{\mathbf{u}}\} \text{ such that} \\ &\forall \mathbf{w} \in \mathcal{W} \;: \int_{\mathcal{D}} \mathbf{w}.\mathcal{R}_i \,\mathrm{dv} + \int_{\partial\mathcal{D}_f} \mathbf{w}.\mathcal{R}_f \,\mathrm{ds} \quad = \\ &\int_{\mathcal{D}} \mathbf{w}.(L_M(\mathbf{u}) - \mathbf{r}) \,\mathrm{dv} + \int_{\partial\mathcal{D}_f} \mathbf{w}.(\bar{\mathbf{f}} - L_f(\mathbf{u})) \,\mathrm{ds} \quad = 0 \end{aligned} \tag{45}$$

- Weak enforcement of differential equation and strong enforcement of all BCs (natural and essential) (for example refer to Bathe's book, section 3.3.3)

$$\begin{aligned} &\text{Find } \mathbf{u} \in \mathcal{V} = \{\mathbf{v} \,|\mathbf{v} \in C^M(\mathcal{D}, L_u(\mathbf{u}) = \bar{\mathbf{u}}, L_f(\mathbf{u}) = \bar{\mathbf{f}}\} \text{ such that} \\ &\forall \mathbf{w} \in \mathcal{W} \;: \int_{\mathcal{D}} \mathbf{w}.\mathcal{R}_i \,\mathrm{dv} = \int_{\mathcal{D}} \mathbf{w}.(L_M(\mathbf{u}) - \mathbf{r}) \,\mathrm{dv} \quad = 0 \end{aligned} \tag{46}$$

Least Square method

- Like Weighted residual method, least square method is an integral statement that is equivalent to the strong form.
- The basis of weighted residual method is the following statement:

$$g(x) = 0 \text{ almost everywhere (a.e.) in } \mathcal{D} \Leftrightarrow \int_{\mathcal{D}} g^2(x)\,\mathrm{d}x = 0 \tag{47}$$

or for a continuus function $g(x)$

$$\forall x \in \mathcal{D}\ g(x) = 0 \Leftrightarrow \int_{\mathcal{D}} g^2(x)\,\mathrm{d}x = 0 \tag{48}$$

These statements are trivial as the integrand $g^2(x) \geq 0$ so the only way the integral is zero is when the function is a.e. zero. If the function is continuous then $g(x)$ must be identically zero.

- Basically, the least square method minimizes a general residual error $\mathcal{R}$ within a given function space $\mathcal{V}$:

$$\text{Find } \mathbf{u} \in \mathcal{V} \text{ such that } R^2(\mathbf{u}) = \int \mathcal{R}^2(\mathbf{u}) \text{ is minimized within } \mathcal{V} \tag{49}$$

Based on which residuals $(\mathcal{R}_i, \mathcal{R}_u, \mathcal{R}_f)$ are strongly satisfied, we can weakly satisfy the remainder of the residuals using the least square approach. Clearly, in continuum setting (not a discrete solution) the least square value is zero (because all residuals are exactly equal to zero).

- In discrete setting we use the least square approach to find the numerical solution that has the smallest residual error withing a space of functions $\mathcal{V}$.

Summary: Variations of continuum Least Square method

- Weak enforcement of all conditions:

$$\begin{aligned} &\text{Find } \mathbf{u} \in \mathcal{V} = \{\mathbf{v} \,|\mathbf{v} \in C^M(\mathcal{D}\} \text{ such that} \\ &\int_{\mathcal{D}} \mathcal{R}_i^2 \,\mathrm{dv} + \int_{\partial\mathcal{D}_u} \mathcal{R}_u^2 \,\mathrm{ds} + \int_{\partial\mathcal{D}_f} \mathcal{R}_f^2 \,\mathrm{ds} \quad = \\ &\int_{\mathcal{D}} (L_M(\mathbf{u}) - \mathbf{r})^2 \,\mathrm{dv} + \int_{\partial\mathcal{D}_u} (\bar{\mathbf{u}} - L_u(\mathbf{u}))^2 \,\mathrm{ds} + \int_{\partial\mathcal{D}_f} (\bar{\mathbf{f}} - L_f(\mathbf{u}))^2 \,\mathrm{ds} \quad = 0 \end{aligned} \tag{50}$$

- Weak enforcement of differential equation and natural BC, strong enforcement of essential BC (the most common form):

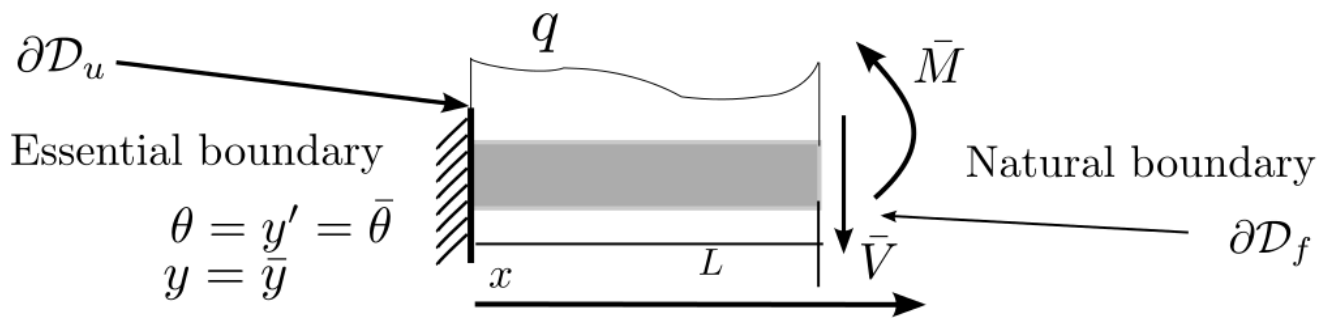
$$\begin{aligned} &\text{Find } \mathbf{u} \in \mathcal{V} = \{\mathbf{v} \,|\mathbf{v} \in C^M(\mathcal{D}, L_u(\mathbf{u}) = \bar{\mathbf{u}}\} \text{ such that} \\ &\int_{\mathcal{D}} \mathcal{R}_i^2 \,\mathrm{dv} + \int_{\partial\mathcal{D}_f} \mathcal{R}_f^2 \,\mathrm{ds} \quad = \\ &\int_{\mathcal{D}} (L_M(\mathbf{u}) - \mathbf{r})^2 \,\mathrm{dv} + \int_{\partial\mathcal{D}_f} (\bar{\mathbf{f}} - L_f(\mathbf{u}))^2 \,\mathrm{ds} \quad = 0 \end{aligned} \tag{51}$$

- Weak enforcement of differential equation and strong enforcement of all BCs (natural and essential) (for example refer to Bathe's book, section 3.3.3)

$$\begin{aligned} &\text{Find } \mathbf{u} \in \mathcal{V} = \{\mathbf{v} \,|\mathbf{v} \in C^M(\mathcal{D}, L_u(\mathbf{u}) = \bar{\mathbf{u}}, L_f(\mathbf{u}) = \bar{\mathbf{f}}\} \text{ such that} \\ &\int_{\mathcal{D}} \mathcal{R}_i^2 \,\mathrm{dv} = \int_{\mathcal{D}} (L_M(\mathbf{u}) - \mathbf{r})^2 \,\mathrm{dv} \quad = 0 \end{aligned} \tag{52}$$

Weighted residual statement to Weak statement

To demonstrate the process of deriving the weak statement from the weighted residual statement consider the following problem:



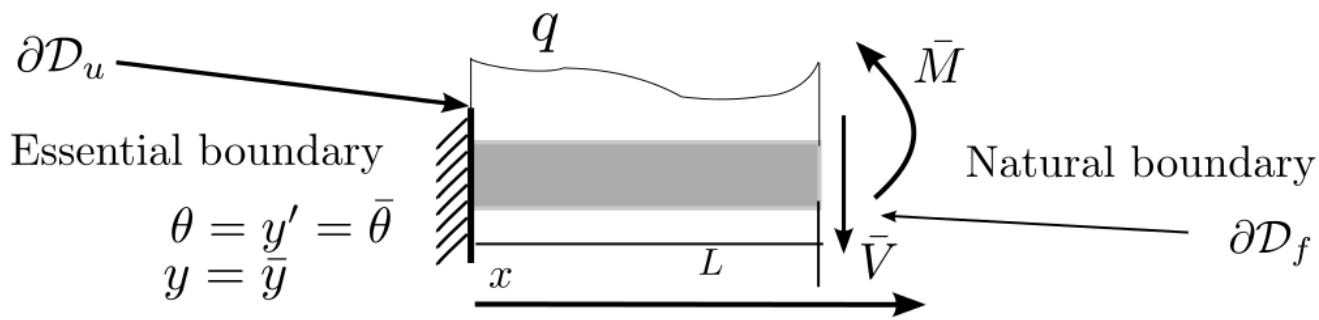
The residuals for this problem are:

$$\begin{aligned}
\mathcal{R}_i &= \frac{\mathrm{d}^2}{\mathrm{d}x^2}\left(EI\frac{\mathrm{d}^2 y}{\mathrm{d}x^2}\right) - q && \text{Interior residual for} && \mathcal{D} = [0, L] \\
\mathcal{R}_f &= \begin{bmatrix} \bar{M} - M \\ \bar{V} - V \end{bmatrix} && \text{Natural BC residual for} && \partial\mathcal{D}_f = \{L\} \\
\mathcal{R}_u &= \begin{bmatrix} \bar{\theta} - \theta \\ \bar{y} - y \end{bmatrix} && \text{Essential BC residual for} && \partial\mathcal{D}_u = \{0\}
\end{aligned} \tag{53}$$

As mentioned previously, we want to drop the weighted residual term for essential boundary condition (why?). Accordingly, we need to strongly enforce the essential boundary condition (This is why this is called "essential" boundary condition). That is, we require:

$$\mathcal{R}_u = \begin{bmatrix} \bar{\theta} - \theta \\ \bar{y} - y \end{bmatrix} = 0 \quad \text{at } x = 0 \ (\partial\mathcal{D}_u). \tag{54}$$

Weighted residual statement to Weak statement



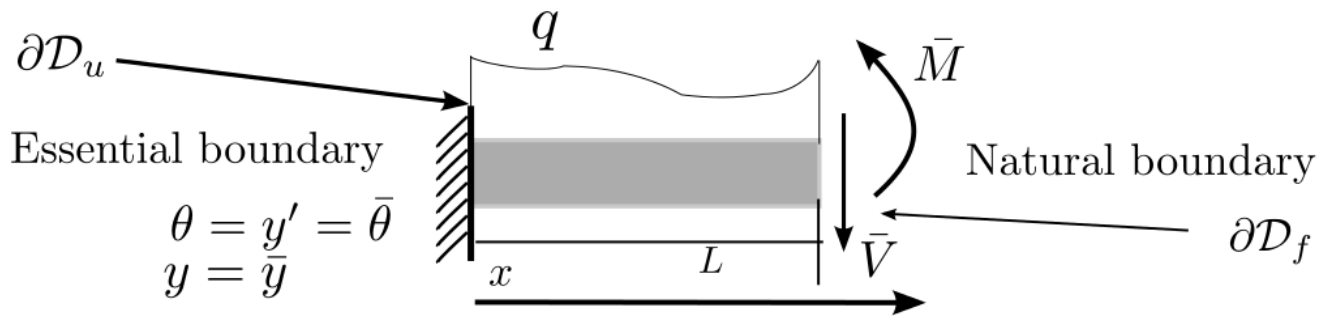
Since we strongly enforce the essential boundary condition, the weighted residual for this problem simplifies to:

$$
\begin{aligned}
0 \quad &= \int_{\mathcal{D}} w\mathcal{R}_i(y)\,\mathrm{d}v + \int_{\partial\mathcal{D}_f} \mathbf{w}_f\mathcal{R}_f(y)\,\mathrm{d}\mathbf{s} \\
&= \int_0^L w\left(\frac{\mathrm{d}^2}{\mathrm{d}x^2}\left(EI\frac{\mathrm{d}^2y}{\mathrm{d}x^2}\right) - q\right)\,\mathrm{d}x + \begin{bmatrix} -\frac{\mathrm{d}w}{\mathrm{d}x} \\ w \end{bmatrix} \cdot \begin{bmatrix} \bar{M} - M \\ \bar{V} - V \end{bmatrix} |_{x=L} \\
&= \int_0^L w\left(\frac{\mathrm{d}^2}{\mathrm{d}x^2}\left(EI\frac{\mathrm{d}^2y}{\mathrm{d}x^2}\right) - q\right)\,\mathrm{d}x - \frac{\mathrm{d}w}{\mathrm{d}x}(\bar{M} - M(y))|_{x=L} + w(\bar{V} - V(y))|_{x=L}
\end{aligned}
\tag{55}
$$

Next, we transfer derivatives from y to w (trial function to weight function). We note that

$$
\begin{aligned}
&\int_0^L w\frac{\mathrm{d}^2}{\mathrm{d}x^2}\left(EI\frac{\mathrm{d}^2y}{\mathrm{d}x^2}\right)\,\mathrm{d}x = \int_0^L \left[-\frac{\mathrm{d}w}{\mathrm{d}x}\frac{\mathrm{d}}{\mathrm{d}x}EI\left(\frac{\mathrm{d}^2y}{\mathrm{d}x^2}\right)\right]\,\mathrm{d}x + \left[w\frac{\mathrm{d}}{\mathrm{d}x}\left(EI\frac{\mathrm{d}^2y}{\mathrm{d}x^2}\right)\right]|_{x=0}^{x=L} \\
&= \int_0^L \left[\frac{\mathrm{d}^2w}{\mathrm{d}x^2}EI\frac{\mathrm{d}^2y}{\mathrm{d}x^2}\right]\,\mathrm{d}x + [wV(y)]|_{x=0}^{x=L} - \left[\frac{\mathrm{d}w}{\mathrm{d}x}\left(EI\frac{\mathrm{d}^2y}{\mathrm{d}x^2}\right)\right]|_{x=0}^{x=L}
\end{aligned}
\tag{56}
$$

Weighted residual statement to Weak statement



Plugging (55) in (56) yields,

$$
\begin{aligned}
0 \quad &= \int_0^L w\left(\frac{\mathrm{d}^2}{\mathrm{d}x^2}\left(EI\frac{\mathrm{d}^2 y}{\mathrm{d}x^2}\right) - q\right)\,\mathrm{d}x - \frac{\mathrm{d}w}{\mathrm{d}x}(\bar{M} - M(y))|_{x=L} + w(\bar{V} - V(y))|_{x=L} \\
&= \left\{\int_0^L \left[\frac{\mathrm{d}^2 w}{\mathrm{d}x^2} EI \frac{\mathrm{d}^2 y}{\mathrm{d}x^2} - wq\right]\,\mathrm{d}x + \left[wV(y)] - \frac{\mathrm{d}w}{\mathrm{d}x}M(y)\right]|_{x=0}^{x=L}\right\} \\
&\quad - \frac{\mathrm{d}w}{\mathrm{d}x}(\bar{M} - M(y))|_{x=L} + w(\bar{V} - V(y))|_{x=L} \\
&= \int_0^L \left[\frac{\mathrm{d}^2 w}{\mathrm{d}x^2} EI \frac{\mathrm{d}^2 y}{\mathrm{d}x^2} - wq\right]\,\mathrm{d}x \\
&\quad + \left\{wV(y) - \frac{\mathrm{d}w}{\mathrm{d}x}M(y) - \frac{\mathrm{d}w}{\mathrm{d}x}(\bar{M} - M(y)) + w(\bar{V} - V(y))\right\}_{x=L} \\
&\quad - \left\{wV(y) - \frac{\mathrm{d}w}{\mathrm{d}x}M(y)\right\}_{x=0}
\end{aligned}
\tag{57}
$$

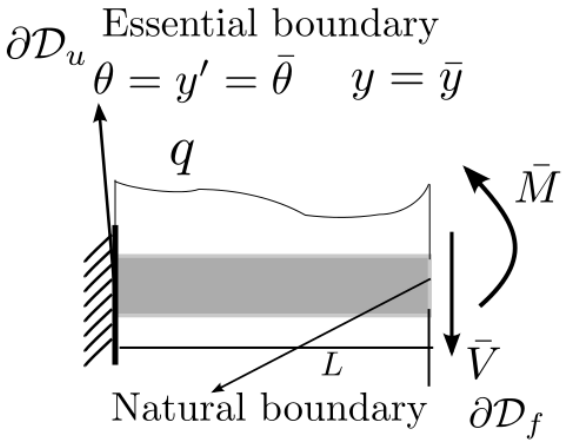
Weighted residual statement to Weak statement

This equation simplifies to

$$0 = \int_0^L \left[\frac{\mathrm{d}^2 w}{\mathrm{d}x^2} EI \frac{\mathrm{d}^2 y}{\mathrm{d}x^2} - wq\right] \mathrm{d}x + \left\{-\frac{\mathrm{d}w}{\mathrm{d}x}\bar{M} + w\bar{V}\right\}_{x=L} \quad (58a)$$

$$+ \left\{w\left(V(y) - V(y)\right) - \frac{\mathrm{d}w}{\mathrm{d}x}\left(M(y) - M(y)\right)\right\}_{x=L} \quad (58b)$$

$$- \left\{wV(y) - \frac{\mathrm{d}w}{\mathrm{d}x}M(y)\right\}_{x=0} \quad (58c)$$



Noting that $M(y) = EI\frac{\mathrm{d}^2 y}{\mathrm{d}x^2}$ and $V(y) = \frac{\mathrm{d}}{\mathrm{d}x}\left(EI\frac{\mathrm{d}^2 y}{\mathrm{d}x^2}\right)$ (second and third order derivatives in x):

- What is the maximum derivative order for y in the interior of the beam (58a)? 2.
- What is the maximum derivative order for w in the interior of the beam (58a)? 2.
- Are the remaining terms at $x = L$ (natural boundary) identically zero (58b)? Yes.
- What is the maximum derivative order for y at $x = 0$ (Essential boundary) (58c)? 3.
- What is the maximum derivative order for w at $x = 0$ (Essential boundary) (58c)? 1.

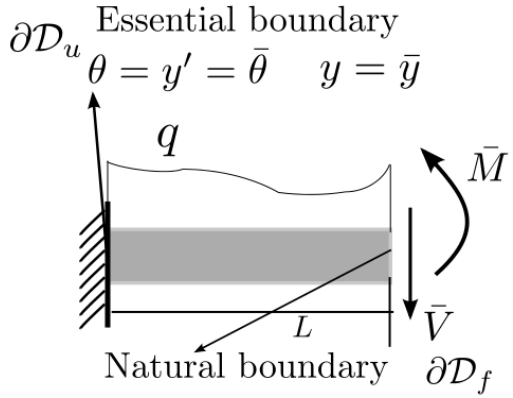
Summary

For the differential equation of order $M = 4$ ($m = 2$) we have been able to equally distribute differential orders between trial function (y) and weight function (w) (order $= m = 2$). The only term that violates this is the essential boundary where y has higher order derivative. We fix this problem by requiring w and $\frac{\mathrm{d}w}{\mathrm{d}x}$ to be identically zero at $x = 0$.

Essential boundary condition

We mentioned that the essential boundary condition is strongly enforced (That is, it is an "essential" condition). The essential conditions (54) require,

$$\mathcal{R}_u = \begin{bmatrix} \bar{\theta} - \theta \\ \bar{y} - y \end{bmatrix} = 0 \Rightarrow \quad \boxed{\left\{ \begin{array}{c} \frac{\mathrm{d}y}{\mathrm{d}x} = \bar{\theta} \\ y = \bar{y} \end{array} \right\}, \text{ at } x = 0 \ (\partial\mathcal{D}_u)} \qquad (59)$$



We discussed that to annihilate the high order derivatives of y in (58c):

$$-\left\{ wV(y) - \frac{\mathrm{d}w}{\mathrm{d}x} M(y) \right\}_{x=0}$$

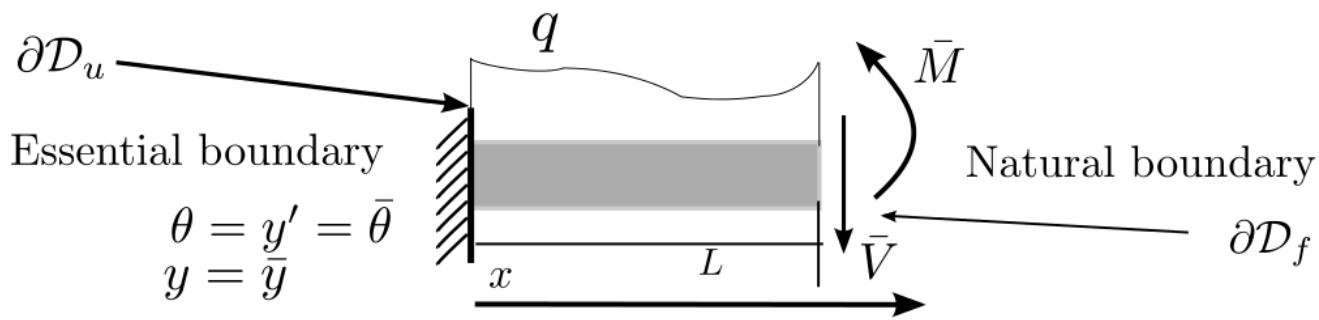
we set the corresponding weight functions identically zero:

$$\left\{ \begin{array}{c} \frac{\mathrm{d}w}{\mathrm{d}x} = 0 \\ w = 0 \end{array} \right\}, \quad \text{at } x = 0 \ (\partial\mathcal{D}_u) \qquad (60)$$

Summary

1. Trial, y, (solution) functions exactly satisfy all essential boundary conditions.
2. Weight, w, functions exactly satisfy the homogeneous essential boundary conditions.
3. If both conditions are satisfied we can form a *weak statement* that requires *only half* the highest derivative order. In fact, this enlarged space of functions is the same as the space of the original balance law.

Weighted Residual Statement (WRS)



The weighted residual for the Euler Bernoulli problem and the boundary conditions in the figure are:

$$\text{Find } y \in \mathcal{V}^{\text{WRS}} = \{u \in C^4(\mathcal{D}) \mid u(0) = \bar{y},\ \frac{\mathrm{d}u}{\mathrm{d}x}(0) = \bar{\theta}\},\ \text{such that,} \tag{61a}$$

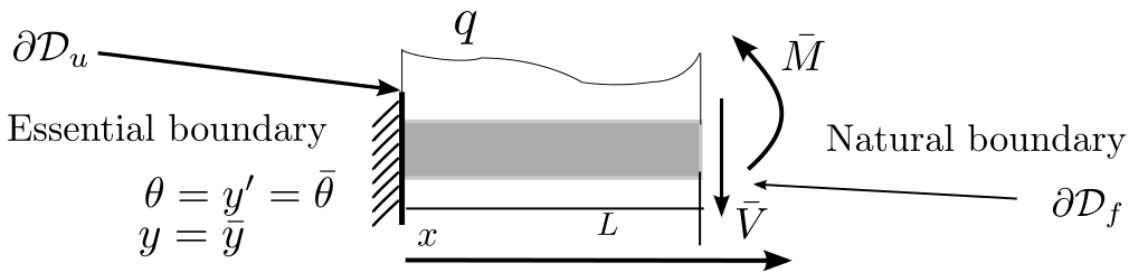
$$\forall w \in \mathcal{W}^{\text{WRS}} = C^1(\mathcal{D}) \text{ no need to enforce the homogeneous essential BCs for WRS} \tag{61b}$$

$$0 = \int_{\mathcal{D}} w\mathcal{R}_i(y)\,\mathrm{d}v + \int_{\partial\mathcal{D}_f} \mathbf{w}_f\mathcal{R}_f(y)\,\mathrm{ds}$$

$$= \int_0^L w\left(\frac{\mathrm{d}^2}{\mathrm{d}x^2}\left(EI\frac{\mathrm{d}^2 y}{\mathrm{d}x^2}\right) - q\right)\mathrm{d}x - \frac{\mathrm{d}w}{\mathrm{d}x}(\bar{M} - M(y))|_{x=L} + w(\bar{V} - V(y))|_{x=L} \tag{61c}$$

So in this second version of the weighted residual statement, we no longer enforce essential boundary conditions weakly. The typical practice, like here, is to enforce the differential equation and natural boundary conditions weakly and the essential boundary conditions strongly.

Weak Statement (WS)



The weak statement for the Euler Bernoulli problem and the BCs in the figure are:

$$\text{Find } y \in \mathcal{V} = \{u \in C^2(\mathcal{D}) \mid u(0) = \bar{y},\ \frac{\mathrm{d}u}{\mathrm{d}x}(0) = \bar{\theta}\},\ \text{such that,} \tag{62a}$$

$$\forall w \in \mathcal{W} = \{u \in C^2(\mathcal{D}) \mid u(0) = 0,\ \frac{\mathrm{d}u}{\mathrm{d}x}(0) = 0\} \tag{62b}$$

$$0 = \int_0^L \left[\frac{\mathrm{d}^2 w}{\mathrm{d}x^2} EI \frac{\mathrm{d}^2 y}{\mathrm{d}x^2} - wq\right] \mathrm{d}x + \left\{-\frac{\mathrm{d}w}{\mathrm{d}x}\bar{M} + w\bar{V}\right\}_{x=L} \tag{62c}$$

Summary

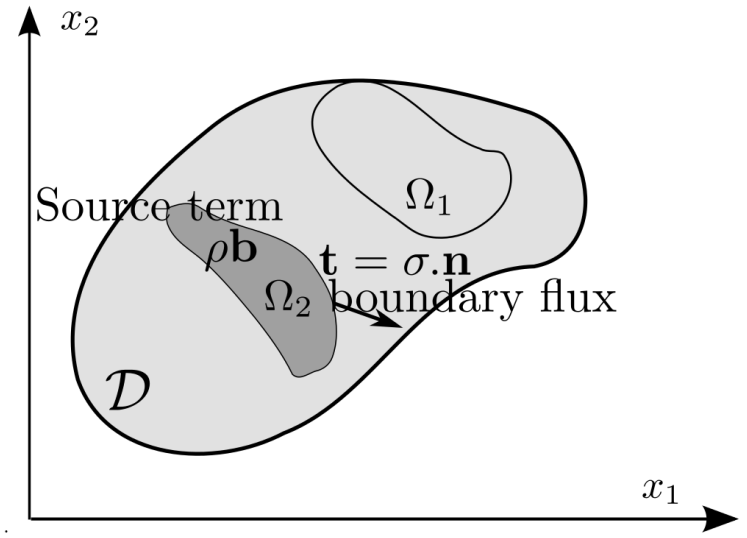
- Both $\mathcal{V}$ and $\mathcal{W}$ have the same regularity ($C^m(\mathcal{D})$): $m = M/2$, $M = 4$ is the order of the differential equation.
- The less demanding regularity conditions for the solution compared to the weighted residual statement ($C^M(\mathcal{D}) \rightarrow C^m(\mathcal{D})$) takes us to the same function space needed for the balance law (balance of linear and angular momentum for Euler Bernoulli beam.
- Both $\mathcal{V}$ and $\mathcal{W}$ exactly enforce the essential boundary conditions, with the difference that $\mathcal{W}$ satisfies the homogeneous version.

Elastostatics: Balance law and Strong Form

Balance law (balance of linear momentum) for solid mechanics reads as

$$\forall \Omega \subset \mathcal{D} : \int_{\partial\Omega} \mathbf{t}\, \mathrm{ds} + \int_{\Omega} \rho \mathbf{b}\, \mathrm{dv} = 0 \qquad (63)$$

$$\text{that is} \quad \int_{\partial\Omega} \sigma.\mathrm{d}\mathbf{s} + \int_{\Omega} \rho \mathbf{b}\, \mathrm{dv} = 0$$



The strong form is

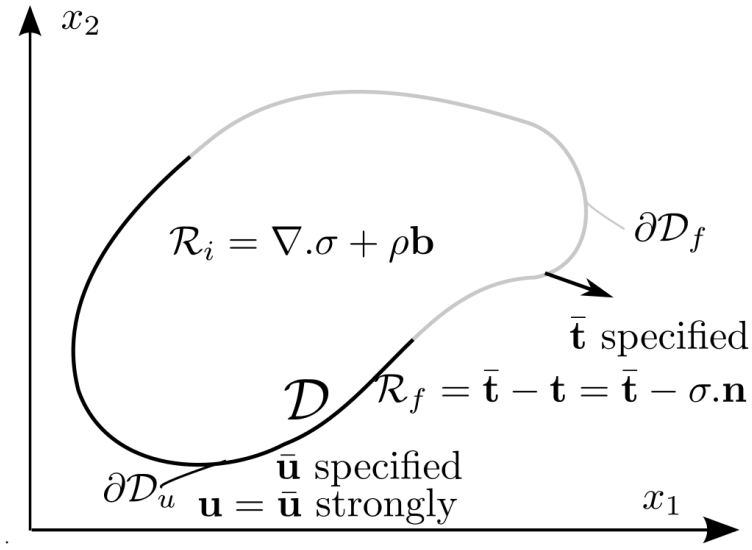
$$\nabla.\sigma + \rho\mathbf{b} = \mathbf{0} \quad \Rightarrow \quad \sigma_{ij,j} + \rho b_i = 0 \qquad (64)$$

Using the constitutive equation $\sigma = \mathbf{C}\epsilon$ and kinematic compatibility $\epsilon = \frac{1}{2}\{\nabla\mathbf{u} + (\nabla\mathbf{u})^{\mathrm{T}}\}$, the closed form of the strong form is,

$$C_{ijkl}\ u_{k,lj} + \rho b_i = 0 \qquad (65)$$

Elastostatics: Weighted Residual Statement

Again to form the weighted residual statement, we take the common approach and strongly enforce the essential boundary conditions, while weakly enforcing the natural boundary conditions and the partial differential equation (strong form)



The Weighted Residual Statement reads as,

$$\text{Find } \mathbf{u} \in \mathcal{V}^{\text{WRS}} = \{\mathbf{v} \in C^2(\mathcal{D}) \mid \forall \mathbf{x} \in \partial\mathcal{D}_u \ \mathbf{v}(\mathbf{x}) = \bar{\mathbf{u}}\}, \text{ such that,} \tag{66a}$$

$$\forall w \in \mathcal{W}^{\text{WRS}} = C^0(\mathcal{D}) \text{ no need to enforce the homogeneous essential BCs for WRS} \tag{66b}$$

$$0 = \int_{\mathcal{D}} \mathbf{w}.(\underbrace{\nabla.\sigma}_{C_{ijkl}\, u_{k,lj}} + \rho\mathbf{b})\, \mathrm{dv} + \int_{\partial\mathcal{D}_f} \mathbf{w}.(\bar{\mathbf{t}} - \mathbf{t})\, \mathrm{ds} \tag{66c}$$

Elastostatics: Toward a weak Statement

The weight function vector $\mathbf{w}$ has zero derivatives and trial (solution) function $\mathbf{u}$ has two derivatives. As before, we want to transfer the derivative from $\mathbf{u}$ to $\mathbf{w}$. We want to show,

$$\int_{\mathcal{D}} \mathbf{w}.(\nabla.\sigma(\mathbf{u}))\,\mathrm{dv} = -\int_{\mathcal{D}} \epsilon(w) : \sigma(\mathbf{u})\,\mathrm{dv} + \int_{\partial\mathcal{D}} \mathbf{w}.\mathbf{t}(\mathbf{u})\,\mathrm{ds}$$

where $\mathbf{t}$ is the traction vector and $\epsilon(w) = \frac{1}{2}\left(\nabla\mathbf{w} + (\nabla\mathbf{w})^{\mathrm{T}}\right)$ is the weight strain field.
We observe,

$$\mathbf{w}.(\nabla.\sigma) = w_i\sigma_{ij,j} = \frac{\partial w_i\sigma_{ij}}{\partial x_j} - w_{i,j}\sigma_{ij} = \frac{\partial w_i\sigma_{ij}}{\partial x_j} - \frac{w_{i,j}+w_{j,i}}{2}\sigma_{ij} - \underbrace{\frac{w_{i,j}-w_{j,i}}{2}}_{=z_{ij}}\sigma_{ij} \tag{67}$$

We note that $z_{ji} = \frac{w_{j,i}-w_{i,j}}{2} = -\frac{w_{i,j}-w_{j,i}}{2} = -z_{ij}$ and $\sigma_{ji} = \sigma_{ij}$ (symmetry of the stress tensor). Then we employ the asymmetry and symmetry of $\mathbf{z}$ and σ to show that:

$$z_{ji}w_{ji} \underbrace{=}_{\text{interchange of dummy indices}} z_{ji}w_{ji} = -(z_{ij})w_{ij} \;\Rightarrow\; 2z_{ij}w_{ij} = 0 \;\Rightarrow\; z_{ij}w_{ij} = 0 \tag{68}$$

also we note that,

$$\frac{w_{i,j}+w_{j,i}}{2}\sigma ij = \epsilon(\mathbf{w}) : \sigma(\mathbf{u}) \tag{69}$$

Elastostatics: Toward a weak Statement

According to (67), (68), and (69) we have:

$$\mathbf{w}.(\nabla.\sigma(\mathbf{u})) = \frac{\partial w_i \sigma_{ij}}{\partial x_j} - \epsilon(\mathbf{w}) : \sigma \quad \Rightarrow$$

$$\int_{\mathcal{D}} \mathbf{w}.(\nabla.\sigma(\mathbf{u})) = \int_{\mathcal{D}} \left(\frac{\partial w_i \sigma_{ij}}{\partial x_j} \right) \, \mathrm{dv} - \int_{\mathcal{D}} \epsilon(\mathbf{w}) : \sigma(\mathbf{u}) \, \mathrm{dv} \quad \Rightarrow \text{(Divergence theorem)}$$

$$\int_{\mathcal{D}} \mathbf{w}.(\nabla.\sigma(\mathbf{u})) = \int_{\partial\mathcal{D}} w_i \underbrace{(\sigma_{ij} n_j)}_{\mathbf{t}_i(\mathbf{u})} \, \mathrm{ds} - \int_{\mathcal{D}} \epsilon(\mathbf{w}) : \sigma \, \mathrm{dv}$$

$$= \int_{\partial\mathcal{D}} \mathbf{w}.\mathbf{t}(\mathbf{u}) \, \mathrm{ds} - \int_{\mathcal{D}} \epsilon(\mathbf{w}) : \sigma(\mathbf{u}) \, \mathrm{dv}$$

which completes the proof

$$\boxed{\int_{\mathcal{D}} \mathbf{w}.(\nabla.\sigma(\mathbf{u})) \, \mathrm{dv} = - \int_{\mathcal{D}} \epsilon(\mathbf{w}) : \sigma(\mathbf{u}) \, \mathrm{dv} + \int_{\partial\mathcal{D}} \mathbf{w}.\mathbf{t}(\mathbf{u}) \, \mathrm{ds}} \tag{70}$$

Elastostatics: Toward a weak Statement

We plug (70) in (66c) to obtain:

$$0 = \int_{\mathcal{D}} (\mathbf{w}.(\nabla.\sigma(\mathbf{u})) + \rho\mathbf{b}) \ \mathrm{dv} + \int_{\partial\mathcal{D}_f} \mathbf{w}.(\bar{\mathbf{t}} - \mathbf{t}) \ \mathrm{ds} \quad \Rightarrow$$

$$0 = \left\{ - \int_{\mathcal{D}} \epsilon(\mathbf{w}) : \sigma(\mathbf{u}) \ \mathrm{dv} + \int_{\partial\mathcal{D}} \mathbf{w}.\mathbf{t}(\mathbf{u}) \ \mathrm{ds} \right\} + \int_{\mathcal{D}} \mathbf{w}.\rho\mathbf{b} \ \mathrm{dv} + \int_{\partial\mathcal{D}_f} \mathbf{w}.(\bar{\mathbf{t}} - \mathbf{t}) \ \mathrm{ds} \ \Rightarrow$$

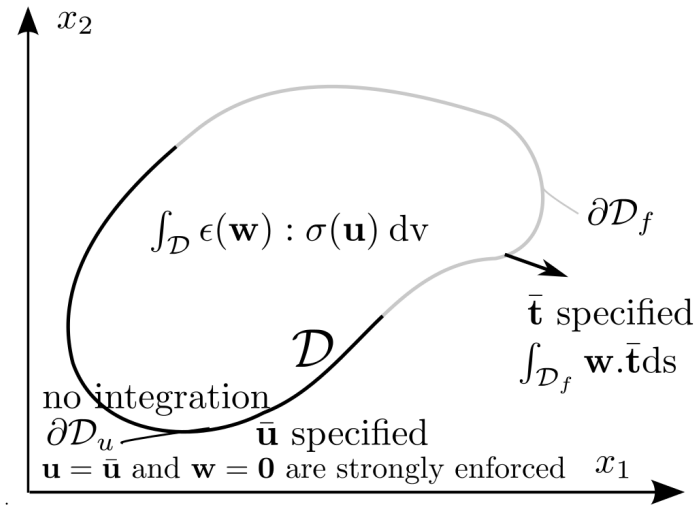
$$\begin{aligned} 0 = & - \int_{\mathcal{D}} \epsilon(\mathbf{w}) : \sigma(\mathbf{u}) \ \mathrm{dv} + \int_{\mathcal{D}} \mathbf{w}.\rho\mathbf{b} \ \mathrm{dv} \\ & + \int_{\partial\mathcal{D}_f} \mathbf{w}.\mathbf{t}(\mathbf{u}) \ \mathrm{ds} + \int_{\partial\mathcal{D}_f} \mathbf{w}.(\bar{\mathbf{t}} - \mathbf{t}) \ \mathrm{ds} \\ & + \int_{\partial\mathcal{D}_u} \mathbf{w}.\mathbf{t}(\mathbf{u}) \ \mathrm{ds} \end{aligned}$$

We used the decomposition $\partial\mathcal{D} = \partial\mathcal{D}_f \cup \partial\mathcal{D}_u, \ \partial\mathcal{D}_f \cap \partial\mathcal{D}_u = \emptyset$. Next We get,

$$\begin{aligned} \int_{\mathcal{D}} \epsilon(\mathbf{w}) : \sigma(\mathbf{u}) \ \mathrm{dv} = & \int_{\mathcal{D}} \mathbf{w}.\rho\mathbf{b} \ \mathrm{dv} + \int_{\partial\mathcal{D}_f} \mathbf{w}.\bar{\mathbf{t}} \ \mathrm{ds} \\ & + \int_{\partial\mathcal{D}_u} \mathbf{w}.\mathbf{t}(\mathbf{u}) \ \mathrm{ds} \end{aligned} \tag{71}$$

Similar to the beam problem we want to eliminate the last integration term on $\partial\mathcal{D}_u$ by enforcing homogeneous boundary conditions for $\mathbf{w}$ on $\partial\mathcal{D}_u$.

Elastostatics: Weak Statement



The weak statement for elastostatics and the boundary conditions are:

$$\text{Find } \mathbf{u} \in \mathcal{V} = \{v \in C^1(\mathcal{D}) \mid \forall \mathbf{x} \in \partial\mathcal{D}_u \ \mathbf{v}(\mathbf{x}) = \bar{\mathbf{u}}(\mathbf{x})\}, \text{ such that,} \tag{72a}$$

$$\forall \mathbf{w} \in \mathcal{W} = \{v \in C^1(\mathcal{D}) \mid \forall \mathbf{x} \in \partial\mathcal{D}_u \ \mathbf{v}(\mathbf{x}) = 0\} \tag{72b}$$

$$\int_{\mathcal{D}} \epsilon(\mathbf{w}) : \sigma(\mathbf{u})\, \mathrm{dv} = \int_{\mathcal{D}} \mathbf{w}.\rho\mathbf{b}\, \mathrm{dv} + \int_{\partial\mathcal{D}_f} \mathbf{w}.\bar{\mathbf{t}}\, \mathrm{ds} \tag{72c}$$

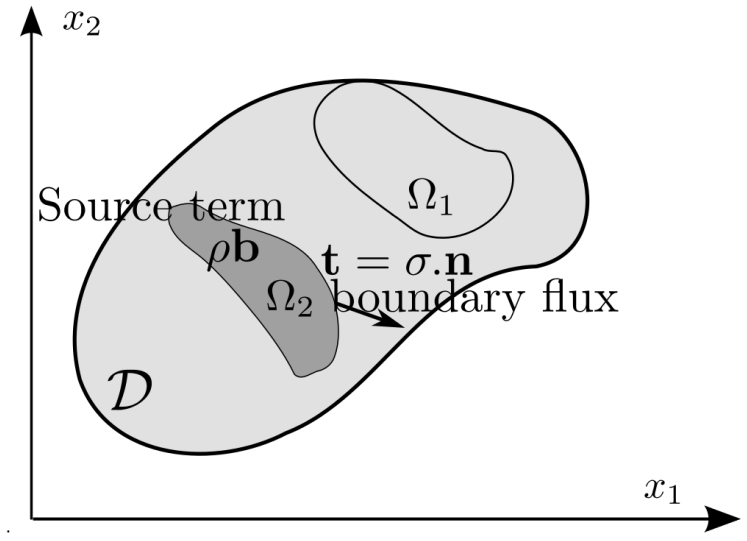
- Both $\mathcal{V}$ and $\mathcal{W}$ have the same regularity ($C^m(\mathcal{D})$): $m = M/2$, $M = 2$ is the order of the differential equation.
- The less demanding regularity conditions for the solution compared to the weighted residual statement ($C^M(\mathcal{D}) \rightarrow C^m(\mathcal{D})$) takes us to the same function space needed for the balance law (highest derivative is for $\sigma(\mathbf{u}) = C_{ijkl}\, u_{k,l}$ is 1).
- Both $\mathcal{V}$ and $\mathcal{W}$ exactly enforce the essential boundary conditions, with the difference that $\mathcal{W}$ satisfies the homogeneous version.

Summary: Elastostatics: Balance law and Strong Form

Balance law (balance of linear momentum) for solid mechanics reads as

$$\forall \Omega \subset \mathcal{D} : \int_{\partial\Omega} \mathbf{t}\, \mathrm{ds} + \int_{\Omega} \rho \mathbf{b}\, \mathrm{dv} = 0$$

$$\text{that is} \quad \int_{\partial\Omega} \sigma.\mathrm{d}\mathbf{s} + \int_{\Omega} \rho \mathbf{b}\, \mathrm{dv} = 0$$



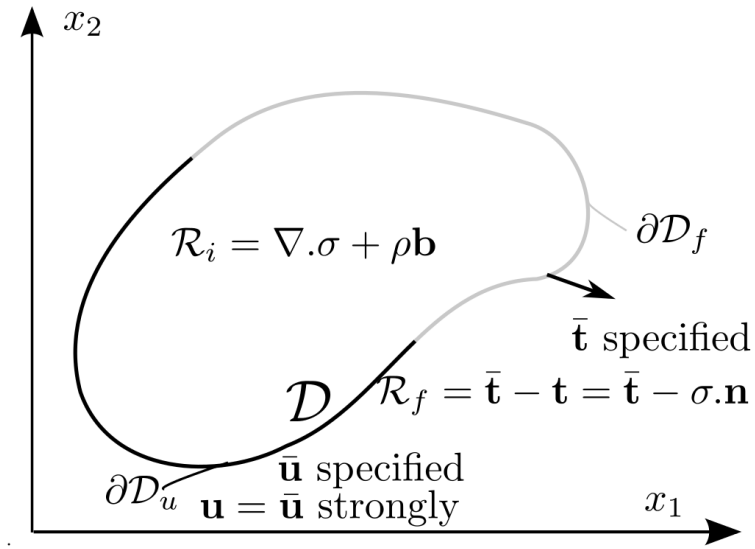
The strong form is

$$\nabla.\sigma + \rho\mathbf{b} = \mathbf{0} \quad \Rightarrow \quad \sigma_{ij,j} + \rho b_i = 0$$

Using the constitutive equation $\sigma = \mathbf{C}\epsilon$ and kinematic compatibility $\epsilon = \frac{1}{2}\{\nabla\mathbf{u} + (\nabla\mathbf{u})^{\mathrm{T}}\}$, the closed form of the strong form is,

$$C_{ijkl}\ u_{k,lj} + \rho b_i = 0$$

Summary: Elastostatics: Weighted Residual Statement



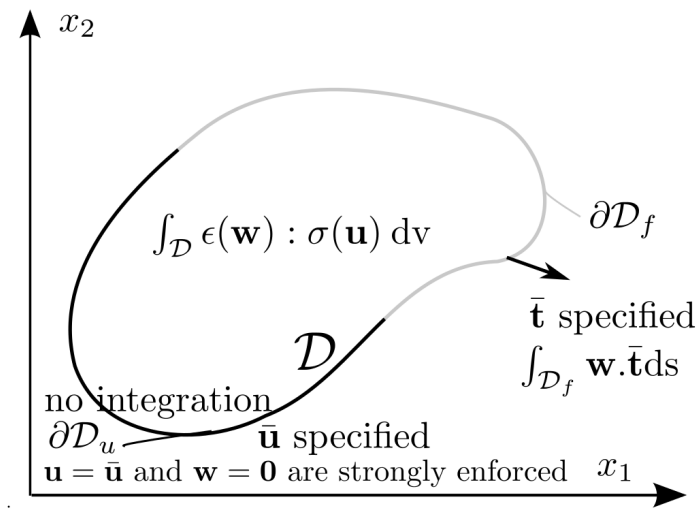
The Weighted Residual Statement reads as,

Find $\mathbf{u} \in \mathcal{V}^{\text{WRS}} = \{\mathbf{v} \in C^2(\mathcal{D}) \mid \forall \mathbf{x} \in \partial\mathcal{D}_u \ \mathbf{v}(\mathbf{x}) = \bar{\mathbf{u}}(\mathbf{x})\}$, such that,

$\forall w \in \mathcal{W}^{\text{WRS}} = C^0(\mathcal{D})$ no need to enforce the homogeneous essential BCs for WRS

$$0 = \int_{\mathcal{D}} \mathbf{w}.(\underbrace{\nabla.\sigma}_{C_{ijkl}\, u_{k,lj}} + \rho\mathbf{b})\, \mathrm{dv} + \int_{\partial\mathcal{D}_f} \mathbf{w}.(\bar{\mathbf{t}} - \mathbf{t})\, \mathrm{ds}$$

Summary: Elastostatics: Weak Statement



The weak statement for elastostatics and the boundary conditions are:

$$\text{Find } \mathbf{u} \in \mathcal{V} = \{v \in H^1(\mathcal{D}) \mid \forall \mathbf{x} \in \partial\mathcal{D}_u \ \mathbf{v}(\mathbf{x}) = \bar{\mathbf{u}}(\mathbf{x})\}, \text{ such that,}$$

$$\forall \mathbf{w} \in \mathcal{W}\{v \in H^1(\mathcal{D}) \mid \forall \mathbf{x} \in \partial\mathcal{D}_u \ \mathbf{v}(\mathbf{x}) = 0\}$$

$$\int_{\mathcal{D}} \epsilon(\mathbf{w}) : \sigma(\mathbf{u})\, \mathrm{dv} = \int_{\mathcal{D}} \mathbf{w}.\rho\mathbf{b}\, \mathrm{dv} + \int_{\partial\mathcal{D}_f} \mathbf{w}.\bar{\mathbf{t}}\, \mathrm{ds}$$

- Strong form requires $C^M(\mathcal{D}) = C^2(\mathcal{D})$ regularity ($M = 2$).
- Weak form requires only half of that regularity: $C^m(\mathcal{D}) = C^1(\mathcal{D})$, for $m = M/2 = 1$.
- As will be explained in , the requirement $C^m(\mathcal{D})$ can further relaxed to $H^m(\mathcal{D})$, which corresponds to global $C^{m-1}(\mathcal{D}) = C^0(\mathcal{D})$ function space for both solution and weights.

Summary Weak Statements

- Elastostatics (72):

$$\text{Find } \mathbf{u} \in \mathcal{V} = \{v \in C^1(\mathcal{D}) \mid \forall \mathbf{x} \in \partial\mathcal{D}_u \ \mathbf{v}(\mathbf{x}) = \bar{\mathbf{u}}(\mathbf{x})\}, \text{ such that,}$$

$$\forall \mathbf{w} \in \mathcal{W} = \{v \in C^1(\mathcal{D}) \mid \forall \mathbf{x} \in \partial\mathcal{D}_u \ \mathbf{v}(\mathbf{x}) = 0\} \ \int_{\mathcal{D}} \epsilon(\mathbf{w}) : \sigma(\mathbf{u}) \, \mathrm{dv} = \int_{\mathcal{D}} \mathbf{w}.\rho\mathbf{b} \, \mathrm{dv} + \int_{\partial\mathcal{D}_f} \mathbf{w}.\bar{\mathbf{t}} \, \mathrm{ds}$$

- 1D bar:

$$\text{Find } u \in \mathcal{V} = \{v \in C^1([0, L]) \mid v(0) = \bar{u}\}, \text{ such that, } \forall w \in \mathcal{W} = \{v \in C^1([0, L]) \mid v(0) = 0\}$$

$$\int_0^L w'(x) E A u'(x) \, \mathrm{d}x = \int_0^L w(x) q(x) \, \mathrm{d}x + w(L)\bar{F} \tag{73}$$

- Bream (62):

$$\text{Find } y \in \mathcal{V} = \{u \in C^2(\mathcal{D}) \mid u(0) = \bar{y}, \ \frac{\mathrm{d}u}{\mathrm{d}x}(0) = \bar{\theta}\}, \text{ such that,}$$

$$\forall w \in \mathcal{W} = \{u \in C^2(\mathcal{D}) \mid u(0) = 0, \ \frac{\mathrm{d}u}{\mathrm{d}x}(0) = 0\}$$

$$\int_0^L \frac{\mathrm{d}^2 w}{\mathrm{d}x^2} E I \frac{\mathrm{d}^2 y}{\mathrm{d}x^2} \, \mathrm{d}x = \int_0^L wq \, \mathrm{d}x + \left\{ \frac{\mathrm{d}w}{\mathrm{d}x} \bar{M} - w\bar{V} \right\}_{x=L}$$

All these equations are self-adjoint and result in the symmetric weight and trial solution differential operator form. For such systems, we can conceptualize the weak form as (note $m = M/2$):

$$\text{Find } \mathbf{u} \in \mathcal{V} = \{v \in C^m(\mathcal{D}) \mid \forall \mathbf{x} \in \partial\mathcal{D}_u \ \mathbf{v}(\mathbf{x}) = \bar{\mathbf{u}}(\mathbf{x})\}, \text{ such that,}$$

$$\forall \mathbf{w} \in \mathcal{W} = \{v \in C^m(\mathcal{D}) \mid \forall \mathbf{x} \in \partial\mathcal{D}_u \ \mathbf{v}(\mathbf{x}) = 0\}$$

$$\int_{\mathcal{D}} L_m^w(\mathbf{w}) L_m(\mathbf{u}) \, \mathrm{dv} = \int_{\mathcal{D}} \mathbf{w}.\mathbf{r} \, \mathrm{dv} + \int_{\partial\mathcal{D}_f} \mathbf{w}^{\mathbf{f}}.\bar{\mathbf{f}} \, \mathrm{ds} \tag{74}$$

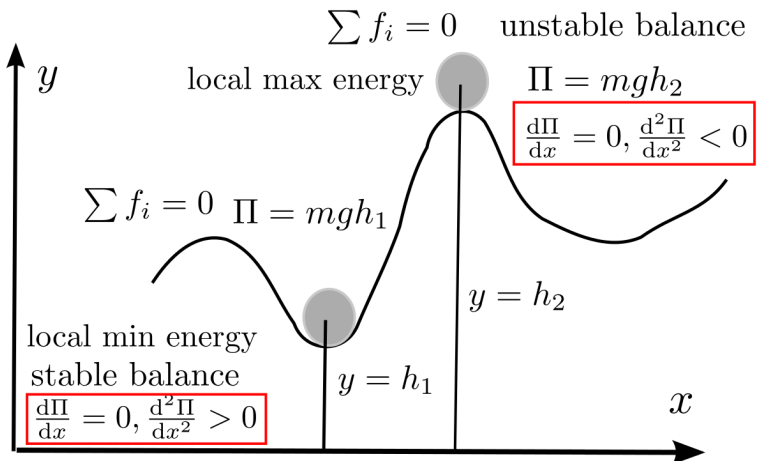
The operators L_m^w and L_m have the following property for self-adjoint problems:

Energy Methods

Energy Methods: Motivation

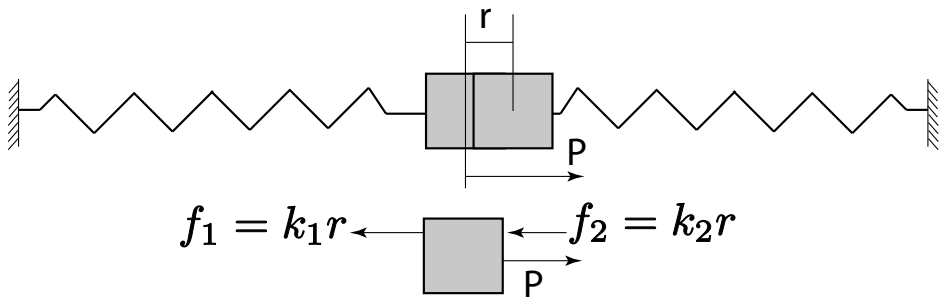
For the gravitational potential shown, the position of the disc is determined by x. The potential energy $\Pi(x) = mgy(x)$ where $y(x)$ is the height function.

- Forces are in balance when $\frac{\mathrm{d}\Pi}{\mathrm{d}x} = 0$. That is, it corresponds to an extremum (minimum or maximum) or the energy function.
- The Balance is stable when the extremum is a local minimum. That is, $\frac{\mathrm{d}\Pi}{\mathrm{d}x} = 0$ and $\frac{\mathrm{d}^2\Pi}{\mathrm{d}x^2} > 0$ (If $\frac{\mathrm{d}^2\Pi}{\mathrm{d}x^2} = 0$ we need higher order derivatives).



- Energy methods seek states for which the energy of the system is minimum. These states are the solutions to the problem.
- Not all problems possess an energy function and balance law has more generality. For example while we cannot define a potential energy corresponding to friction, balance of linear momentum still works for frictional systems.
- In most practical applications, we only consider the extremum condition $\frac{\mathrm{d}\Pi}{\mathrm{d}x} = 0$, as the solutions we obtain are typically a local minimum $\frac{\mathrm{d}^2\Pi}{\mathrm{d}x^2} > 0$.

Energy Methods for discrete systems



- Balance of linear momentum (Newton's law):

$$F = \Sigma f_i = -f_1 - f_2 + P = 0 \Rightarrow$$
$$r = \frac{P}{k_1 + k_2} \tag{76}$$

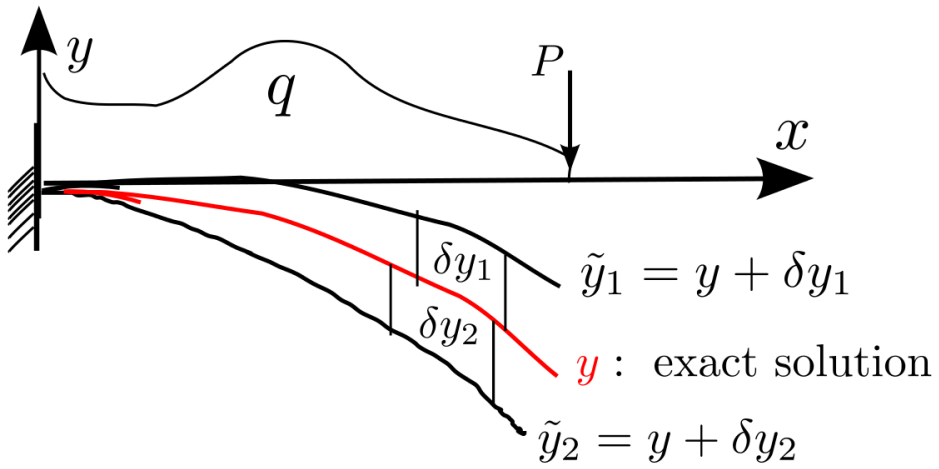
- Energy approach:
  Energy of the system is:

$$\Pi = \frac{1}{2} k_1 r^2 + \frac{1}{2} k_2 r^2 - Pr \tag{77}$$

  Note that $P = -\frac{\mathrm{d}\Pi_P}{\mathrm{d}r}$, where $\Pi_P = -Pr$ is the potential of the applied force P. Then the extremum condition for energy is:

$$\frac{\mathrm{d}\Pi}{\mathrm{d}r} = 0 \quad \Rightarrow k_1 r + k_2 r - P = 0 \quad \Rightarrow \quad r = \frac{P}{k_1 + k_2} \tag{78}$$

Energy Methods for continuum systems



In energy methods, the solution corresponds to a function that minimizes the energy. For example for the beam problem shown, we assume that y is the solution and $\tilde{y}$ is an arbitrary trial function that is related to y by,

$$\tilde{y} = y + \delta y \tag{79}$$

where the function $\delta y(x)$, is the increment (the difference) between the solution y and the trial function $\tilde{y}$. Minimization of the energy for the solution means,

$$\begin{aligned} & y \in \mathcal{V} \text{ is a solution iff,} \\ \forall\, \tilde{y} \in \mathcal{V} : \quad & \Pi(y) \leq \Pi(\tilde{y}) \quad \text{or alternatively} \\ \forall\, \delta y \in \mathcal{W} : \quad & \Pi(y) \leq \Pi(y + \delta y) \end{aligned} \tag{80}$$

The function spaces $\mathcal{V}$ and $\mathcal{W}$ will be discussed next.

Energy Method: Essential Boundaries and Function spaces

As will be shown, the energy for the beam in the figure is,

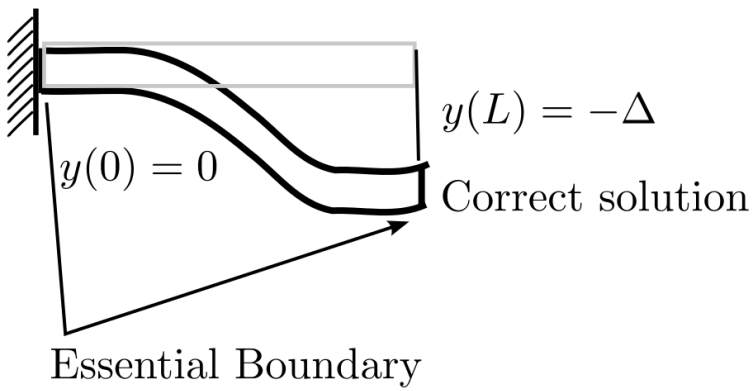
$$\Pi = \int_0^L \frac{1}{2} EI \left(\frac{\mathrm{d}^2 y}{\mathrm{d}x^2} \right)^2 \mathrm{d}x \qquad (81)$$

incorrect solutions that do not satisfy essential boundary conditions.

yet their energy is smaller than that of correct solution

The Essential boundary conditions for this problem are:

$$\begin{aligned} y(0) &= 0 \\ y(L) &= -\Delta \end{aligned} \qquad (82)$$



The requirements for trial functions are:

- Smoothness: Clearly, the first requirement for the trial functions–for which we evaluate the energy–is that the equation (81) can be evaluated. Since the maximum derivative order is $m = 2$, for now we assume the functions should be in $C^2([0, L])$. We will later discuss how this smoothness requirement can be relaxed.
- Essential boundary conditions: If essential boundary conditions are not enforced for the trial functions, we notice that any of the rigid body configurations (gray rectangles) have zero energy, while the correct solution (shown in black) clearly has a positive energy bases on equation (81). Thus, it is essential to restrict the trial function space to those that satisfy essential boundary conditions.

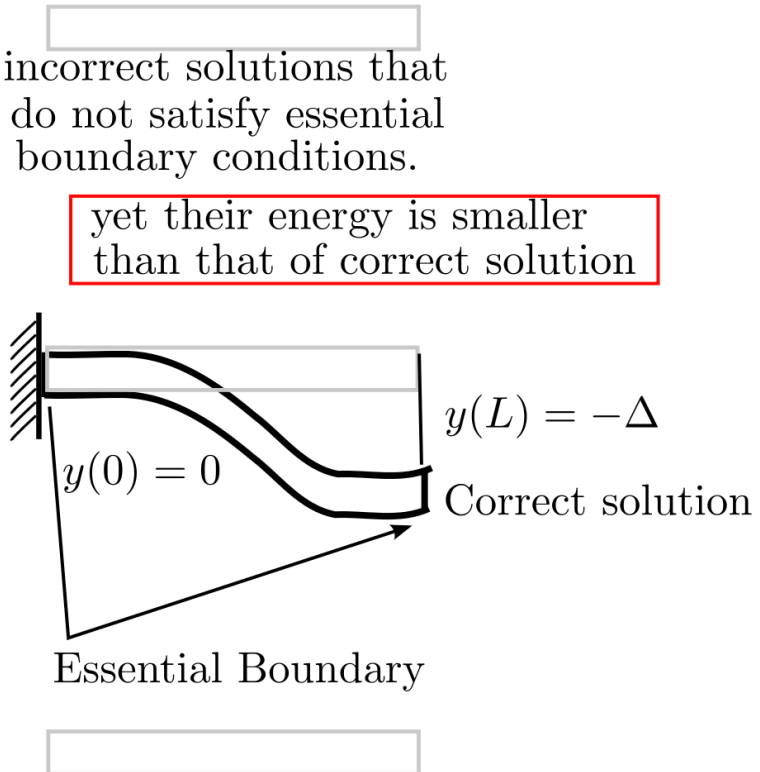
Energy Method: Function spaces

Based on the need to satisfy essential boundary conditions, derivative order in (81), and corresponding essential boundary conditions, the function space for trial functions, $\tilde{y}$ is,

$$\mathcal{V} = \{u \mid u \in C^2([0,\ L]),\ u(0) = 0,\ u(L) = -\Delta\}. \quad (83)$$

For the increments, we notice that all trial functions $\tilde{y}$ (including y itself) satisfy essential boundary conditions from (83). So for an arbitrary increment function δy (*cf.* (79)) we have $\delta y(0) = \tilde{y}(0) - y(0) = 0 - 0 = 0$, and $\delta y(0) = \tilde{y}(L) - y(L) = -\Delta - (-\Delta) = 0$. So the space of increment functions, δy is,

$$\mathcal{W} = \{u \mid u \in C^2([0,\ L]),\ u(0) = 0,\ u(L) = 0\}. \quad (84)$$



Conceptually, compare these spaces with trial and weight functions we obtained for a sample Euler Bernoulli beam problem in (61).

Energy Method for Solid Mechanics

The total energy in solid mechanics is,

$$\Pi = T + V - W = \text{Total energy} \tag{85a}$$

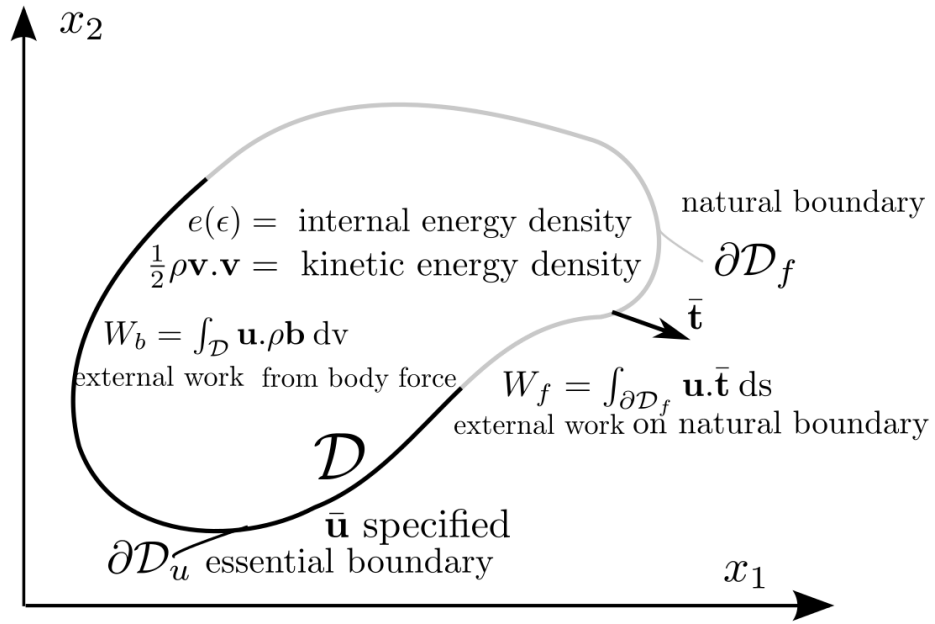
$$T = \int_{\mathcal{D}} \frac{1}{2}\rho \mathbf{v}.\mathbf{v}\, \mathrm{dv} = \text{ Kinetic energy} \tag{85b}$$

$$V = \int_{\mathcal{D}} e(\epsilon)\, \mathrm{dv} = \text{ Internal energy} \tag{85c}$$

$$W = W_b + W_f = \text{ External work} \tag{85d}$$

$$W_b = \int_{\mathcal{D}} \mathbf{u}.\rho\mathbf{b}\, \mathrm{dv} \tag{85e}$$

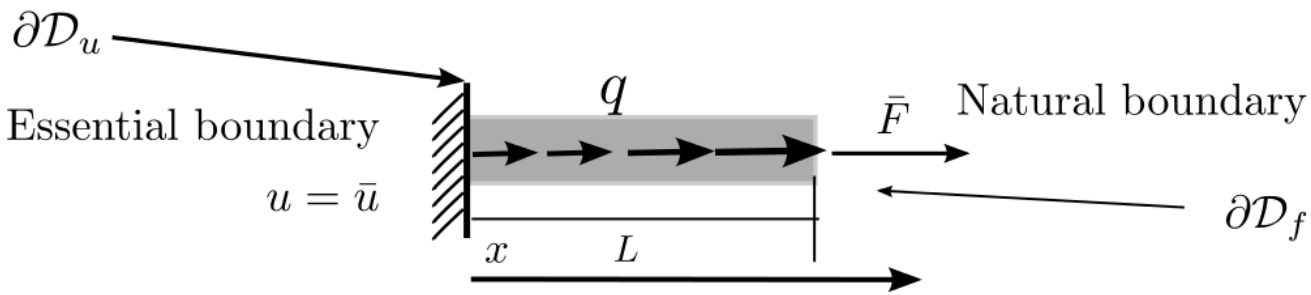
$$W_f = \int_{\partial\mathcal{D}_f} \mathbf{u}.\bar{\mathbf{t}}\, \mathrm{ds} \tag{85f}$$



- For static problems $T = 0$.
- Internal energy density, $e(\epsilon) = \frac{1}{2}\epsilon : \sigma(\epsilon) = \frac{1}{2}C_{ijkl}\epsilon_{ij}\epsilon_{kl}$ for linear solid.
- Natural boundary forces are naturally incorporated into the energy (W_f).
- Essential boundary conditions are incorporated into function space:

$$\mathbf{u} \in \mathcal{V} = \{\mathbf{v} \,|\mathbf{v} \in C^1(\mathcal{D}) : \ \forall \mathbf{x} \in \partial\mathcal{D}_u \ \mathbf{v}(\mathbf{x}) = \bar{\mathbf{u}}(\mathbf{x})\}, \text{ is a solution if}$$
$$\forall \tilde{\mathbf{u}} \in \mathcal{V}, \quad \Pi(\mathbf{u}) \leq \Pi(\tilde{\mathbf{u}}). \tag{86}$$

Energy Method for 1D solid bar (statics)



$$T = 0 \text{ statics} \tag{87a}$$

$$V = \int_0^L \frac{1}{2}\epsilon\sigma \,\mathrm{dv} = \int_0^L \frac{1}{2}\epsilon(E\epsilon)\,(A\mathrm{d}x) = \int_0^L \frac{1}{2}EA\left(\frac{\mathrm{d}u}{\mathrm{d}x}\right)^2 \mathrm{d}x \tag{87b}$$

$$W_b = \int_0^L u(x)q(x)\,\mathrm{d}x \tag{87c}$$

$$W_f = u(L)\bar{F} \tag{87d}$$

where A is the cross section area and $q(x)$ is distributed load. Thus the total energy is,

$$\boxed{\Pi(u) = \int_0^L \frac{1}{2}EA\left(\frac{\mathrm{d}u}{\mathrm{d}x}\right)^2 \mathrm{d}x - \int_0^L u(x)q(x)\,\mathrm{d}x - u(L)\bar{F}} \tag{88}$$

Since $\partial\mathcal{D}_u = \{0\}$ with the essential boundary condition $u(0) = \bar{u}$, the energy statement is,

$$\begin{aligned} &\text{Find } u \in \mathcal{V} = \{v \,|v \in C^1(\mathcal{D}) :\ v(0) = \bar{u}\}, \text{ such that} \\ &\forall \tilde{u} \in \mathcal{V}, \quad \Pi(u) \le \Pi(\tilde{u}). \end{aligned} \tag{89}$$

First variation of a function / extremum condition

Let $f(x)$ be a function from $\mathbb{R} \to \mathbb{R}$ ($\mathbb{R}$ is the real number set). We are interested in finding the increment to the function value due to change in the function argument x_0:

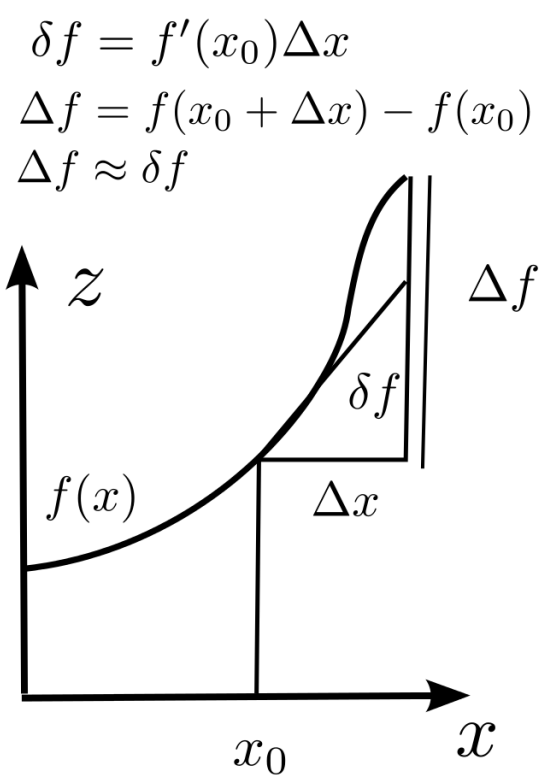
$$x_0 \ \to x_0 + \Delta x \ : \ f(x_0) \ \to ?$$

We adopt the following definitions:

- Total variation: $\Delta f(x_0, \Delta x) = f(x_0 + \Delta x) - f(x_0)$
- First variation: $\delta f(x_0, \Delta x) = \frac{\mathrm{d}f}{\mathrm{d}x}(x_0)\Delta x$

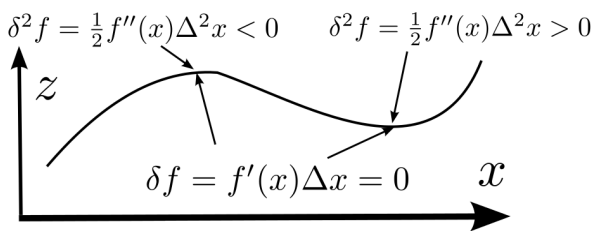
We often drop the arguments x_0 and Δx as shown. For a differentiable function we expect:

$$\Delta f \approx \delta f \text{ for "small" } |\Delta x|$$



Extremum condition (min/max)

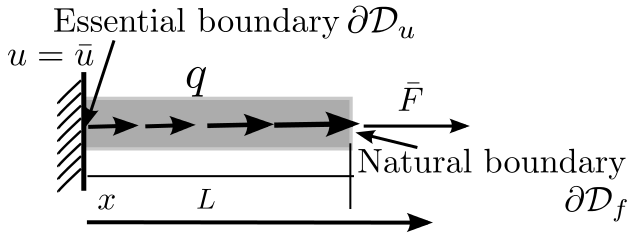
- Necessary condition: $\frac{\mathrm{d}f}{\mathrm{d}x} = 0 \Rightarrow \delta f = 0$ (first variation $= 0$)
- Whether or not the point is a min/max or otherwise is determined by higher derivatives (variations). Some examples are shown on the right.



Functionals / first variation

From (88) we obtained the total energy for the bar,

$$\Pi(u) = \int_0^L \frac{1}{2} EA \left(\frac{\mathrm{d}u}{\mathrm{d}x}\right)^2 \mathrm{d}x - \int_0^L u(x)q(x)\,\mathrm{d}x - u(L)\bar{F}$$



for a trial function u. We call objects such as Π a functional, that is a function whose argument itself is a function.
The change of the functional argument,

$$\tilde{y} = y + \delta y \tag{90}$$

is similar to the change of $x_0 \to x_0 + \Delta x$ for a function. Similar to that case, we want to obtain the first variation of the functional Π denoted by $\delta\Pi$ as the first order approximation for the total variation $\Delta\Pi$:

$$\Delta\Pi(y, \delta y) = \Pi(y + \delta y) - \Pi(y) \tag{91a}$$

$$\delta\Pi(y, \delta y) = \text{first order approximation (w.r.t. } \delta y) \text{ to } \Delta\Pi \tag{91b}$$

Functionals Optimality condition

- For a function $f : \mathbb{R} \to \mathbb{R}$ we observed that a necessary condition for optimality of f at x_0 was, $\delta f = \frac{\mathrm{d}f}{\mathrm{d}x}(x_0)\Delta x = 0$.
- What do we expect a necessary optimality condition for a functional Π be?
  a necessary extremum condition for Π at y is

$$\delta\Pi = 0, \text{ where } \delta\Pi \text{ is a shorthand for } \delta\Pi(y, \delta y) \tag{92}$$

- How to evaluate $\delta\Pi$?

$$\Pi = \Pi(y, \frac{\mathrm{d}y}{\mathrm{d}x}, \ldots, \frac{\mathrm{d}^n y}{\mathrm{d}x^n}) \quad \Rightarrow \quad \delta\Pi = \frac{\partial\Pi}{\partial y}\delta y + \frac{\partial\Pi}{\partial \frac{\mathrm{d}y}{\mathrm{d}x}}\delta(\frac{\mathrm{d}y}{\mathrm{d}x}) + \cdots + \frac{\partial^n\Pi}{\partial \frac{\mathrm{d}^n y}{\mathrm{d}x^n}^n}\delta(\frac{\mathrm{d}^n y}{\mathrm{d}x^n}) \tag{93}$$

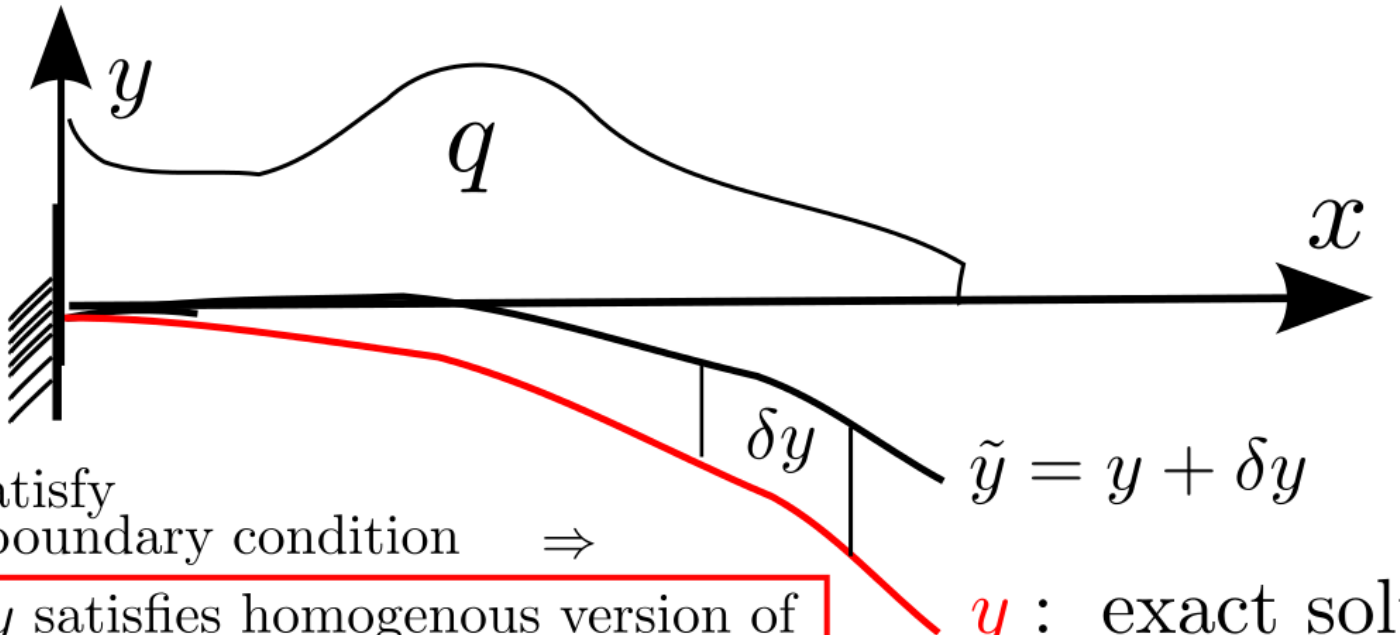
Note the similarity to the corresponding conditions for a function $f(x)$: $\delta f = \frac{\mathrm{d}f}{\mathrm{d}x}\Delta x = 0$.

Having a $\delta y \Rightarrow \tilde{y} = y + \delta y \Rightarrow \frac{\mathrm{d}^n\tilde{y}}{\mathrm{d}x^n} = \frac{\mathrm{d}^n y}{\mathrm{d}x^n} + \frac{\mathrm{d}^n\delta y}{\mathrm{d}x^n} \Rightarrow$ $\boxed{\delta(\frac{\mathrm{d}^n y}{\mathrm{d}x^n}) = (\frac{\mathrm{d}^n\delta y}{\mathrm{d}x^n}) := \delta y^{(n)}}$ (94)

Thus, noting that $y^{(n)} := \frac{\mathrm{d}^n y}{\mathrm{d}x^n}$, (93) can be rewritten as,

$$\Pi = \Pi(y, y', \ldots, y^{(n)}) \quad \Rightarrow \quad \boxed{\delta\Pi = \frac{\partial\Pi}{\partial y}\delta y + \frac{\partial\Pi}{\partial y'}\delta y' + \cdots + \frac{\partial^n\Pi}{\partial y^{(n)^n}}\delta y^{(n)}.} \tag{95}$$

Summary: Functionals and function spaces



y and $\tilde{y}$ satisfy essential boundary condition $\Rightarrow$

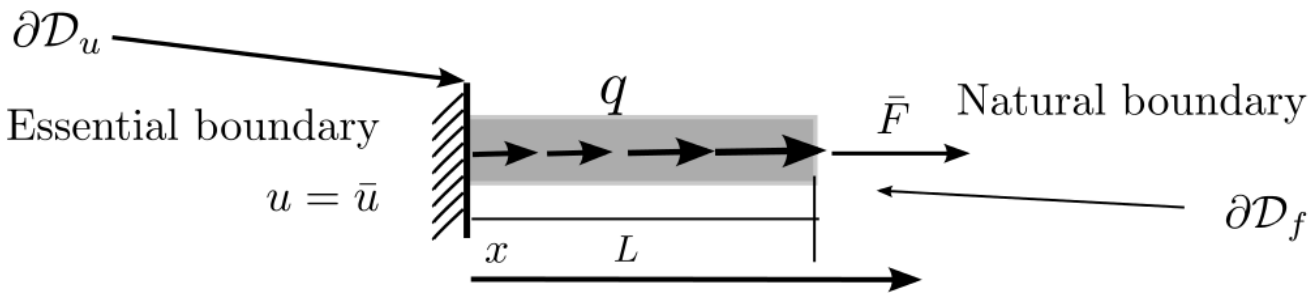
$\delta y = \tilde{y} - y$ satisfies homogenous version of essential boundary condition

1. A necessary condition for the optimality of functional Π at y is (*cf.* (95)),

$$\Pi = \Pi(y, y', \ldots, y^{(n)}) \quad \Rightarrow \quad \boxed{\delta\Pi = \frac{\partial \Pi}{\partial y}\delta y + \frac{\partial \Pi}{\partial y'}\delta y' + \cdots + \frac{\partial^n \Pi}{\partial {y^{(n)}}^n}\delta y^{(n)}.} \tag{96}$$

2. Solution y satisfies all essential boundary conditions.
3. Increment δy satisfies the homogeneous version of all essential boundary conditions.

Example: 1D solid bar



- Functional: We determined the total energy of the solid bar to be (*cf.* (88)),

$$\Pi(u) = \int_0^L \frac{1}{2} EA \left(\frac{\mathrm{d}u}{\mathrm{d}x}\right)^2 \mathrm{d}x - \int_0^L u(x)q(x)\,\mathrm{d}x - u(L)\bar{F}$$

The optimality condition is,

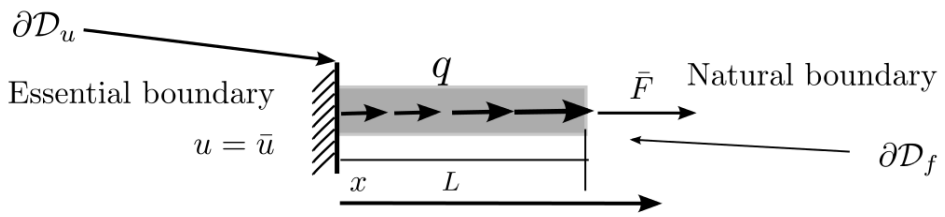
$$\begin{aligned}
\Pi = \Pi(u, u') \Rightarrow \delta\Pi &= \delta \int_0^L \frac{1}{2} EA \left(\frac{\mathrm{d}u}{\mathrm{d}x}\right)^2 \mathrm{d}x - \delta \int_0^L u(x)q(x)\,\mathrm{d}x - \delta u(L)\bar{F} \\
&= \int_0^L \frac{1}{2} EA\delta \left(\frac{\mathrm{d}u}{\mathrm{d}x}\right)^2 \mathrm{d}x - \int_0^L \delta u(x)q(x)\,\mathrm{d}x - \delta u(L)\bar{F} \\
&= \int_0^L \delta u' EAu'\,\mathrm{d}x - \int_0^L \delta u(x)q(x)\,\mathrm{d}x - \delta u(L)\bar{F} \qquad (97)
\end{aligned}$$

Note that $\delta\left(\frac{\partial u}{\partial x}\right)^2 = \frac{\mathrm{d}\left(\frac{\mathrm{d}u}{\mathrm{d}x}\right)^2}{\mathrm{d}\left(\frac{\mathrm{d}u}{\mathrm{d}x}\right)}\delta\left(\frac{\mathrm{d}u}{\mathrm{d}x}\right) = \left(2\frac{\mathrm{d}u}{\mathrm{d}x}\right)\frac{\mathrm{d}\delta u}{\mathrm{d}x} = 2\delta u' u'$.

- Essential boundary conditions are:

$$u(0) = \bar{u}$$

Example: 1D solid bar



Based on the condition $\delta\Pi = 0$, u the solution from the energy method is,

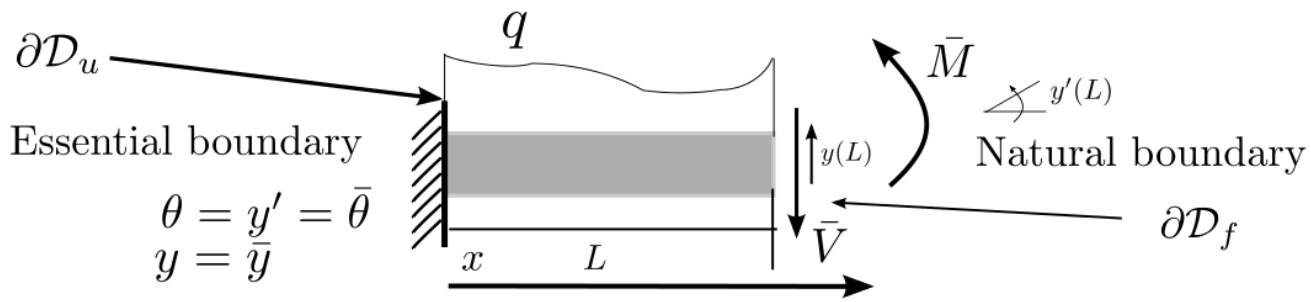
$$\text{Find } u \in \mathcal{V} = \{v \in C^1([0,L]) \mid v(0) = \bar{u}\}, \text{ such that,}$$
$$\forall \delta u \in \mathcal{W} = \{v \in C^1([0,L]) \mid v(0) = 0\}$$
$$\delta\Pi = \int_0^L \delta u'(x) EA u'(x)\, \mathrm{d}x - \int_0^L \delta u(x) q(x)\, \mathrm{d}x - \delta u(L)\bar{F} = 0 \tag{98}$$

- What would be the weak form for this problem based on the balance law/Differential Equation approach (*cf.* (73))?

$$\text{Find } u \in \mathcal{V} = \{v \in C^1([0,L]) \mid v(0) = \bar{u}\}, \text{ such that,}$$
$$\forall w \in \mathcal{W} = \{v \in C^1([0,L]) \mid v(0) = 0\}$$
$$\delta\Pi = \int_0^L w'(x) EA u'(x)\, \mathrm{d}x - \int_0^L w(x) q(x)\, \mathrm{d}x - w(L)\bar{F} = 0 \tag{99}$$

- What are the differences between this weak statement and the optimality condition obtained from energy method?

Example: Euler Bernoulli beam



We determined the internal energy of the beam to be (*cf.* (85c)),

$$V = \int_{\mathcal{D}} \frac{1}{2}\epsilon\sigma \,\mathrm{dv} = \int_0^L \left(\int_A \frac{1}{2}\epsilon^2 E \,\mathrm{d}A \right) \mathrm{d}x = \int_0^L \left(\int_A \frac{1}{2}(\frac{\mathrm{d}^2 y}{\mathrm{d}x^2} z)^2 E \,\mathrm{d}A \right) \mathrm{d}x$$
$$= \int_0^L \frac{1}{2} E (\frac{\mathrm{d}^2 y}{\mathrm{d}x^2})^2 (\underbrace{\int_A z^2 \,\mathrm{d}A}_{I}) \,\mathrm{d}x \Rightarrow$$

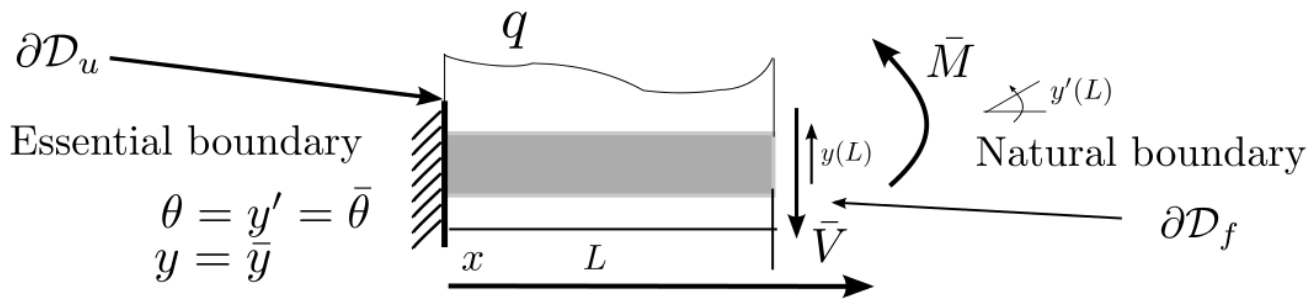
$$\boxed{V = \int_0^L \frac{1}{2} EI (\frac{\mathrm{d}^2 y}{\mathrm{d}x^2})^2 \,\mathrm{d}x} \tag{100}$$

The external works are:

$$W_b = \int_0^L y(x) q(x) \,\mathrm{d}x \tag{101a}$$

$$W_f = y(L)(-\bar{V}) + \frac{\mathrm{d}y}{\mathrm{d}x}(L)\bar{M} = -y(L)\bar{V} + \frac{\mathrm{d}y}{\mathrm{d}x}(L)\bar{M} \tag{101b}$$

Example: Euler Bernoulli beam



- Functional: From equations (85), (100), (101) and the fact that $T = 0$ (statics) the total energy of the beam is,

$$\Pi(y) = \int_0^L \frac{1}{2} EI(\frac{\mathrm{d}^2 y}{\mathrm{d}x^2})^2 \,\mathrm{d}x - \int_0^L y(x)q(x)\,\mathrm{d}x + y(L)\bar{V} - \frac{\mathrm{d}y}{\mathrm{d}x}(L)\bar{M} \tag{102}$$

The optimality for $\Pi = \Pi(y, y', y'')$ is,

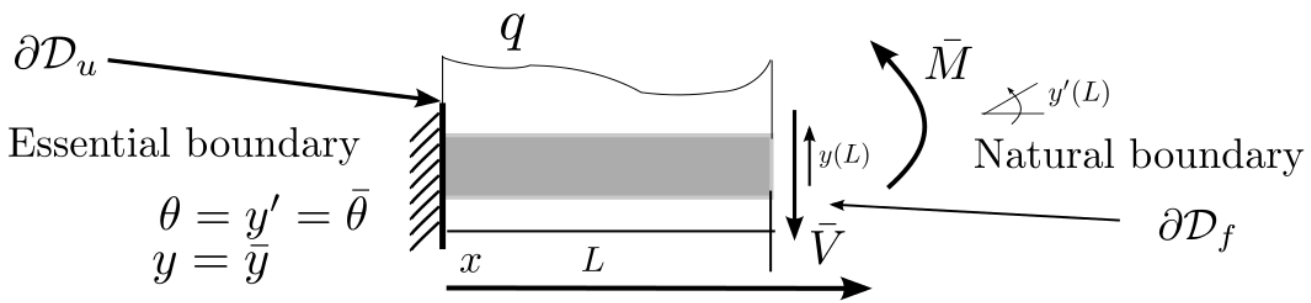
$$\delta\Pi = \delta \int_0^L \frac{1}{2} EI(\frac{\mathrm{d}^2 y}{\mathrm{d}x^2})^2 \,\mathrm{d}x - \delta \int_0^L y(x)q(x)\,\mathrm{d}x + \delta y(L)\bar{V} - \delta(\frac{\mathrm{d}y}{\mathrm{d}x}(L))\bar{M} \tag{103}$$

$$= \int_0^L \frac{1}{2} EI\delta \left(\frac{\mathrm{d}^2 y}{\mathrm{d}x^2}\right)^2 \mathrm{d}x - \int_0^L \delta y(x)q(x)\,\mathrm{d}x + \delta y(L)\bar{V} - \frac{\mathrm{d}\delta y}{\mathrm{d}x}(L)\bar{M}$$

$$= \int_0^L \delta y'' EI y''\,\mathrm{d}x - \int_0^L \delta y q\,\mathrm{d}x + \delta y(L)\bar{V} - \delta y'(L)\bar{M} \tag{104}$$

Note that $\delta\left(\frac{\partial^2 y}{\partial x^2}\right)^2 = \frac{\mathrm{d}\left(\frac{\mathrm{d}^2 y}{\mathrm{d}x^2}\right)^2}{\mathrm{d}\left(\frac{\mathrm{d}^2 y}{\mathrm{d}x^2}\right)}\delta\left(\frac{\mathrm{d}^2 y}{\mathrm{d}x^2}\right) = \left(2\frac{\mathrm{d}^2 y}{\mathrm{d}x^2}\right)\frac{\mathrm{d}^2\delta y}{\mathrm{d}x^2} = 2\delta y'' y''$.

Example: Euler Bernoulli beam



- Functional: From (103) for the solution y we have,

$$\boxed{\delta\Pi = \int_0^L \delta y'' EI y'' \,\mathrm{d}x - \int_0^L \delta y q \,\mathrm{d}x + \delta y(L)\bar{V} - \delta y'(L)\bar{M}} \tag{105}$$

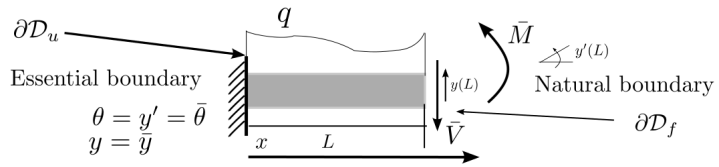
- Essential boundary conditions are:

$$y(0) = \bar{y} \tag{106a}$$

$$\frac{\mathrm{d}y}{\mathrm{d}x}(0) = y'(0) = \theta(0) = \bar{\theta} \tag{106b}$$

Example: Euler Bernoulli beam

From (105) and (106) the solution from energy method is,



$$\text{Find } y \in \mathcal{V} = \{v \in C^2([0,L]) \mid v(0) = \bar{y}, \frac{\mathrm{d}v}{\mathrm{d}x}(0) = \bar{\theta}\}, \text{ such that,}$$

$$\forall \delta y \in \mathcal{W} = \{v \in C^2([0,L]) \mid v(0) = 0, \frac{\mathrm{d}v}{\mathrm{d}x}(0) = 0\}$$

$$\delta\Pi = \int_0^L \delta y''(x) EI y''(x)\,\mathrm{d}x - \int_0^L \delta y(x) q(x)\,\mathrm{d}x + \delta y(L)\bar{V} - \delta y'(L)\bar{M} = 0 \tag{107}$$

- What would be the weak form for this problem based on the balance law/Differential Equation approach (equation (62))?

$$\text{Find } y \in \mathcal{V} = \{v \in C^2([0,L]) \mid v(0) = \bar{u}, \frac{\mathrm{d}v}{\mathrm{d}x}(0) = \bar{\theta}\}, \text{ such that,}$$

$$\forall w \in \mathcal{W} = \{v \in C^2([0,L]) \mid v(0) = 0, \frac{\mathrm{d}v}{\mathrm{d}x}(0) = 0\}$$

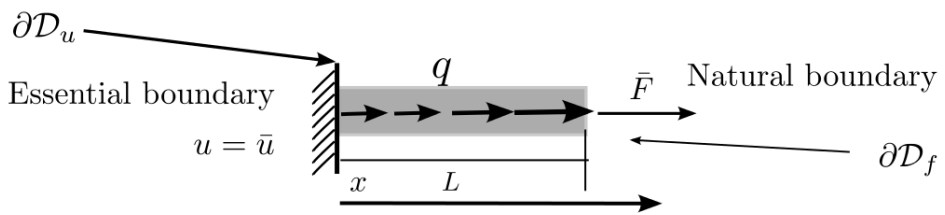
$$\delta\Pi = \int_0^L w''(x) EI u''(x)\,\mathrm{d}x - \int_0^L w(x) q(x)\,\mathrm{d}x + w(L)\bar{V} - w'(L)\bar{M} = 0 \tag{108}$$

- What are the differences between this weak statement and the optimality condition obtained from energy method?

Energy Method vs. Principle of Virtual Work

- Principle of virtual work or virtual displacement in solid mechanics states that if $\mathbf{u}$ is the solution to a boundary value problem, the virtual internal and external works produces by admissible virtual displacements $\delta\mathbf{u}$ are equal.
- Virtual displacements $\delta\mathbf{u}$ refer to displacements that are zero at essential boundary values (so that solution displacement plus virtual displacement $\tilde{\mathbf{u}} = \mathbf{u} + \delta\mathbf{u}$ (*cf.* (79)) as another admissible trial function also satisfies essential boundary conditions).
- Virtual Displacement/Virtual work is basically the equation we obtain by minimizing the energy function $\delta\Pi = 0$.
- Similar principles (virtual temperature for heat flow in solids and virtual velocities for fluid flow) are also directly derived from $\delta\Pi = 0$.
- While principle of virtual work can be obtained from $\delta\Pi = 0$, it is often quite easy to directly write and equate internal and external works for a given problem.

Virtual work: 1D solid bar



Equation (98) can be written as,

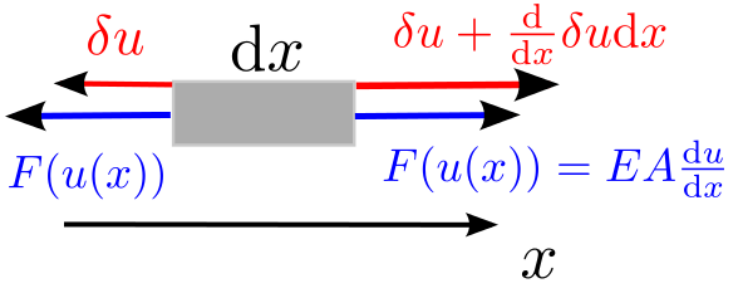
Find $u \in \mathcal{V} = \{v \in C^1([0, L]) \mid v(0) = \bar{u}\}$, such that,

$\forall \delta u \in \mathcal{W} = \{v \in C^1([0, L]) \mid v(0) = 0\}$

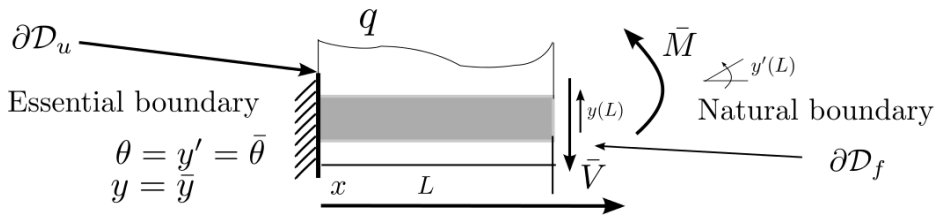
$$\underbrace{\int_0^L \overbrace{\delta u'(x)}^{\frac{\mathrm{d}}{\mathrm{d}x}\delta u} \overbrace{EAu'(x)}^{F(u(x))} \,\mathrm{d}x}_{\text{Virtual Internal Work}} = \underbrace{\int_0^L \delta u(x) q(x) \,\mathrm{d}x + \delta u(L)\bar{F}}_{\text{Virtual External Work}} \tag{109}$$

Note that the internal work differential is:

$$\begin{aligned} \mathrm{d}V &= F(u(x)).(\delta u + \frac{\mathrm{d}}{\mathrm{d}x}\delta u \mathrm{d}x) - F(u(x)).\delta u \\ &= \frac{\mathrm{d}\delta u}{\mathrm{d}x}.F(u(x))\mathrm{d}x \end{aligned} \tag{110}$$



Virtual work for Euler Bernoulli beam



Equation (107) can be written as,

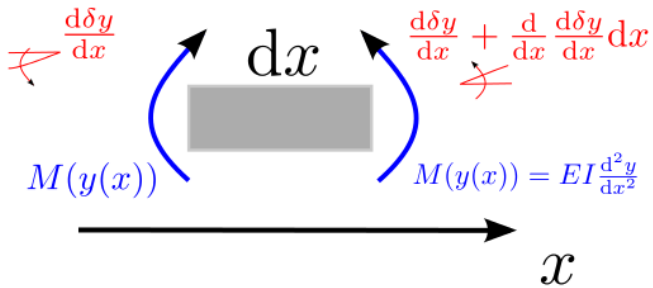
$$\text{Find } y \in \mathcal{V} = \{v \in C^2([0,L]) \mid v(0) = \bar{y}, \frac{\mathrm{d}v}{\mathrm{d}x}(0) = \bar{\theta}\}, \text{ such that,}$$

$$\forall \delta y \in \mathcal{W} = \{v \in C^2([0,L]) \mid v(0) = 0, \frac{\mathrm{d}v}{\mathrm{d}x}(0) = 0\}$$

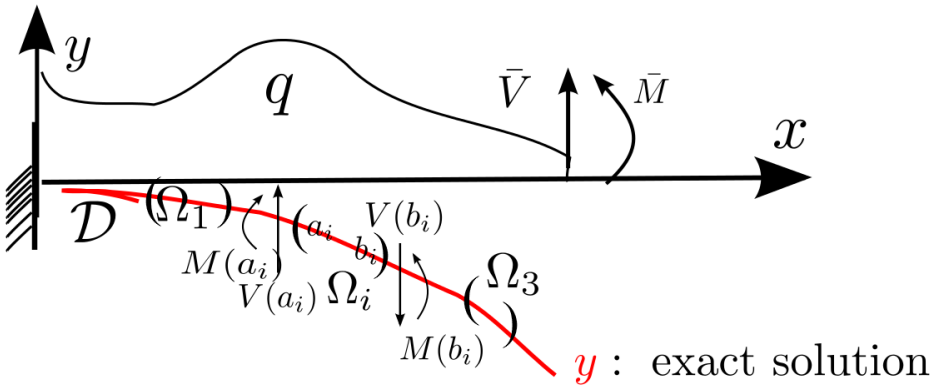
$$\underbrace{\int_0^L \overbrace{\delta y''(x)}^{\frac{\mathrm{d}}{\mathrm{d}x}\frac{\mathrm{d}\delta y}{\mathrm{d}x}} \overbrace{EIy''(x)}^{M(y(x))} \,\mathrm{d}x}_{\text{Virtual Internal Work}} = \underbrace{\int_0^L \delta y(x) q(x)\,\mathrm{d}x - \delta y(L)\bar{V} + \delta y'(L)\bar{M}}_{\text{Virtual External Work}} \tag{111}$$

Note that the internal work differential is:

$$\begin{aligned} \mathrm{d}V &= M(y(x)).(\frac{\mathrm{d}\delta y}{\mathrm{d}x} + \frac{\mathrm{d}}{\mathrm{d}x}\frac{\mathrm{d}\delta y}{\mathrm{d}x}\mathrm{d}x) - M(y(x)).\frac{\mathrm{d}\delta y}{\mathrm{d}x} \\ &= \frac{\mathrm{d}}{\mathrm{d}x}\frac{\mathrm{d}\delta y}{\mathrm{d}x}.M(y(x))\mathrm{d}x \end{aligned} \tag{112}$$



Balance method versus Energy method

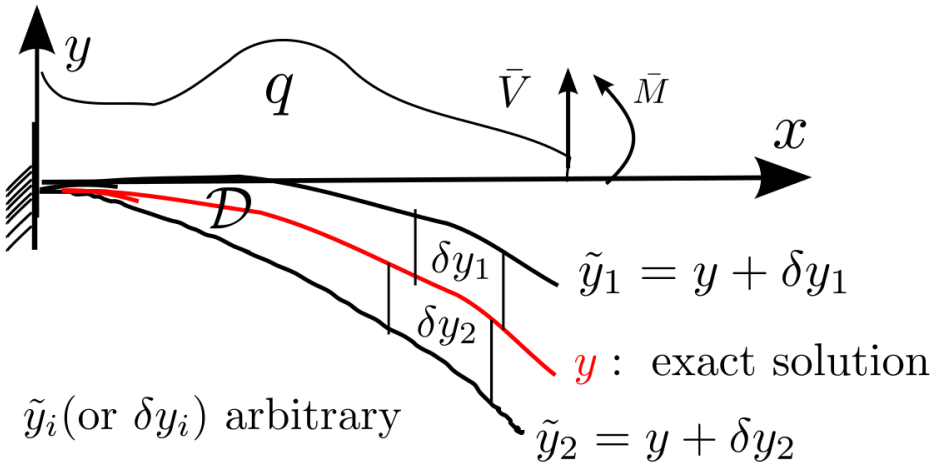


Ω is arbitrary

$\forall \Omega_i = (a_i, b_i) \subset \mathcal{D}$:

$$M(b_i) - M(a_i) - V(b_i)(b_i - a_i) + \int_{a_i}^{b_i} xq(x)\,\mathrm{d}x = 0$$

$$-V(b_i) + V(a_i) + \int_{a_i}^{b_i} q(x)\,\mathrm{d}x = 0$$



$\tilde{y}_i$(or δy_i) arbitrary

Find solution $y \in \mathcal{V} = \{v |\ v \in C^2(\mathcal{D}), v(0) = \bar{y}, v'(0) = \bar{\theta}\ \}$

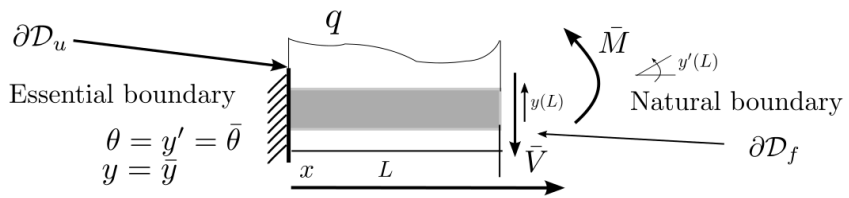
$\forall \tilde{y} \in \mathcal{V}$

$\Pi(y) \leq \Pi(\tilde{y}) \ \Rightarrow \delta Pi(y) = 0$

- Balance law $\rightarrow$ Strong form $\rightarrow$ Weighted Residual Form $\rightarrow$ Weak Form
- Energy Principle $\rightarrow$ Weak Form
- Balance law is stated on arbitrary subsets (Ω)of the domain $\mathcal{D}$. We derive a strong form (higher derivatives needed in the DE), obtain a integral form weighted residual statement and finally by reducing the derivative orders reach the "balanced" weak form which is suitable for numerical methods.
- Energy method yields the weak form in just one step and is much simpler because its statement on $\mathcal{D}$ is much closer in form to the final weak statement.

Energy method to Strong Form and Boundary Conditions

We realized the convenience of energy methods in deriving the weak form in one step. They can also be used to strong form and boundary conditions by the common approach of integration by part (divergence theorem in $D > 1$).
The weak form from (107) is:



$$\delta\Pi = \int_0^L \delta y''(x) EI y''(x)\,\mathrm{d}x - \int_0^L \delta y(x) q(x)\,\mathrm{d}x + \delta y(L)\bar{V} - \delta y'(L)\bar{M} = 0$$

Two consecutive integration by parts yield:

$$\int_0^L -\frac{\mathrm{d}\delta y}{\mathrm{d}x}\frac{\mathrm{d}}{\mathrm{d}x}(EI\frac{\mathrm{d}^2 y}{\mathrm{d}x^2})\,\mathrm{d}x - \int_0^L \delta y(x) q(x)\,\mathrm{d}x + \delta y(L)\bar{V} - \delta y'(L)\bar{M} + \delta y'(EI\frac{\mathrm{d}^2 y}{\mathrm{d}x^2})\Big|_0^L = 0 \Rightarrow$$

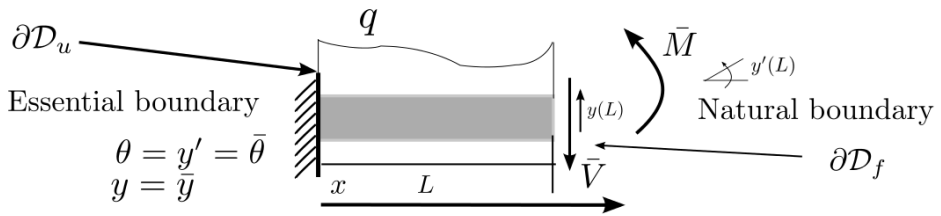
$$\int_0^L \delta y\left(\frac{\mathrm{d}^2 EI}{\mathrm{d}x^2}\frac{\mathrm{d}^2 y}{\mathrm{d}x^2} - q\right)\mathrm{d}x - \delta y'(L)\left(\bar{M} - EI\frac{\mathrm{d}^2 y}{\mathrm{d}x^2}(L)\right) - \delta y'(0) EI\frac{\mathrm{d}^2 y}{\mathrm{d}x^2}(0)$$

$$+ \delta y(L)\bar{V} - \{\delta y\frac{\mathrm{d}}{\mathrm{d}x}(EI\frac{\mathrm{d}^2 y}{\mathrm{d}x^2})\}\Big|_0^L = 0$$

$$\int_0^L \delta y\left(\frac{\mathrm{d}^2 EI}{\mathrm{d}x^2}\frac{\mathrm{d}^2 y}{\mathrm{d}x^2} - q\right)\mathrm{d}x - \delta y'(L)\left(\bar{M} - EI\frac{\mathrm{d}^2 y}{\mathrm{d}x^2}(L)\right) + \delta y(L)\left(\bar{V} - \frac{\mathrm{d}}{\mathrm{d}x}(EI\frac{\mathrm{d}^2 y}{\mathrm{d}x^2})\right)$$

$$+ \delta y(0)\frac{\mathrm{d}}{\mathrm{d}x}(EI\frac{\mathrm{d}^2 y}{\mathrm{d}x^2})(0) = 0$$

Energy method to Strong Form and Boundary Conditions



The terms in red in previous equation are zero because the function increment belongs to $\mathcal{W}$ given in (107):

$$\mathcal{W} = \{v \in C^2([0, L]) \mid v(0) = 0, \frac{\mathrm{d}v}{\mathrm{d}x}(0) = 0\}$$

Thus the last equation reduces to

$$\int_0^L \delta y \left(\frac{\mathrm{d}^2 EI}{\mathrm{d}x^2} \frac{\mathrm{d}^2 y}{\mathrm{d}x^2} - q \right) \mathrm{d}x \tag{113a}$$

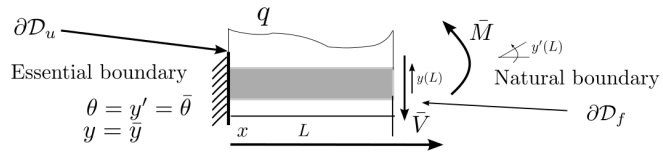
$$- \delta y'(L) \left(\bar{M} - EI \frac{\mathrm{d}^2 y}{\mathrm{d}x^2}(L) \right) \tag{113b}$$

$$+ \delta y(L) \left(\bar{V} - \frac{\mathrm{d}}{\mathrm{d}x}(EI \frac{\mathrm{d}^2 y}{\mathrm{d}x^2}) \right) = 0 \tag{113c}$$

By choosing δy such that $\delta y(L) = \frac{\mathrm{d}\delta y}{\mathrm{d}x}(L) = 0$ we reduce (113) to (113a) equal to zero for such δy increments:

Energy method to Strong Form and Boundary Conditions

That is, assuming $\frac{\mathrm{d}^2 EI}{\mathrm{d}x^2}\frac{\mathrm{d}^2 y}{\mathrm{d}x^2} - q \in C^4(\mathcal{D})$ and using (32) we obtain the fourth order DE:



$$\boxed{\forall x \in \mathcal{D} \quad \frac{\mathrm{d}^2 EI}{\mathrm{d}x^2}\frac{\mathrm{d}^2 y}{\mathrm{d}x^2}(x) - q(x) = 0} \tag{114}$$

Plugging this back in (113) we obtain: Thus the last equation reduces to

$$-\delta y'(L)\left(\bar{M} - EI\frac{\mathrm{d}^2 y}{\mathrm{d}x^2}(L)\right)$$
$$+\delta y(L)\left(\bar{V} - \frac{\mathrm{d}}{\mathrm{d}x}(EI\frac{\mathrm{d}^2 y}{\mathrm{d}x^2})\right) = 0$$

Then, first choosing $\frac{\mathrm{d}\delta y}{\mathrm{d}x}(L) = 0$ and $\delta y(L) \neq 0$ and vice versa we get:

$$\bar{M}(L) = EI\frac{\mathrm{d}^2 y}{\mathrm{d}x^2}(L) \tag{115a}$$

$$\bar{V}(L) = \frac{\mathrm{d}}{\mathrm{d}x}(EI\frac{\mathrm{d}^2 y}{\mathrm{d}x^2})(L) \tag{115b}$$

Equations (114) and (115) are the differential equation (strong form) and natural boundary conditions (in general equations for M and V) that we derived before. That is, energy method provides both the DE and natural boundary conditions.

Discretization

Discretization of the solution

- Discretization means reducing the infinite number of unknowns of the continuum problem to a discrete set of unknowns. Often, the discrete solution does not exactly match the exact solution. The accuracy of the approximation typically improves by increasing the number of unknowns.
- We denote the number of unknowns by n.
- Some common forms of discretization are:
  1. Values at a finite number of positions represent the solution:

     $$\text{Solution } u \text{ is represented by: } \{u(x_1), u(x_2), \ldots, u(x_n)\}.$$

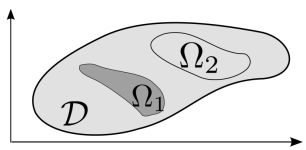
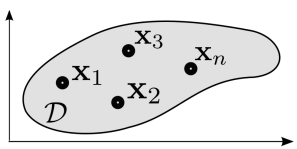
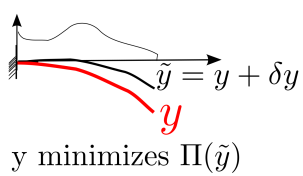
     This approach is used by Finite Difference (FD) and Finite Volume(FV) methods.
  2. Solution is represented by a finite number of functions:

     $$u^h(x) = \phi_p(x) + \sum_{i=1}^{n} a_i \phi_i(x)$$

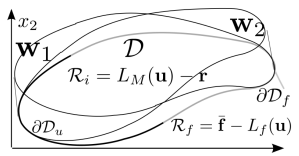
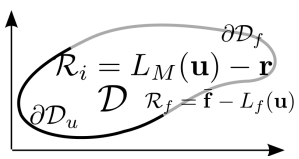
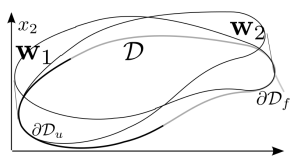
     where u^h is the symbol for discrete solution and $\phi_i(x)$ are trial or test functions. $\phi_p(x)$ is set to satisfy essential boundary conditions and will be discussed later. This approach is used by (discrete) weighted residual method, weak form, least square, and Ritz energy method.

Equations for discrete systems

- After discretizing the solution with n unknowns we need n equations to solve the discrete problem.
- The n equations are derived from different interpretation of the equations we derived so far. All these equations have a "for all" condition. In discrete systems the "for all" condition is replaced by a finite set.

Approach	Equation	Figure	Discretization	Discretization method
Balance Law (20)	$\forall \Omega \subset \mathcal{D} : \int_{\partial\Omega} (\mathbf{f}.\mathbf{n}) \mathrm{ds} - \int_{\Omega} \mathbf{r} \, \mathrm{dv} = \mathbf{0}$		Change $\forall \Omega$ to $\{\Omega_1, \Omega_2, \ldots, \Omega_n\}$	Similar to subdomain method in WRM
Strong Form (23)	$\forall \mathbf{x} \in \mathcal{D} : \nabla.\mathbf{f} - \mathbf{r} = \mathbf{0}$		Change $\forall \mathbf{x}$ to $\{\mathbf{x}_1, \mathbf{x}_2, \ldots, \mathbf{x}_n\}$	Collocation method in WRM. Also FD & FV.
Energy Method (80)	$\forall \tilde{y} \in \mathcal{V} : \ \Pi(y) \leq \Pi(\tilde{y})$		$\forall \{\tilde{a}_1, \ldots, \tilde{a}_n\} : \Pi(a_1, \ldots, a_n) \leq \Pi(\tilde{a}_1, \ldots, \tilde{a}_n) \Rightarrow \frac{\partial \Pi}{\partial a_1} = \cdots = \frac{\partial \Pi}{\partial a_n} = 0$	Ritz Energy Method. Also yields Weak Form.

Equations for discrete systems

Approach	Equation	Figure	Discretization	Discretization method
Weighted Residual Method (45)	$\forall \mathbf{w} \in \mathcal{W}$: $\int_{\mathcal{D}} \mathbf{w}.\mathcal{R}_i \ \mathrm{dv} + \int_{\partial\mathcal{D}_f} \mathbf{w}^f.\mathcal{R}_f \ \mathrm{ds} = \mathbf{0}$		Change $\forall \mathbf{w}$ to $\{\mathbf{w}_1, \mathbf{w}_2, \ldots, \mathbf{w}_n\}$	Weighted Residual Method (WRM)
Least Square (51)	$R^2 = \int_{\mathcal{D}} \mathcal{R}_i^2 \ \mathrm{dv} + \int_{\partial\mathcal{D}_f} \mathcal{R}_f^2 \ \mathrm{ds} = \mathbf{0}$		Change $R^2 = 0$ to $\forall\{\tilde{a}_1, \ldots, \tilde{a}_n\}$: $R^2(a_1, \ldots, a_n) \leq R^2(\tilde{a}_1, \ldots, \tilde{a}_n) \Rightarrow \frac{\partial R^2}{\partial a_1} = \cdots = \frac{\partial R^2}{\partial a_n} = 0$	Least Square method, a WRM for linear L_M (& L_f).
Weak Form (74)	$\forall \mathbf{w} \in \mathcal{W}$ $\int_{\mathcal{D}} L_m^w(\mathbf{w}) L_m(\mathbf{u}) \ \mathrm{dv} = \int_{\mathcal{D}} \mathbf{w}.\mathbf{r} \mathrm{dv} + \int_{\partial\mathcal{D}_f} \mathbf{w}.\bar{\mathbf{f}} \mathrm{ds}$		Change $\forall \mathbf{w}$ to $\{\mathbf{w}_1, \mathbf{w}_2, \ldots, \mathbf{w}_n\}$	Weak Formulation

Discrete systems

- In all approaches described, a continuum problem is discretized to a system of finite (n) unknowns. We observed how in each approach, by enforcing n discrete conditions corresponding to the continuum version, we could derive a system of n equations for n unknowns.
- The differential operators L_M (L_m for weak form) and L_f can all be nonlinear. In that case we obtain an n by n system of nonlinear equations.
- In all the approached discussed except Finite Difference and Finite Volume for the discrete version of the strong form, the solution is approximated by the summation of finite number of trial functions.

Trial functions

- As mentioned before, it is almost always beneficial (or necessary, *e.g.*, for energy method) to strongly satisfy the essential boundary conditions. That is the solution $\mathbf{u}$.

$$\mathbf{u} \in \mathcal{V} = \{\mathbf{v} \,|\mathbf{v} \in C^M(\mathcal{D}), \forall \mathbf{x} \in \partial\mathcal{D}_u \; L_u(\mathbf{u}) = \bar{\mathbf{u}}\} \tag{116}$$

We observe that for all the approached in the previous table, the solution $\mathbf{u}$ is approximated with trial functions $\boldsymbol{\phi}_i$ ($\boldsymbol{\phi}_i$ are assumed to be linearly independent):

$$\boxed{\mathbf{u} \approx \mathbf{u}^h = \sum_{j=1}^{n} a_j \boldsymbol{\phi}_j + \boldsymbol{\phi}_p} \quad \text{where} \tag{117a}$$

$$\mathbf{u} = \text{continuum (exact) solution} \tag{117b}$$

$$\mathbf{u}^h = \text{discrete (approximate) solution} \tag{117c}$$

$$\boldsymbol{\phi}_j = \text{trial (test) functions} \tag{117d}$$

$$a_j = \text{Unknown coefficients (unknowns of the discrete problem)} \tag{117e}$$

$$\boldsymbol{\phi}_p = \text{A solution satisfying essential boundary conditions} \tag{117f}$$

Trial functions

- We still want to enforce the essential boundary conditions for the discrete solution $\mathbf{u}$. We define,

$$\mathcal{V}_0 = \{\mathbf{v} \,|\mathbf{v} \in C^M(\mathcal{D}), \forall \mathbf{x} \in \partial\mathcal{D}_u \; L_u(\mathbf{u}) = 0\} \tag{118}$$

  Thus we observe,

$$\left.\begin{array}{r} \forall i \; \boldsymbol{\phi}_i \in \mathcal{V}_0 \\ \boldsymbol{\phi}_p \in \mathcal{V} \end{array}\right\} \Rightarrow \forall \mathbf{x} \in \partial\mathcal{D}_f \; \mathbf{u}^h(\mathbf{x}) = \sum_{j=1}^{n} a_j \boldsymbol{\phi}_j(\mathbf{x}) + \boldsymbol{\phi}_p(\mathbf{x}) = \mathbf{0} + \bar{\mathbf{u}} = \bar{\mathbf{u}} \tag{119}$$

  That is, the discrete solution **strongly** satisfies the essential boundary conditions.

- Function requirements for $\boldsymbol{\phi}_j$ and $\boldsymbol{\phi}_p$:

Function space	Continuity Requirement	Boundary Conditions
$\boldsymbol{\phi}_i \in \mathcal{V}_0$	$\boldsymbol{\phi}_i \in C^M(\mathcal{D})$	Trial functions satisfy homogeneous essential boundary conditions
$\boldsymbol{\phi}_p \in \mathcal{V}$	$\boldsymbol{\phi}_p \in C^M(\mathcal{D})$	A particular function satisfies the essential boundary conditions

- Note that if all the essential boundary conditions are zero, there is no need for the function $\boldsymbol{\phi}_p$ (*i.e.*, $\boldsymbol{\phi}_p$ can be chosen to be identically zero).

Discrete function spaces **(Optional)**

- The continuum function space $\mathcal{V}_0$ is infinite dimensional.
- We seek a discrete solution $\mathbf{u}^h$ of the form (117a):

$$\mathbf{u} \approx \mathbf{u}^h = \sum_{j=1}^{n} a_j \boldsymbol{\phi}_j + \boldsymbol{\phi}_p$$

  where as mentioned before all $\boldsymbol{\phi}_j \in \mathcal{V}_0$.
- The function space $\mathcal{V}_0^h$ is defined as:

$$\mathcal{V}_0^h = \{\sum_{j=1}^{n} \alpha_j \boldsymbol{\phi}_j |\ \forall \alpha_j \in \mathbb{R}\} \tag{120}$$

  This is nothing but the span of the functions $\boldsymbol{\phi}_j$:

$$\text{span}(\boldsymbol{\phi}_1, \ldots, \boldsymbol{\phi}_n) = [\boldsymbol{\phi}_1, \ldots, \boldsymbol{\phi}_n] = \{\sum_{j=1}^{n} \alpha_j \boldsymbol{\phi}_j |\ \forall \alpha_j \in \mathbb{R}\} \tag{121}$$

- $\mathcal{V}_0$ and $\mathcal{V}_0^h$ are both vector spaces with $\mathcal{V}_0^h$ being a subspace of $\mathcal{V}_0$.
- Eventually, we define the discrete function space $\mathcal{V}^h$ as,

$$\mathcal{V}^h = \boldsymbol{\phi}_p + \mathcal{V}_0^h := \{\mathbf{v} = \boldsymbol{\phi}_p + \mathbf{v}_0 |\ \mathbf{v}_0 \in \mathcal{V}_0^h\} = \{\mathbf{v} = \boldsymbol{\phi}_p + \sum_{j=1}^{n} \alpha_j \boldsymbol{\phi}_j |\ \forall \alpha_j \in \mathbb{R}\} \tag{122}$$

  where $\boldsymbol{\phi}_p$ can be any arbitrary function in $\mathcal{V}$.
- The discrete solution space $\mathcal{V}^h$ is n-dimensional ($\boldsymbol{\phi}_i$ must be linearly independent).

Discrete function spaces **(Optional)**

- Similar to the discrete solution space, we need to formalize the discrete weight function space for the methods that utilize a weight function (weighted residual and weak statement).
- For these methods, at continuum we require a statement to hold for all weight functions in $\mathcal{W}$.
- In discrete setting, as the solution space is n dimensional, we need to form an n-dimensional *subspace* of $\mathcal{W}$.
- Thus, we choose n linearly independent weight functions $\mathbf{w}_i \in \mathcal{W}$ to define a discrete weight function space $\mathcal{W}^h$:

$$\mathcal{W}^h = \text{span}(\mathbf{w}_1, \dots, \mathbf{w}_n) = [\mathbf{w}_1, \dots, \mathbf{w}_n] = \{\sum_{j=1}^{n} \alpha_j \mathbf{w}_j |\ \forall \alpha_j \in \mathbb{R}\} \tag{123}$$

- Weight functions always appear as linear differential operators in weighted residual and weak statements. Thus, we only need to enforce the discrete statement for n weight functions in $\mathcal{W}^h$ and the weak integral statement automatically holds for all $\mathbf{w} \in \mathcal{W}^h$. For example, consider the sample weighted residual statement below:

$$\int_{\mathcal{D}} \mathbf{w}.\mathcal{R}_i \, \text{dv} = 0 \quad \text{then} \quad \left.\begin{array}{l} \int_{\mathcal{D}} \mathbf{w}_1.\mathcal{R}_i \, \text{dv} = \mathbf{0} \\ \int_{\mathcal{D}} \mathbf{w}_2.\mathcal{R}_i \, \text{dv} = \mathbf{0} \end{array}\right\} \quad \Rightarrow$$

$$\forall \alpha_1, \alpha_2 \in \mathbb{R} : \int_{\mathcal{D}} (\alpha_1 \mathbf{w}_1 + \alpha_1 \mathbf{w}_2).\mathcal{R}_i \, \text{dv} = \alpha_1 \int_{\mathcal{D}} \mathbf{w}_1.\mathcal{R}_i \, \text{dv} + \alpha_2 \int_{\mathcal{D}} \mathbf{w}_2.\mathcal{R}_i \, \text{dv} = \mathbf{0}$$

Summary: Discrete function spaces

- To find a discrete solution of the form $\mathbf{u} \approx \mathbf{u}^h = \sum_{j=1}^{n} a_j \boldsymbol{\phi}_j + \boldsymbol{\phi}_p$, based on the space $\mathcal{V} = \{\mathbf{v} \,|\mathbf{v} \in C^M(\mathcal{D}, \forall \mathbf{x} \in \partial\mathcal{D}_f \;\; L_u(\mathbf{u}) = \bar{\mathbf{u}}\}$ (116) we define the vector space (118):

$$\mathcal{V}_0 = \{\mathbf{v} \,|\mathbf{v} \in C^M(\mathcal{D}, \forall \mathbf{x} \in \partial\mathcal{D}_u \;\; L_u(\mathbf{u}) = 0\}$$

  and choose n linearly independent trial functions $\boldsymbol{\phi}_j \in \mathcal{V}_0$ and an arbitrary $\boldsymbol{\phi}_p \in \mathcal{V}$.
- The the discrete solution space is an n-dimensional function space given by (122):

$$\mathcal{V}^h = \boldsymbol{\phi}_p + \mathcal{V}_0^h := \boldsymbol{\phi}_p + \{\sum_{j=1}^{n} \alpha_j \boldsymbol{\phi}_j | \; \forall \alpha_j \in \mathbb{R}\} = \{\mathbf{v} = \boldsymbol{\phi}_p + \sum_{j=1}^{n} \alpha_j \boldsymbol{\phi}_j | \; \forall \alpha_j \in \mathbb{R}\}$$

- For problems that are stated based on weight functions (weighted residual and weak problem), based on the continuum weight function space $\mathcal{W}$, we choose n linearly independent weight functions $\mathbf{w}_i \in \mathcal{W}$ and form the discrete weight function space (123):

$$\mathcal{W}^h = \text{span}(\mathbf{w}_1, \ldots, \mathbf{w}_n) = \{\sum_{j=1}^{n} \alpha_j \mathbf{w}_j | \; \forall \alpha_j \in \mathbb{R}\}$$

- Due to linearity of the weight functions in weighted statements we only need to satisfy the discrete statement for n weight functions in $\mathcal{W}^h$ (for example $\mathbf{w}_1, \ldots, \mathbf{w}_n$) to ensure its satisfaction for all $\mathbf{w} \in \mathcal{W}^h$.

Indicial notation

For brevity we follow the following notations

$$a_j b_j = \sum_{j=1}^{n} a_j b_j \quad \textbf{Einstein notation} \tag{124a}$$

$$\text{statement for } a_i : \text{ statement for all } i \in \{1, 2, \ldots, n\} \tag{124b}$$

The range of indices depend on a particular problem. For example for ϕ_i, n refers to the number of test functions and for tensor expressions such as $\sigma_{ij} = C_{ijkl}\epsilon_{kl}$ it refers to spatial dimension.

Matrix equations for discrete systems

- For all the discrete systems discussed we obtain a n by n system of equations.
- To better understand the details of each approach we assume the differential operators L_M (L_m weak form) and L_f are all linear. The same solution process with minor modification can be applied to nonlinear problems where a linear system of equation should be solved for each solution iteration.
- For linear systems, we obtain a linear matrix equation of the form:

$$\boxed{\mathbf{K}\mathbf{a} = \mathbf{F}} \tag{125}$$

  where $\mathbf{K}$ is an $n \times n$ matrix and $\mathbf{a}$ and $\mathbf{F}$ are solution coefficient and right hand side (force) vectors, respectively.
- Linear independence of trial (and weight functions when applicable) is an essential condition for the solvability of (125) (*i.e.*, $\det \mathbf{K} \neq 0$).
- Next, we obtain the matrix equation (125) for various discrete solution schemes and show that several of them can be cast into a weighted residual form.

Weighted Residual Method

We want to obtain the discrete solution corresponding to the continuum weighted residual statement (45):

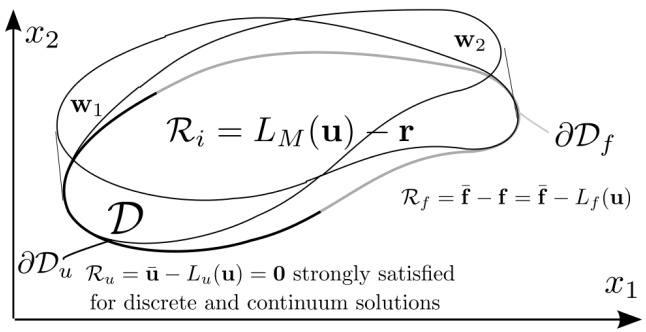
$$\text{Find } \mathbf{u} \in \mathcal{V} \text{ such that } \forall \mathbf{w} \in \mathcal{W}: \int_{\mathcal{D}} \mathbf{w}.\mathcal{R}_i(\mathbf{u})\,\mathrm{dv} + \int_{\partial\mathcal{D}_f} \mathbf{w}^f.\mathcal{R}_f(\mathbf{u})\,\mathrm{ds} = \mathbf{0}$$

$\mathbf{w}^f$ refers to the form of the weight functions used for the boundary integrals. Often, it is the same as $\mathbf{w}$ (that is $w_i^f = w_i$) but to keep the generality, w_i^f corresponds to w_i.
By changing $\mathbf{u}$ to $\mathbf{u}^h \in \mathcal{V}^h$ and restricting $\mathbf{w}$ to $\mathcal{W}^h$:

$$\text{Find } \mathbf{u}^h \in \mathcal{V}^h \text{ such that } \forall \mathbf{w} \in \mathcal{W}^h: \int_{\mathcal{D}} \mathbf{w}.\mathcal{R}_i(\mathbf{u}^h)\,\mathrm{dv} + \int_{\partial\mathcal{D}_f} \mathbf{w}^f.\mathcal{R}_f(\mathbf{u}^h)\,\mathrm{ds} = \mathbf{0} \tag{126}$$

For $\mathbf{u}^h = \sum_{j=1}^{n} a_j \boldsymbol{\phi}_j + \boldsymbol{\phi}_p$ (117a) we have,

$$\left.\begin{aligned} \mathcal{R}_i(\mathbf{u}) &= L_M(\mathbf{u}) - \mathbf{r} \\ \mathcal{R}_f(\mathbf{u}) &= \bar{\mathbf{f}} - L_f(\mathbf{u}) \\ L_M(\mathbf{u}), L_f(\mathbf{u}) &: \text{ assumed to be linear} \\ \mathbf{u}^h &= a_j \boldsymbol{\phi}_j + \boldsymbol{\phi}_p \end{aligned}\right\} \Rightarrow \tag{127}$$



$$\begin{aligned} \mathcal{R}_i(\mathbf{u}^h) &= a_j L_M(\boldsymbol{\phi}_j) + (L_M(\boldsymbol{\phi}_p) - \mathbf{r}) = [L_M(\boldsymbol{\phi})]^{\mathrm{T}}[a] + (L_M(\boldsymbol{\phi}_p) - \mathbf{r}) \\ \mathcal{R}_f(\mathbf{u}^h) &= -a_j L_f(\boldsymbol{\phi}_j) + (\bar{\mathbf{f}} - L_f(\boldsymbol{\phi}_p)) = -[L_f(\boldsymbol{\phi})]^{\mathrm{T}}[a] + (\bar{\mathbf{f}} - L_f(\boldsymbol{\phi}_p)) \end{aligned} \tag{128}$$

Weighted Residual Method

- In (127) the row and column matrix product is:

$$[L_M(\boldsymbol{\phi})]^{\mathrm{T}}[a] = \begin{bmatrix} L_M(\boldsymbol{\phi}_1) & L_M(\boldsymbol{\phi}_2) & \cdots & L_M(\boldsymbol{\phi}_n) \end{bmatrix} \begin{bmatrix} a_1 \\ a_2 \\ \vdots \\ a_n \end{bmatrix} = a_j L_M(\boldsymbol{\phi}_j) \tag{129}$$

- We plug (127) into (126) to obtain,

$$\begin{aligned}
&\text{Find } [a] = \begin{bmatrix} a_1 & a_2 & \cdots & a_n \end{bmatrix}^{\mathrm{T}} \text{ such that} \\
&\int_{\mathcal{D}} [\mathbf{w}].\left\{[L_M(\boldsymbol{\phi})]^{\mathrm{T}}[a] + (L_M(\boldsymbol{\phi}_p) - \mathbf{r})\right\} \,\mathrm{dv} + \\
&\int_{\partial\mathcal{D}_f} [\mathbf{w}^f].\left\{-[L_f(\boldsymbol{\phi})]^{\mathrm{T}}[a] + (\bar{\mathbf{f}} - L_f(\boldsymbol{\phi}_p))\right\} \,\mathrm{ds} = \mathbf{0} \quad \Rightarrow \\
&\left\{\int_{\mathcal{D}} [\mathbf{w}].[L_M(\boldsymbol{\phi})]^{\mathrm{T}} \,\mathrm{dv} - \int_{\partial\mathcal{D}_f} [\mathbf{w}^f].[L_f(\boldsymbol{\phi})]^{\mathrm{T}} \,\mathrm{ds}\right\}[a] = \\
&\left\{\int_{\mathcal{D}} [\mathbf{w}].(\mathbf{r} - L_M(\boldsymbol{\phi}_p)) \,\mathrm{dv} + \int_{\partial\mathcal{D}_f} [\mathbf{w}^f].(L_f(\boldsymbol{\phi}_p) - \bar{\mathbf{f}}) \,\mathrm{ds}\right\}
\end{aligned} \tag{130}$$

$[\mathbf{w}]$ and $[\mathbf{w}^f]$ correspond to vectors of weight functions on $\mathcal{D}$ and $\partial\mathcal{D}_f$:

$$[\mathbf{w}] = \begin{bmatrix} \mathbf{w}_1 \\ \vdots \\ \mathbf{w}_n \end{bmatrix} \quad \text{and} \quad [\mathbf{w}^f] = \begin{bmatrix} w_1^f \\ \vdots \\ w_n^f \end{bmatrix} \tag{131}$$

Weighted Residual Method

- According to equations (125), and (130), for a given $\boldsymbol{\phi}_p$ the solution to the discrete weighted residual statement (126) is obtained from $\mathbf{Ka} = \mathbf{F}$:

$$\mathbf{K} = \int_{\mathcal{D}} [\mathbf{w}].[L_M(\boldsymbol{\phi})]^{\mathrm{T}} \,\mathrm{dv} - \int_{\partial\mathcal{D}_f} [\mathbf{w}^f].[L_f(\boldsymbol{\phi})]^{\mathrm{T}} \,\mathrm{ds} \tag{132a}$$

$$\mathbf{F} = \int_{\mathcal{D}} [\mathbf{w}].(\mathbf{r} - L_M(\boldsymbol{\phi}_p)) \,\mathrm{dv} + \int_{\partial\mathcal{D}_f} [\mathbf{w}^f].(L_f(\boldsymbol{\phi}_p) - \bar{\mathbf{f}}) \,\mathrm{ds} \tag{132b}$$

  or alternatively the individual components are given,

$$K_{ij} = \int_{\mathcal{D}} \mathbf{w}_i L_M(\boldsymbol{\phi}_j) \,\mathrm{dv} - \int_{\partial\mathcal{D}_f} \mathbf{w}_i^f L_f(\boldsymbol{\phi}_j) \,\mathrm{ds} \tag{133a}$$

$$F_i = \int_{\mathcal{D}} \mathbf{w}_i(\mathbf{r} - L_M(\boldsymbol{\phi}_p)) \,\mathrm{dv} + \int_{\partial\mathcal{D}_f} \mathbf{w}_i^f (L_f(\boldsymbol{\phi}_p) - \bar{\mathbf{f}}) \,\mathrm{ds} \tag{133b}$$

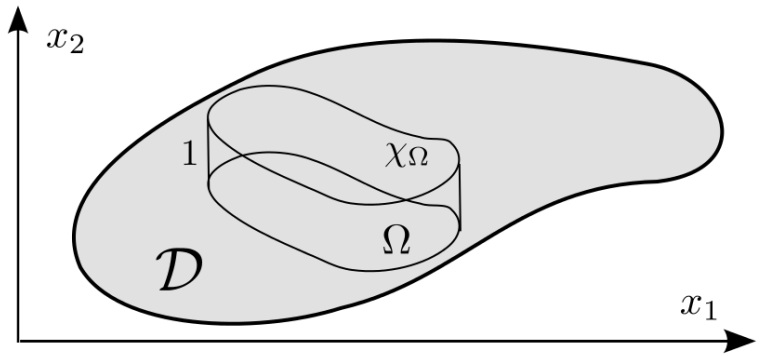
- After solving for $\mathbf{a}$ the discrete solution is obtained from $\mathbf{u}^h = \sum_{j=1}^{n} a_j \boldsymbol{\phi}_j + \boldsymbol{\phi}_p$ (117a).

Weighted Residual Method: Weight options

Weighted residual method is the basis of many numerical methods. In the following we only discuss a few of the weight function options and elaborate how they are related to the continuum principles discussed in the table at the beginning of the section:

- Subdomain method: Weight functions are the so-called characteristic functions of arbitrary sets $\Omega \subset \mathcal{D}$. As will be discussed, this choice of weighted functions resemble satisfaction of the balance law for a finite number of Ω_i.
- Collocation method: The weight functions are delta Dirac "functions". This form corresponds to satisfaction of the strong form (differential equations) at a finite number of points.
- Least Square method: For linear operators L_M and L_f we observe that least square method corresponds to a particular choice of weight function in weighed residual method.
- Galerkin method corresponds to weight function being equal to trial functions. A large group of numerical methods, including spectral and various finite element methods fall into this group.

Weighted Residual Method: Subdomain Method



- We define the characteristic function for the set $\Omega \subset \mathcal{D}$

$$\chi_\Omega(\mathbf{x}) = \begin{cases} 1 & \mathbf{x} \in \Omega \\ 0 & \text{otherwise} \end{cases} \tag{134}$$

- In subdomain method, the weight functions are n characteristic functions for sets $\Omega_1, \ldots, \Omega_n$:

$$[\mathbf{w}] = \begin{bmatrix} \chi_{\Omega_1} \\ \chi_{\Omega_2} \\ \vdots \\ \chi_{\Omega_n} \end{bmatrix} \tag{135}$$

Weighted Residual Method: Subdomain Method

- According to (135) and (132) the matrix equations for the subdomain method are:

$$\mathbf{K} = \int_{\mathcal{D}} [\chi_{\Omega_i}].[L_M(\boldsymbol{\phi})]^{\mathrm{T}} \,\mathrm{dv} - \int_{\partial\mathcal{D}_f} [\chi_{\Omega_i}].(L_f(\boldsymbol{\phi}_p) - \bar{\mathbf{f}}) \,\mathrm{ds} \tag{136a}$$

$$\mathbf{F} = \int_{\mathcal{D}} [\chi_{\Omega_i}].(\mathbf{r} - L_M(\boldsymbol{\phi}_p)) \,\mathrm{dv} + \int_{\partial\mathcal{D}_f} [\chi_{\Omega_i}].(L_f(\boldsymbol{\phi}_p) - \bar{\mathbf{f}}) \,\mathrm{ds} \tag{136b}$$

- Based on (134) and (133) the component expressions are:

$$K_{ij} = \int_{\Omega_i} L_M(\boldsymbol{\phi}_j) \,\mathrm{dv} - \int_{\partial(\Omega_i)_f} L_f(\boldsymbol{\phi}_j) \,\mathrm{ds} \tag{137a}$$

$$F_i = \int_{\Omega_i} \mathbf{w}_i(\mathbf{r} - L_M(\boldsymbol{\phi}_p)) \,\mathrm{dv} + \int_{\partial(\Omega_i)_f} (L_f(\boldsymbol{\phi}_p) - \bar{\mathbf{f}}) \,\mathrm{ds} \tag{137b}$$

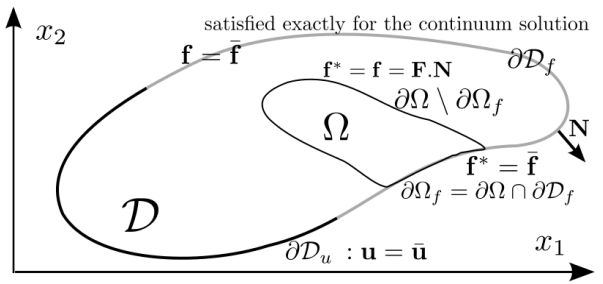
where $\partial(\Omega_i)_f := \Omega_i \cap \partial\mathcal{D}_f$ is the intersection of the boundary of Ω_i with natural boundary $\partial\mathcal{D}_f$.

Optional: Subdomain Method vs. Balance laws

- In subdomain method for a weight function $\mathbf{w} = \chi_\Omega$ the discrete solution $\mathbf{u}^h \in \mathcal{V}^h$ satisfies (126):

$$\int_\Omega \mathcal{R}_i(\mathbf{u}^h)\,\mathrm{dv} + \int_{\partial\Omega_f} \mathcal{R}_f(\mathbf{u}^h)\,\mathrm{ds} = \mathbf{0} \qquad (138a)$$

$$\partial\Omega_f = \partial\Omega \cap \partial\mathcal{D}_f \qquad (138b)$$



- If the residuals are obtained from a balance law we have,

$$\mathcal{R}_i(\mathbf{u}^h) = \nabla.\mathbf{F}(\mathbf{u}^h) - \mathbf{r} \qquad \text{residual of strong form} \qquad (139a)$$

$$\mathcal{R}_f(\mathbf{u}^h) = \bar{\mathbf{f}} - \mathbf{f}(\mathbf{u}^h) \qquad \text{residual of natural boundary condition} \qquad (139b)$$

where $\mathbf{F}, \mathbf{r}, \bar{\mathbf{f}}$ are the flux tensor, source term, and natural boundary flux for the given balance law, respectively. The vector $\mathbf{f}(\mathbf{u}^h) = \mathbf{F}(\mathbf{u}^h).\mathbf{N}$ is the flux through the boundary and $\mathbf{N}$ is the normal vector.

Optional: Subdomain Method vs. Balance laws

- We plug (139) for the weight function χ_Ω and employ divergence theorem to obtain,

$$\int_\Omega \nabla.\mathbf{F}(\mathbf{u}^h) - \mathbf{r}\,\mathrm{dv} + \int_{\partial\Omega_f} \left\{\bar{\mathbf{f}} - \mathbf{f}(\mathbf{u}^h)\right\}\,\mathrm{ds} = \mathbf{0} \quad \text{Divergence theorem} \Rightarrow$$

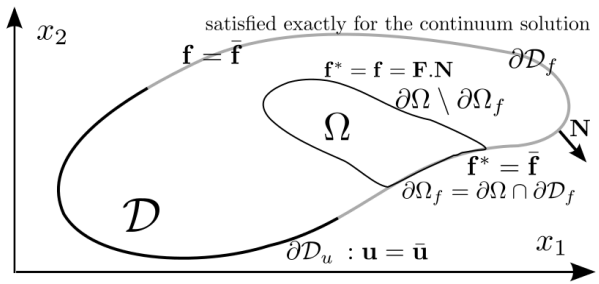
$$\int_{\partial\Omega} \underbrace{\mathbf{F}(\mathbf{u}^h).\mathbf{N}}_{\mathbf{f}(\mathbf{u}^h)}\,\mathrm{ds} + \int_{\partial\Omega\cap\partial\mathcal{D}_f} \left\{\bar{\mathbf{f}} - \mathbf{f}(\mathbf{u}^h)\right\}\,\mathrm{ds} = \int_\Omega \mathbf{r}\,\mathrm{dv} \Rightarrow$$

$$\int_{\partial\Omega\setminus\partial\mathcal{D}_f} \mathbf{f}(\mathbf{u}^h)\,\mathrm{ds} + \int_{\partial\Omega\cap\partial\mathcal{D}_f} \left\{\mathbf{f}(\mathbf{u}^h) + \left[\bar{\mathbf{f}} - \mathbf{f}(\mathbf{u}^h)\right]\right\}\,\mathrm{ds} = \int_\Omega \mathbf{r}\,\mathrm{dv} \quad \Rightarrow$$

$$\int_{\partial\Omega\setminus\partial\mathcal{D}_f} \mathbf{f}(\mathbf{u}^h)\,\mathrm{ds} + \int_{\partial\Omega\cap\partial\mathcal{D}_f} \bar{\mathbf{f}}\,\mathrm{ds} = \int_\Omega \mathbf{r}\,\mathrm{dv}$$

- That is if we define the numerical flux $\mathbf{f}^*$ we conclude,

$$\mathbf{f}^*(\mathbf{u}(\mathbf{x})^h) = \begin{cases} \mathbf{f}(\mathbf{u}(\mathbf{x})^h) & \mathbf{x} \in \partial\Omega \setminus \partial\mathcal{D}_f \\ \bar{\mathbf{f}}(\mathbf{x}) & \mathbf{x} \in \partial\Omega \cap \partial\mathcal{D}_f \end{cases} \tag{140}$$



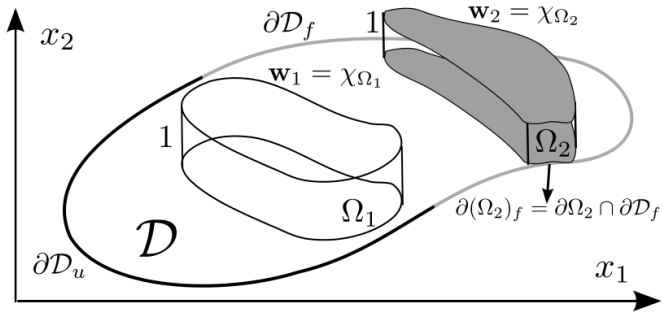
Subdomain WR method for Ω is equivalent to the balance law with respect to $\mathbf{f}^*$

$$\int_{\partial\Omega} \mathbf{f}^*(\mathbf{u}^h)\,\mathrm{ds} = \int_\Omega \mathbf{r} \tag{141}$$

Summary: Subdomain Method vs. Balance laws

- We mentioned that for the solution for the discrete problem, we seek for an "approximate solution" of the form $\mathbf{u}^h = \sum_{j=1}^{n} a_j \boldsymbol{\phi}_j + \boldsymbol{\phi}_p$ (117a) in the n-dimensional space $\mathcal{V}^h$.
- The n equations for the solutions of the unknowns a_i can be formed by,

Subdomain method

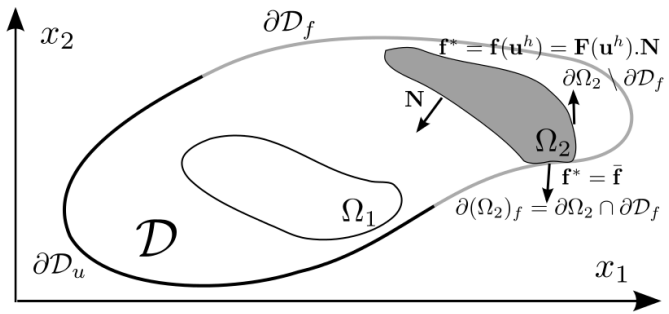


- For weight functions $\mathbf{w}_i = \chi_{\Omega_i}$ the weighted residual statement is $\forall i \in \{1, \ldots, n\}$ (*cf.* (126) and (138)):

$$\int_{\mathcal{D}} \mathbf{w}_i . \mathcal{R}_i(\mathbf{u}^h)\, \mathrm{dv} + \int_{\partial\mathcal{D}_f} \mathbf{w}_i^f . \mathcal{R}_f(\mathbf{u}^h)\, \mathrm{ds} = \mathbf{0}$$

$$\int_{\Omega_i} \mathcal{R}_i(\mathbf{u}^h)\, \mathrm{dv} + \int_{\partial(\Omega_i)_f} \mathcal{R}_f(\mathbf{u}^h)\, \mathrm{ds} = \mathbf{0}$$

Balance law approach



- We demonstrated that the subdomain weighted residual statement is equivalent to satisfying the balance law for $\forall i \in \{1, \ldots, n\}$ (*cf.* (141)):

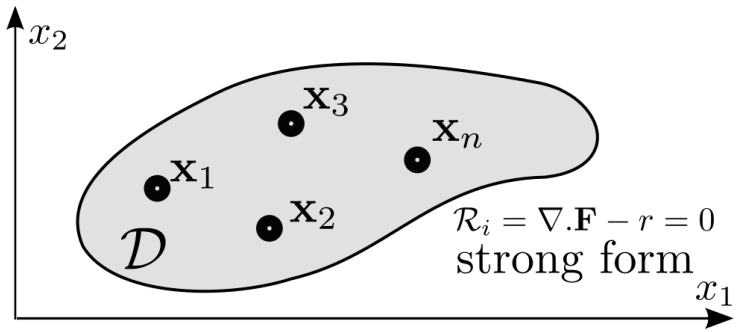
$$\int_{\partial\Omega_i} \mathbf{f}^*(\mathbf{u}^h)\, \mathrm{ds} = \int_{\Omega_i} \mathbf{r}$$

- The numerical flux $\mathbf{f}^*$ is defined by (140):

$$\mathbf{f}^*(\mathbf{u}(\mathbf{x})^h) = \begin{cases} \mathbf{f}(\mathbf{u}(\mathbf{x})^h) & \mathbf{x} \in \partial\Omega_i \setminus \partial\mathcal{D}_f \\ \bar{\mathbf{f}}(\mathbf{x}) & \mathbf{x} \in \partial\Omega_i \cap \partial\mathcal{D}_f \end{cases}$$

Optional: While for continuum solution $\mathbf{f}^* = \mathbf{f}$ everywhere (because natural BCs are strongly satisfied) in discrete setting the definition of $\mathbf{f}^*$ ensures the statement of the balance law with respect to the correct flux $\bar{\mathbf{f}}$ on $\partial\mathcal{D}_f$.

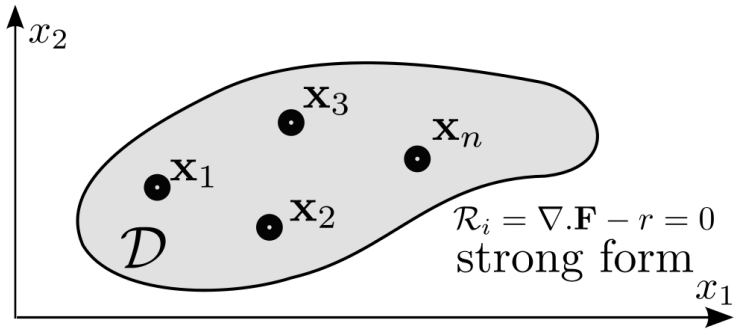
Weighted Residual Method: Collocation Method



- While subdomain method corresponds to discrete version of balance laws, the collocation method corresponds to discrete satisfaction of the strong form
- For simplicity, we assume that ϕ_p not only satisfies both essential and natural boundary conditions while functions ϕ_i satisfy the homogeneous version of both boundary conditions. Thus, the discrete solution $\mathbf{u}^h$ strongly satisfies all the boundary conditions. We want to demonstrate how the collocation method satisfies the strong form at finite number of points.
- With the temporary change the discrete weighted residual (126) simplifies to,

$$\text{Find } \mathbf{u}^h \in \mathcal{V}^h \text{ such that } \forall \mathbf{w} \in \mathcal{W}^h : \int_{\mathcal{D}} \mathbf{w}.\mathcal{R}_i(\mathbf{u}^h)\,\mathrm{dv} = \int_{\mathcal{D}} \mathbf{w}.\{L_M(\mathbf{u}^h(\mathbf{x})) - \mathbf{r}\}\,\mathrm{dv} = \mathbf{0} \tag{142}$$

Weighted Residual Method: Collocation Method



- In collocation method, the weight functions are defined as,

$$\boxed{\mathbf{w}_i(\mathbf{x}) = \delta(\mathbf{x} - \mathbf{x}_i) \text{ for } i \in \{1, 2, \ldots, n\}} \tag{143}$$

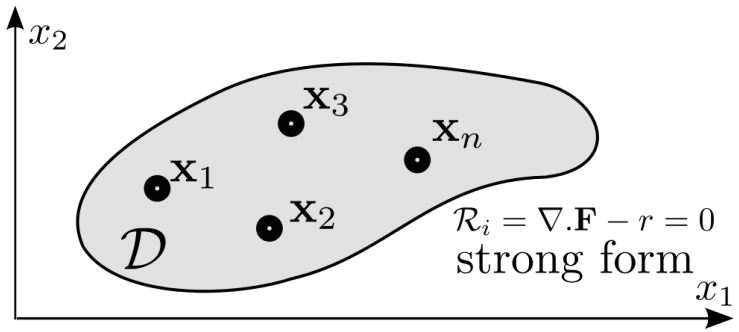
where $\mathbf{x}_1, \mathbf{x}_2, \ldots, \mathbf{x}_n$ are n arbitrary points in $\mathcal{D}$. Note that $\mathbf{w}_i$ are not even "functions" let alone the $C^0(\mathcal{D})$ condition that we typically stipulate for weight functions in WRM! However, all the integrals with these weight functions can be evaluated. This would be further commented later.

- We plug (143) into (142) to obtain,

$$\text{Find } \mathbf{u}^h \in \mathcal{V}^h \text{ such that } \forall i \in \{1, 2, \ldots, n\} \int_{\mathcal{D}} \mathbf{w}_i(\mathbf{x}).\mathcal{R}_i(\mathbf{u}^h(\mathbf{x}))\,\mathrm{dv} = \tag{144}$$

$$\int_{\mathcal{D}} \delta(\mathbf{x} - \mathbf{x}_i).\mathcal{R}_i(\mathbf{u}^h(\mathbf{x}))\,\mathrm{dv} = \mathcal{R}_i(\mathbf{u}^h(\mathbf{x}_i)) = \mathbf{0}$$

Weighted Residual Method: Collocation Method



- According to (144) in collocation method we exactly satisfy the strong form at n discrete points:

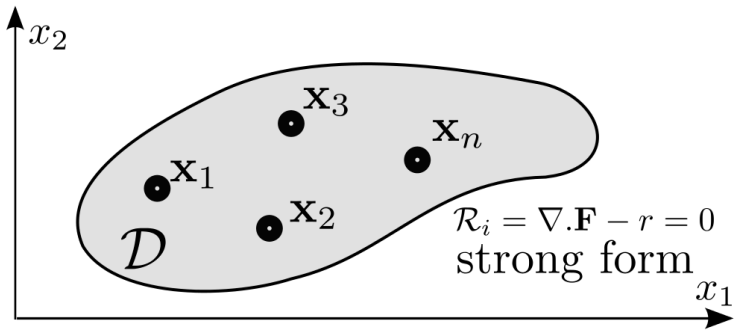
$$\text{Solution } \mathbf{u}^h = a_j \boldsymbol{\phi}_j + \boldsymbol{\phi}_p \in \mathcal{V}^h \text{ satisfies} \tag{145}$$

$$\forall i \in \{1, 2, \ldots, n\}\ \mathcal{R}_i(\mathbf{u}^h(\mathbf{x}_i)) = L_M(\mathbf{u}^h(\mathbf{x}_i)) - \mathbf{r}(\mathbf{x}_i) = \mathbf{0}$$

- For linear operator L_M (*cf.* (128)):

$$L_M(\mathbf{u}^h(\mathbf{x}_i)) = a_j L_M(\boldsymbol{\phi}_j)(\mathbf{x}_i) + L_M(\boldsymbol{\phi}_p)(\mathbf{x}_i) \tag{146}$$

Weighted Residual Method: Collocation Method

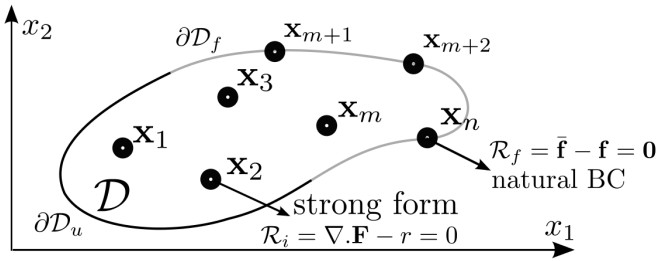


- From equations (145) and (146) the linear system $\mathbf{Ka} = \mathbf{F}$ (*cf.* (125), (132), and (133)) takes the form:

$$K_{ij} = \int_{\mathcal{D}} \mathbf{w}_i L_M(\boldsymbol{\phi}_j)\,\mathrm{dv} = L_M(\boldsymbol{\phi}_j)(\mathbf{x}_i) \tag{147a}$$

$$F_i = \int_{\mathcal{D}} \mathbf{w}_i(\mathbf{r} - L_M(\boldsymbol{\phi}_p))\,\mathrm{dv} = \mathbf{r}(\mathbf{x}_i) - L_M(\boldsymbol{\phi}_j)(\mathbf{x}_i) \tag{147b}$$

Optional: Weighted Residual Method: Collocation Method



- Without going to detail, equations for more practical case where natural boundary conditions are not strongly satisfied are provided. In this case, we solve the usual discrete system (126)

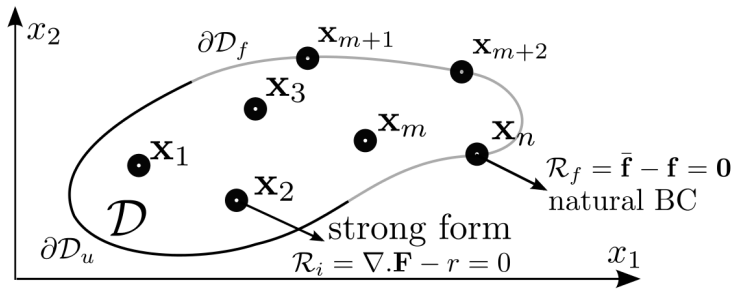
$$\text{Find } \mathbf{u}^h \in \mathcal{V}^h \text{ such that } \forall \mathbf{w} \in \mathcal{W}^h : \int_{\mathcal{D}} \mathbf{w}.\mathcal{R}_i(\mathbf{u}^h)\,\mathrm{dv} + \int_{\partial\mathcal{D}_f} \mathbf{w}^f.\mathcal{R}_f(\mathbf{u}^h)\,\mathrm{ds} = \mathbf{0}$$

- As shown in the figure

$$\text{For } i = 1, \ldots, m,\ \mathbf{x}_i \in \mathcal{D} \setminus \partial\mathcal{D} \text{ (interior of } \mathcal{D}) \qquad (148)$$
$$\text{For } i = m+1, \ldots, n,\ \mathbf{x}_i \in \partial\mathcal{D}_f$$

- $\mathbf{w}_i$ are set to delta Dirac function on set $\mathcal{D}$ for $\mathbf{x}_1, \ldots, \mathbf{x}_m$ and on set $\partial\mathcal{D}_f$ for $\mathbf{x}_{m+1}, \ldots, \mathbf{x}_n$.
- For $i = 1, \ldots, m$ interior residuals, $\mathcal{R}_i(\mathbf{x}_i)$ will be activated, while for $m+1, \ldots, n$ $\mathcal{R}_f(\mathbf{x}_i)$ will be activated.

Optional: Weighted Residual Method: Collocation Method

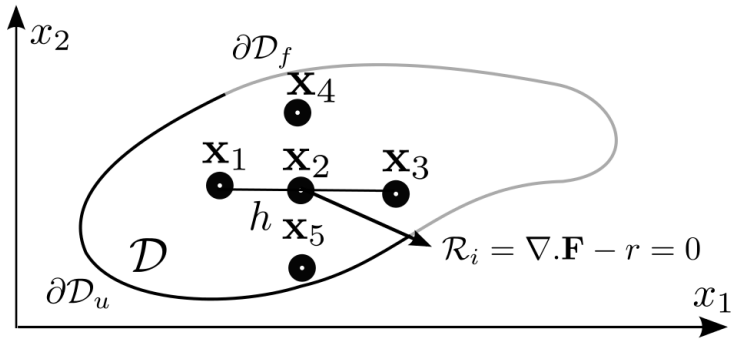


- From the previous discussion the linear system $\mathbf{Ka} = \mathbf{F}$ (*cf.* (125), (132), and (133)) takes the form:

$$K_{ij} = \begin{cases} \int_{\mathcal{D}} \mathbf{w}_i L_M(\boldsymbol{\phi}_j)\,\mathrm{dv} = L_M(\boldsymbol{\phi}_j)(\mathbf{x}_i) & i \leq m \\ -\int_{\partial\mathcal{D}_f} \mathbf{w}_i^f . L_f(\boldsymbol{\phi}_j)\,\mathrm{ds} = -L_f(\boldsymbol{\phi}_j)(\mathbf{x}_i) & m < i \leq n \end{cases} \tag{149a}$$

$$F_i = \begin{cases} \int_{\mathcal{D}} \mathbf{w}_i(\mathbf{r} - L_M(\boldsymbol{\phi}_p))\,\mathrm{dv} = \mathbf{r}(\mathbf{x}_i) - L_M(\boldsymbol{\phi}_j)(\mathbf{x}_i) & i \leq m \\ \int_{\partial\mathcal{D}_f} \mathbf{w}_i^f .(L_f(\boldsymbol{\phi}_p) - \bar{\mathbf{f}})\,\mathrm{ds} = L_f(\boldsymbol{\phi}_p)(\mathbf{x}_i) - \bar{\mathbf{f}}(\mathbf{x}_i) & m < i \leq n \end{cases} \tag{149b}$$

Collocation method versus Finite Difference

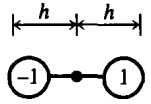
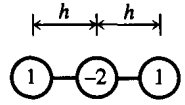
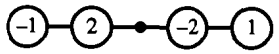
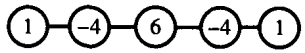
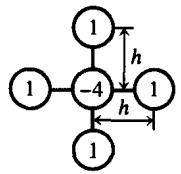
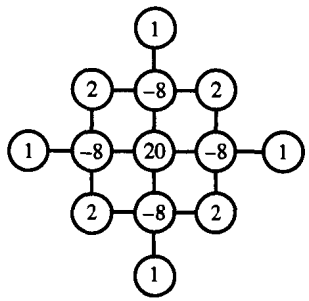


- Both Collocation and Finite Difference methods directly work with the strong form and boundary conditions.
- Collocation method is a particular class of weighted residual method where the solution is interpolated as $\mathbf{u}^h = a_j \boldsymbol{\phi}_j + \boldsymbol{\phi}_p$.
- Finite Difference does not interpolate the solution with trial function. Rather, it uses discrete values of the function on often regular grids to approximate differential operators.
- Differential operators in Finite Difference method are approximate, where as in collocation method the solution $\mathbf{u}^h$ exactly satisfies the strong form at $\mathbf{x}_i$.
- As an example, let us assume the differential operator L_M in $\mathcal{R}_i$ includes a Laplacian operator $\Delta u = \frac{\partial^2 u}{\partial x_1^2} + \frac{\partial^2 u}{\partial x_2^2}$. The finite difference approximation of Laplacian on a uniform grid with size h would be,

$$\Delta u(\mathbf{x}_2) = \frac{1}{h^2}\left(u(\mathbf{x}_1) + u(\mathbf{x}_3) + u(\mathbf{x}_4) + u(\mathbf{x}_5) - 4u(\mathbf{x}_2)\right) \tag{150}$$

Finite Difference Stencils

TABLE 3.1 *Finite difference approximations for various differentiations*

Differentiation	Finite difference approximation	Molecules
$\left.\frac{dw}{dx}\right\|_i$	$\frac{w_{i+1} - w_{i-1}}{2h}$	
$\left.\frac{d^2w}{dx^2}\right\|_i$	$\frac{w_{i+1} - 2w_i + w_{i-1}}{h^2}$	
$\left.\frac{d^3w}{dx^3}\right\|_i$	$\frac{w_{i+2} - 2w_{i+1} + 2w_{i-1} - w_{i-2}}{2h^3}$	
$\left.\frac{d^4w}{dx^4}\right\|_i$	$\frac{w_{i+2} - 4w_{i+1} + 6w_i - 4w_{i-1} + w_{i-2}}{h^4}$	
$\nabla^2 w\|_{i,j}$	$\frac{-4w_{i,j} + w_{i+1,j} + w_{i,j+1} + w_{i-1,j} + w_{i,j-1}}{h^2}$	
$\nabla^4 w\|_{i,j}$	$[20w_{i,j} - 8(w_{i+1,j} + w_{i-1,j} + w_{i,j+1} + w_{i,j-1}) + 2(w_{i+1,j+1} + w_{i-1,j+1} + w_{i-1,j-1} + w_{i+1,j-1}) + w_{i+2,j} + w_{i-2,j} + w_{i,j+2} + w_{i,j-2}]/h^4$	

Source:Bathe's book, section 3.3.5.

Weighted Residual Method: Galerkin Method

- In Galerkin method weight functions are set to be the same as trial functions:

$$\boxed{\forall i \in \{1, 2, \ldots, n\} : \ \mathbf{w}_i = \boldsymbol{\phi}_i \ \Rightarrow [\mathbf{w}] = [\boldsymbol{\phi}]} \tag{151}$$

- The matrix equation $\mathbf{Ka} = \mathbf{F}$ components for Galerkin method are obtained from (132) and (133) by substituting $\boldsymbol{\phi}$ for $\mathbf{w}$:

$$\mathbf{K} = \int_{\mathcal{D}} [\boldsymbol{\phi}].[L_M(\boldsymbol{\phi})]^{\mathrm{T}} \, \mathrm{dv} - \int_{\partial \mathcal{D}_f} [\boldsymbol{\phi}].[L_f(\boldsymbol{\phi})]^{\mathrm{T}} \, \mathrm{ds} \tag{152a}$$

$$\mathbf{F} = \int_{\mathcal{D}} [\boldsymbol{\phi}].(\mathbf{r} - L_M(\boldsymbol{\phi}_p)) \, \mathrm{dv} + \int_{\partial \mathcal{D}_f} [\boldsymbol{\phi}].(L_f(\boldsymbol{\phi}_p) - \bar{\mathbf{f}}) \, \mathrm{ds} \tag{152b}$$

or alternatively the individual components are given,

$$K_{ij} = \int_{\mathcal{D}} \boldsymbol{\phi}_i L_M(\boldsymbol{\phi}_j) \, \mathrm{dv} - \int_{\partial \mathcal{D}_f} \boldsymbol{\phi}_i L_f(\boldsymbol{\phi}_j) \, \mathrm{ds} \tag{153a}$$

$$F_i = \int_{\mathcal{D}} \boldsymbol{\phi}_i (\mathbf{r} - L_M(\boldsymbol{\phi}_p)) \, \mathrm{dv} + \int_{\partial \mathcal{D}_f} \boldsymbol{\phi}_i (L_f(\boldsymbol{\phi}_p) - \bar{\mathbf{f}}) \, \mathrm{ds} \tag{153b}$$

Least Square Method

We want to obtain the discrete solution corresponding to the continuum least square method (51):

$$\begin{aligned} \text{Find } \mathbf{u} \in \mathcal{V} = \{\mathbf{v} \,|\mathbf{v} \in C^M(\mathcal{D}, L_u(\mathbf{u}) = \bar{\mathbf{u}}\} \text{ such that} \\ \int_{\mathcal{D}} \mathcal{R}_i^2(\mathbf{u})\,\mathrm{dv} + \int_{\partial\mathcal{D}_f} \mathcal{R}_f^2(\mathbf{u})\,\mathrm{ds} &= \\ \int_{\mathcal{D}} (L_M(\mathbf{u}) - \mathbf{r})^2\,\mathrm{dv} + \int_{\partial\mathcal{D}_f} (\bar{\mathbf{f}} - L_f(\mathbf{u}))^2\,\mathrm{ds} &= 0 \end{aligned}$$

By changing $\mathbf{u}$ to $\mathbf{u}^h \in \mathcal{V}^h$ and minimizing R^2 with respect to solution coefficients $\mathbf{a}$ instead of continuum condition $R^2 = 0$, we have

Find $\mathbf{u}^h \in \mathcal{V}^h$ such that (154)

$\forall \tilde{\mathbf{u}}^h \in \mathcal{V}^h : \quad R^2(\mathbf{u}^h) \leq R^2(\tilde{\mathbf{u}}^h)$ where

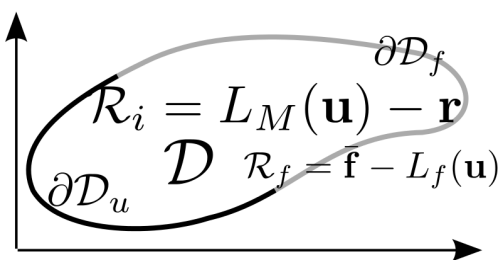
$$R^2(\tilde{\mathbf{u}}^h) = \int_{\mathcal{D}} \mathcal{R}_i^2(\tilde{\mathbf{u}}^h)\,\mathrm{dv} + \int_{\partial\mathcal{D}_f} \mathcal{R}_f^2(\tilde{\mathbf{u}}^h)\,\mathrm{ds} = \int_{\mathcal{D}} (L_M(\tilde{\mathbf{u}}^h) - \mathbf{r})^2\,\mathrm{dv} + \int_{\partial\mathcal{D}_f} (\bar{\mathbf{f}} - L_f(\tilde{\mathbf{u}}^h))^2\,\mathrm{ds}$$

Noting that $\mathcal{V}^h$ is an n-dimensional space (122) and $\mathbf{u}^h = \sum_{j=1}^n a_j \boldsymbol{\phi}_j + \boldsymbol{\phi}_p$ (117a) the minimum condition can be expressed as,

Find $[\mathbf{a}] \in \mathbb{R}^n$ such that (155)

$\forall [\tilde{\mathbf{a}}] \in \mathbb{R}^n : \quad R^2([\mathbf{a}]) \leq R^2([\tilde{\mathbf{a}}])$ where

$R^2([\tilde{\mathbf{a}}]) = R^2(\tilde{\mathbf{u}}^h)$ for $\tilde{\mathbf{u}}^h = [\boldsymbol{\phi}]^{\mathrm{T}}[\tilde{\mathbf{a}}] + \boldsymbol{\phi}_p$



Least Square Method

- The minimum condition for the solution $[\mathbf{a}](\mathbf{u}^h = [\boldsymbol{\phi}]^{\mathrm{T}}[\mathbf{a}] + \phi_p)$ in (155) can be expressed as,

$$[\mathbf{a}] \text{ is minimizer } \Rightarrow \frac{\partial R^2}{\partial a_i} = 0 \Rightarrow$$
$$\int_{\mathcal{D}} 2\frac{\partial L_M(\mathbf{u}^h)}{\partial a_i}(L_M(\mathbf{u}^h) - \mathbf{r})\,\mathrm{dv} + \int_{\partial\mathcal{D}_f}(-2)\frac{\partial L_f(\tilde{\mathbf{u}}^h)}{\partial a_i}(\bar{\mathbf{f}} - L_f(\tilde{\mathbf{u}}^h))\,\mathrm{ds} = 0 \tag{156}$$

- Noting the linearity of L_M and L_f and $[\mathbf{a}](\mathbf{u}^h = [\boldsymbol{\phi}]^{\mathrm{T}}[\mathbf{a}] + \phi_p)$ we observe,

$$L_M(\mathbf{u}^h) = a_i L_M(\boldsymbol{\phi}_i) + \boldsymbol{\phi}_p \Rightarrow \frac{\partial L_M(\mathbf{u}^h)}{\partial a_i} = L_M(\boldsymbol{\phi}_i) \tag{157a}$$
$$L_f(\mathbf{u}^h) = a_i L_f(\boldsymbol{\phi}_i) + \boldsymbol{\phi}_p \Rightarrow \frac{\partial L_f(\mathbf{u}^h)}{\partial a_i} = L_f(\boldsymbol{\phi}_i) \tag{157b}$$

Least Square Method

- Equations (156) and (157) yield,

$$\forall \boldsymbol{\phi}_i \in \mathcal{V}^h : \int_{\mathcal{D}} L_M(\boldsymbol{\phi}_i) \underbrace{(L_M(\mathbf{u}^h) - \mathbf{r})}_{\mathcal{R}_i} \, \mathrm{dv} + \int_{\partial \mathcal{D}_f} (-L_f(\boldsymbol{\phi}_i)) \underbrace{(\bar{\mathbf{f}} - L_f(\tilde{\mathbf{u}}^h))}_{\mathcal{R}_f} \, \mathrm{ds} = \mathbf{0} \tag{158}$$

- In comparison to (126) for the general statement of weighted residual methods we observe,

$$\forall \mathbf{w} \in \mathcal{W}^h : \int_{\mathcal{D}} \mathbf{w}.\mathcal{R}_i(\mathbf{u}^h) \, \mathrm{dv} + \int_{\partial \mathcal{D}_f} \mathbf{w}^f.\mathcal{R}_f(\mathbf{u}^h) \, \mathrm{ds} = \mathbf{0}$$

- Discrete Least Square problem for linear differential operators L_M and L_f is equivalent to a discrete weighted residual statement with the weight functions:

Weight functions corresponding to Least Square Method

$$\mathbf{w} = L_M(\boldsymbol{\phi}) \tag{159a}$$

$$\mathbf{w}^f = (-L_f(\boldsymbol{\phi})) \tag{159b}$$

Least Square Method

- According to (159), discrete least square method corresponds to a weighted residual method with particular weight functions given therein. Accordingly, (132) and (133) take the form:

$$\mathbf{K} = \int_{\mathcal{D}} [L_M(\boldsymbol{\phi})].[L_M(\boldsymbol{\phi})]^{\mathrm{T}} \,\mathrm{dv} + \int_{\partial\mathcal{D}_f} [L_f(\boldsymbol{\phi})].[L_f(\boldsymbol{\phi})]^{\mathrm{T}} \,\mathrm{ds} \tag{160a}$$

$$\mathbf{F} = \int_{\mathcal{D}} [L_M(\boldsymbol{\phi})].(\mathbf{r} - L_M(\boldsymbol{\phi}_p)) \,\mathrm{dv} - \int_{\partial\mathcal{D}_f} [L_f(\boldsymbol{\phi})].(L_f(\boldsymbol{\phi}_p) - \bar{\mathbf{f}}) \,\mathrm{ds} \tag{160b}$$

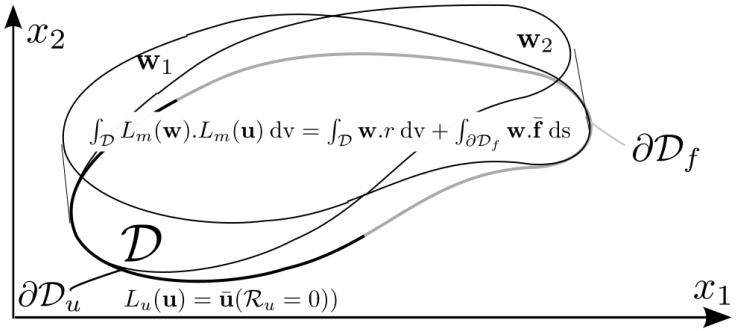
  or alternatively the individual components are given,

$$K_{ij} = \int_{\mathcal{D}} L_M(\boldsymbol{\phi}_i) L_M(\boldsymbol{\phi}_j) \,\mathrm{dv} + \int_{\partial\mathcal{D}_f} L_f(\boldsymbol{\phi}_i) L_f(\boldsymbol{\phi}_j) \,\mathrm{ds} \tag{161a}$$

$$F_i = \int_{\mathcal{D}} L_M(\boldsymbol{\phi}_i)(\mathbf{r} - L_M(\boldsymbol{\phi}_p)) \,\mathrm{dv} - \int_{\partial\mathcal{D}_f} L_f(\boldsymbol{\phi}_i)(L_f(\boldsymbol{\phi}_p) - \bar{\mathbf{f}}) \,\mathrm{ds} \tag{161b}$$

- The matrix $\mathbf{K}$ is always symmetric for least square method
- **Optional:** While all three variations of the continuum least square method (50), (51), and (52) can be used to form the discrete least square method, (52) (PDE residual only) is often used for presentation purposes. For other cases, care should be taken to ensure that all integrals have the same physical dimension.

Discrete Weak Statement

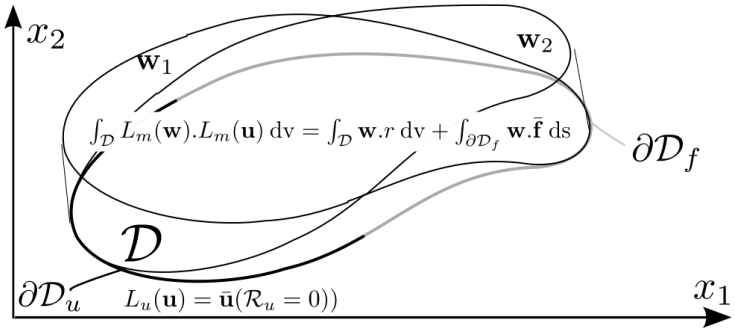


- Weak statement is basically a weighted residual statement where the derivative orders on the solution and weight function has been distributed to obtain a (more) balanced derivative order between the two.
- The advantage of the weak statement is the enlargement of the allowable solution functions.
- As will be shown shortly, this function space enlargement has immense practical importance in the design of finite element method.
- While the approach from balance law $\rightarrow$ strong form $\rightarrow$ weak form is quite cumbersome, we observed that the energy method provides a very direct way to obtain the weak statement.
- According to (74) the continuum weak statement is,

$$\text{Find } \mathbf{u} \in \mathcal{V} \text{ such that } \forall \mathbf{w} \in \mathcal{W} \int_{\mathcal{D}} L_m^w(\mathbf{w}) L_m(\mathbf{u})\,\mathrm{dv} = \int_{\mathcal{D}} \mathbf{w}.\mathbf{r}\,\mathrm{dv} + \int_{\partial\mathcal{D}_f} \mathbf{w}.\bar{\mathbf{f}}\,\mathrm{ds}$$

where L_m^w and L_m are differential operators acting on weight and solution functions.

Discrete Weak Statement



- The discrete weak statement is obtained in the same fashion that we derived the discrete weighted residual statement from continuum statement (as mentioned weak statement can be considered as a manipulated weighted residual statement).
- The same trial function and weight functions spaces (122) and (123) are used for the discrete statement. The discrete weak form reads as:

$$\text{Find } \mathbf{u}^h \in \mathcal{V}^h \text{ such that } \forall \mathbf{w} \in \mathcal{W}^h \int_{\mathcal{D}} L_m^w(\mathbf{w}) L_m(\mathbf{u}^h)\,\mathrm{dv} = \int_{\mathcal{D}} \mathbf{w}.\mathbf{r}\,\mathrm{dv} + \int_{\partial\mathcal{D}_f} \mathbf{w}.\bar{\mathbf{f}}\,\mathrm{ds} \tag{162}$$

- For linear L_m (L_m^w is always linear) by using the expansion $\mathbf{u}^h = \sum_{j=1}^{n} a_j \boldsymbol{\phi}_j + \boldsymbol{\phi}_p$ (117a) and $\mathbf{w} = [\mathbf{w}_1, \mathbf{w}_2, \ldots, \mathbf{w}_n]^{\mathrm{T}}$ we can derive the following linear system:

$$\int_{\mathcal{D}} L_m^w([\mathbf{w}]) L_m([\boldsymbol{\phi}]^{\mathrm{T}}[\mathbf{a}] + \boldsymbol{\phi}_p)\,\mathrm{dv} = \int_{\mathcal{D}} [\mathbf{w}].\mathbf{r}\,\mathrm{dv} + \int_{\partial\mathcal{D}_f} [\mathbf{w}].\bar{\mathbf{f}}\,\mathrm{ds} \;\Rightarrow \tag{163}$$

$$\left\{\int_{\mathcal{D}} L_m^w([\mathbf{w}]) L_m([\boldsymbol{\phi}])^{\mathrm{T}}\,\mathrm{dv}\right\}[\mathbf{a}] = \int_{\mathcal{D}} \{[\mathbf{w}].\mathbf{r} - L_m^w([\mathbf{w}]).L_m(\boldsymbol{\phi}_p)\}\,\mathrm{dv} + \int_{\partial\mathcal{D}_f} [\mathbf{w}].\bar{\mathbf{f}}\,\mathrm{ds}$$

Discrete Weak Statement

- From (163) the matrix equations for $\mathbf{Ka} = \mathbf{F}$ (125) are (*cf.* (132) and (133) for comparison):

$$\mathbf{K} = \int_{\mathcal{D}} L_m^w([\mathbf{w}]).[L_m(\boldsymbol{\phi})]^{\mathrm{T}} \,\mathrm{dv} \tag{164a}$$

$$\mathbf{F} = \int_{\mathcal{D}} \{[\mathbf{w}].\mathbf{r} - L_m^w([\mathbf{w}]).L_m(\boldsymbol{\phi}_p\} \,\mathrm{dv} + \int_{\partial\mathcal{D}_f} [\mathbf{w}].\bar{\mathbf{f}} \,\mathrm{ds} \tag{164b}$$

or alternatively the individual components are given,

$$K_{ij} = \int_{\mathcal{D}} L_m^w(\mathbf{w}_i).L_m(\boldsymbol{\phi}_j) \,\mathrm{dv} \tag{165a}$$

$$F_i = \int_{\mathcal{D}} \{\mathbf{w}_i.\mathbf{r} - L_m^w(\mathbf{w}_i).L_m(\boldsymbol{\phi}_p\} \,\mathrm{dv} + \int_{\partial\mathcal{D}_f} \mathbf{w}_i.\bar{\mathbf{f}} \,\mathrm{ds} \tag{165b}$$

Discrete Weak Statement: Self-adjoint property

- Self-adjoint operators are prevalent in physics and mathematics. They convey commutative property of a binary operator. For example, for the weak statement the self-adjoint property of the problem translates to:

$$\forall \mathbf{w}, \mathbf{v} \in \mathcal{V}^h : \int_{\mathcal{D}} L_m^w(\mathbf{w}).L_m(\mathbf{v}) \, \mathrm{dv} = \int_{\mathcal{D}} L_m^w(\mathbf{v}).L_m(\mathbf{w}) \, \mathrm{dv} \tag{166}$$

- We want to investigate what self-adjointness translates to in discrete setting.
- We consider a Galerkin weak statement, that is weight functions are equal to trial functions: $\mathbf{w}_i = \boldsymbol{\phi}_i$ (so $\mathcal{W}^h = \mathcal{V}^h$). Then from (164) and (165) linear matrix equation components become,

$$\mathbf{K} = \int_{\mathcal{D}} L_m^w([\boldsymbol{\phi}]).[L_m(\boldsymbol{\phi})]^{\mathrm{T}} \, \mathrm{dv} \tag{167a}$$

$$\mathbf{F} = \int_{\mathcal{D}} \{[\boldsymbol{\phi}].\mathbf{r} - L_m^w([\boldsymbol{\phi}]).L_m(\boldsymbol{\phi}_p\} \, \mathrm{dv} + \int_{\partial \mathcal{D}_f} [\boldsymbol{\phi}].\bar{\mathbf{f}} \, \mathrm{ds} \tag{167b}$$

or alternatively the individual components are given,

$$K_{ij} = \int_{\mathcal{D}} L_m^w(\boldsymbol{\phi}_i).L_m(\boldsymbol{\phi}_j) \, \mathrm{dv} \tag{168a}$$

$$F_i = \int_{\mathcal{D}} \{\boldsymbol{\phi}_i.\mathbf{r} - L_m^w(\boldsymbol{\phi}_i).L_m(\boldsymbol{\phi}_p\} \, \mathrm{dv} + \int_{\partial \mathcal{D}_f} \boldsymbol{\phi}_i.\bar{\mathbf{f}} \, \mathrm{ds} \tag{168b}$$

Discrete Weak Statement: Self-adjoint property

- If the problem is self adjoint from (166) and (168a) we observe:

$$K_{ij} = \int_{\mathcal{D}} L_m^w(\boldsymbol{\phi}_i).L_m(\boldsymbol{\phi}_j)\,\mathrm{dv} = \int_{\mathcal{D}} L_m^w(\boldsymbol{\phi}_j).L_m(\boldsymbol{\phi}_i)\,\mathrm{dv} = K_{ji} \quad \Rightarrow$$

Self adjoint property implications in discrete setting

For a self-adjoint problem, the discrete matrix $\mathbf{K}$ is symmetric

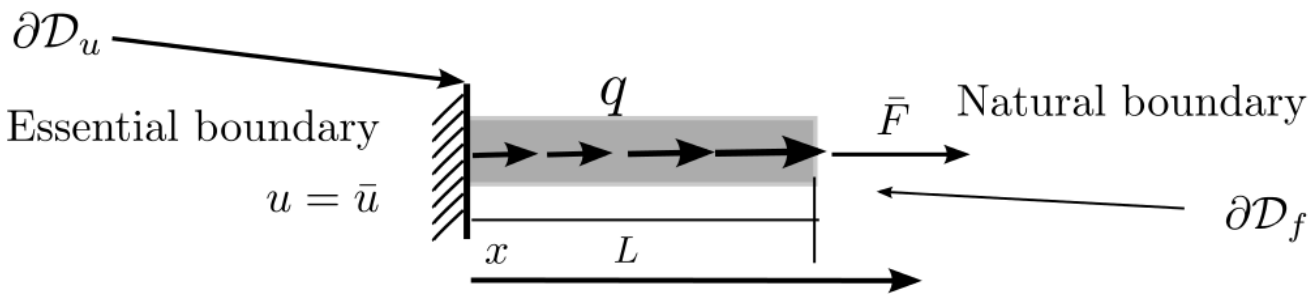
- Recalling the matrix $\mathbf{K}$ components for Least Square method (161a), how does this results compare to least square method:

$$K_{ij} = \int_{\mathcal{D}} L_M(\boldsymbol{\phi}_i)L_M(\boldsymbol{\phi}_j)\,\mathrm{dv} + \int_{\partial\mathcal{D}_f} L_f(\boldsymbol{\phi}_i)L_f(\boldsymbol{\phi}_j)\,\mathrm{ds} \quad \text{for least square method}$$

  While least square method **always** generates symmetric matrices, symmetry only occurs for self adjoint problems in weak statement with Galerkin weight option

- The self adjoint property can also be checked on the weighted residual statement, but it is easier to check this property in the weak statement because the weight and trial functions take a similar form.
- What are the benefits of a symmetric matrix in discrete setting?

Example: Self-adjoint problems



- Consider the 1D solid bar example shown in the figure. The residuals for this problem are (essential boundary conditions are strongly satisfied.

$$\mathcal{R}_i(x) = L_M(u(x)) - r(x) = \frac{\mathrm{d}\overbrace{F(x)}^{A\sigma(u(x))}}{\mathrm{d}x} + q(x) \quad \Rightarrow L_M(u(x)) = \frac{\mathrm{d}}{\mathrm{d}x}\left(EA\frac{\mathrm{d}u(x)}{\mathrm{d}x}\right)$$

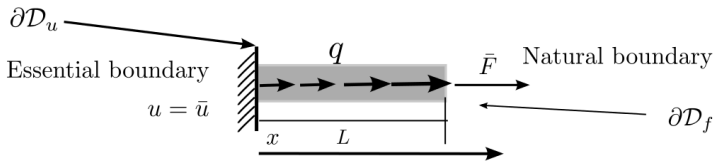
$$\mathcal{R}_f(x) = \bar{F} - F(x) = \bar{F} - A\sigma(u(x)) = \bar{F} - AE\frac{\mathrm{d}u(x)}{\mathrm{d}x}$$

- The weighted residual statement for this problem is,

$$\text{Find } u \in \mathcal{V}, \text{ such that } \forall w \in \mathcal{W}: \int_0^L \mathcal{R}_i(x)\,\mathrm{d}x + w(x).(\bar{F} - F(x))|_{x=2} = 0$$

Example: Self-adjoint problems

- We obtain the weak statement for this problem (by one integration by part) to assess if the problem is self adjoint or not (this is equation (73) we directly derived from energy method):



$$\text{Find } u \in \mathcal{V} = \{v \in C^1([0,L]) \mid v(0) = \bar{u}\}, \text{ such that,}$$

$$\forall w \in \mathcal{W} = \{v \in C^1([0,L]) \mid v(0) = 0\}$$

$$\int_0^L \underbrace{w'(x)}_{\epsilon(w(x))} \underbrace{AEu'(x)}_{F(u(x))=A\sigma(u(x))} \, \mathrm{d}x = \int_0^L w(x)q(x)\,\mathrm{d}x + w(L)\bar{F}$$

- That is,

$$L_m^w(w) = w'(x) = \epsilon(w(x)) \tag{169a}$$

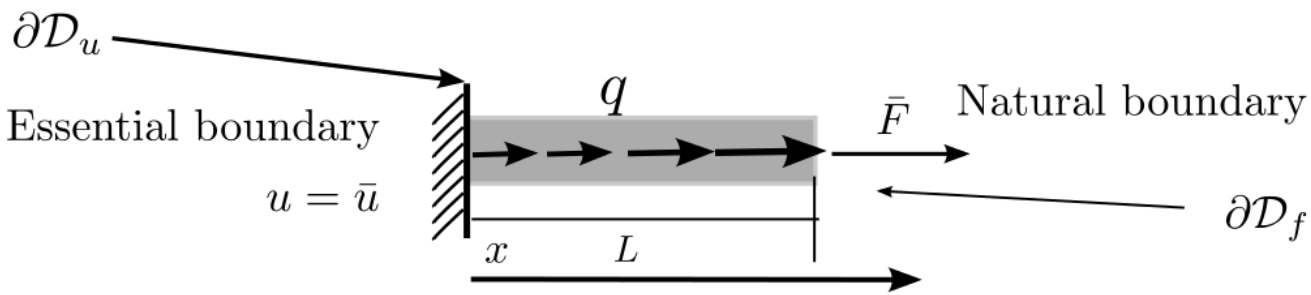
$$L_m(u) = AEu'(x) = A\sigma(u(x)) = F(u(x)) \tag{169b}$$

- It is easy to verify (166):

$$\int_{\mathcal{D}} L_m^w(w).L_m(\mathbf{u})\,\mathrm{dv} = \int_0^L w'(x).AEu'(x)\,\mathrm{d}x$$

$$= \int_0^L u'(x).AEw'(x)\,\mathrm{d}x = \int_{\mathcal{D}} L_m^w(u).L_m(\mathbf{w})\,\mathrm{dv}$$

Example: Self-adjoint problems



- So, 1D bar example is self adjoint.
- It is easy to verify (168a) by using (169)

$$K_{ij} = \int_0^L \frac{\mathrm{d}u_i}{\mathrm{d}x} . AE \frac{\mathrm{d}u_j}{\mathrm{d}x} \, \mathrm{d}x = \int_0^L \frac{\mathrm{d}u_j}{\mathrm{d}x} . AE \frac{\mathrm{d}u_i}{\mathrm{d}x} \, \mathrm{d}x = K_{ji}$$

- For elastostatics (linear or nonlinear) the self adjoint property well stemming from a more general property:

$$\epsilon(\mathbf{w}) : \sigma(\mathbf{u}) = \epsilon(\mathbf{u}) : \sigma(\mathbf{w})$$

  which is equivalent to the 1D case where we had:

$$\epsilon(w)F(u) = \frac{\mathrm{d}w}{\mathrm{d}x} AE \frac{\mathrm{d}u}{\mathrm{d}x} = \frac{\mathrm{d}u}{\mathrm{d}x} AE \frac{\mathrm{d}w}{\mathrm{d}x} = \epsilon(\mathbf{u})F(\mathbf{w})$$

Example: Self-adjoint problems

- As an example of a problem that is not self-adjoint consider the following boundary value problem for $u \in \mathcal{D} = [0\ 1]$:

$$\begin{cases} \frac{\mathrm{d}u}{\mathrm{d}x} - r(x) = 0 & \text{Differential equation} \\ u(0) = u(1) = 0 & \text{Essential boundary condition} \end{cases} \tag{170}$$

- The weighted residual statement for this problem is,

$$\text{Find } u \in \mathcal{V} = \{v \in C^1([0,1]) \mid v(0) = v(1) = 0\}, \text{ such that,}$$
$$\forall w \in \mathcal{W} = \{v \in C^0([0,1]) \mid v(0) = v(1) = 0\}$$
$$\int_0^1 w(x).\underbrace{(\frac{\mathrm{d}u(x)}{\mathrm{d}x} - r(x))}_{\mathcal{R}_i(x)}\,\mathrm{d}x = 0$$

- For the problem to be self adjoint the bilinear form (a linear function in each of its arguments):

$$a(u,w) = \int_0^1 w(x).\frac{\mathrm{d}u(x)}{\mathrm{d}x}\,\mathrm{d}x$$

should be commutative. However, It is clear that $a(u,w) \neq a(w,u)$ (for example consider $w(x) = 1, u(x) = x$).

- **Optional** Self adjoint property not only has many computational advantages (half the storage and faster solution algorithm for symmetric matrices) but also has several physical interpretations and facilitates many existence and uniqueness proofs.

Energy Method, approach A, Ritz method

We want to obtain the discrete solution corresponding to the continuum energy method (80):

$$\begin{aligned} &\text{Find } \mathbf{u} \in \mathcal{V} = \{\mathbf{v} \,|\mathbf{v} \in C^m(\mathcal{D}), L_u(\mathbf{u}) = \bar{\mathbf{u}}\} \text{ such that} \\ &\forall \tilde{\mathbf{u}} \in \mathcal{V}: \quad \boxed{\Pi(\mathbf{u}) \leq \Pi(\tilde{\mathbf{u}})} \quad \text{or alternatively} \\ \forall\, \delta\mathbf{u} \in \mathcal{V}_0 = \{\mathbf{v} \,|\mathbf{v} \in C^m(\mathcal{D}, L_u(\mathbf{u}) = 0\}: &\quad \Pi(\mathbf{u}) \leq \Pi(\mathbf{u} + \delta\mathbf{u}) \end{aligned} \tag{171}$$

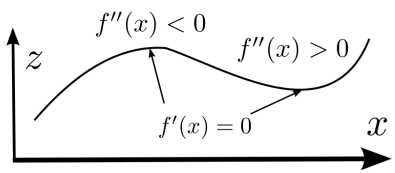
There are two paths to discretize the energy method.

1. **Ritz Method**: Direct discretization of the continuum energy minimization principle: By changing $\mathbf{u}$ to $\mathbf{u}^h \in \mathcal{V}^h$ and minimizing Π with respect to solution coefficients $\mathbf{a}$ instead of continuum condition $\Pi(\mathbf{u}) \leq \Pi(\tilde{\mathbf{u}})$ for $\mathbf{u}, \tilde{\mathbf{u}} \in \mathcal{V}$ (infinite dimensional space) we obtain,

$$\begin{aligned} &\text{Find } \mathbf{u}^h \in \mathcal{V}^h \text{ such that} \\ &\forall \tilde{\mathbf{u}}^h \in \mathcal{V}^h: \quad \Pi(\mathbf{u}^h) \leq \Pi(\mathbf{u}) \end{aligned} \tag{172}$$

Noting that $\mathcal{V}^h$ is an n-dimensional space (122) and $\mathbf{u}^h = \sum_{j=1}^n a_j \boldsymbol{\phi}_j + \boldsymbol{\phi}_p$ (117a) the minimum condition can be expressed as,

$$\begin{aligned} &\text{Find } [\mathbf{a}] \in \mathbb{R}^n \text{ such that} \\ &\forall [\tilde{\mathbf{a}}] \in \mathbb{R}^n: \quad \Pi([\mathbf{a}]) \leq \Pi([\tilde{\mathbf{a}}]) \text{ where} \\ &\Pi([\tilde{\mathbf{a}}]) = \Pi(\tilde{\mathbf{u}}^h) \text{ for } \tilde{\mathbf{u}}^h = [\boldsymbol{\phi}]^{\mathrm{T}}[\tilde{\mathbf{a}}] + \boldsymbol{\phi}_p \end{aligned} \tag{173}$$



Necessary optimality condition $\nabla\Pi([\mathbf{a}]) = 0$ (gradient $= 0$) for Π at $\mathbf{a} \Rightarrow$:

$$\boxed{\nabla\Pi(\mathbf{a}) = 0 \;\Rightarrow \frac{\partial \Pi}{\partial a_i}(\mathbf{a}) = 0 \quad \text{Ritz method minimizes } \Pi \text{ of the discrete system}} \tag{174}$$

Energy Method, approach B, weak formulation

2. **Weak method**: Minimizing the continuum problem then discretizing: In the second approach we proceed by first minimizing the continuum statement. From (96)

$$\Pi = \Pi(y, y', \ldots, y^{(n)}) \quad \Rightarrow \quad \delta\Pi = \frac{\partial \Pi}{\partial y}\delta y + \frac{\partial \Pi}{\partial y'}\delta y' + \cdots + \frac{\partial \Pi}{\partial y^{(n)}}\delta y^{(n)}. \tag{175}$$

the continuum energy method would directly result in a weak statement of the form:

$$\text{Find } \mathbf{u} \in \mathcal{V} : \ \forall \mathbf{w} \in \mathcal{V}_0$$
$$\int_{\mathcal{D}} L_m^w(\mathbf{w}) L_m(\mathbf{u}) \, \mathrm{dv} = \int_{\mathcal{D}} \mathbf{w}.\mathbf{r} \, \mathrm{dv} + \int_{\partial \mathcal{D}_f} \mathbf{w}.\bar{\mathbf{f}} \, \mathrm{ds}$$

where $\mathbf{w} = \delta\mathbf{u}$ stands for the variation of the solution $\mathbf{u}$ and $\mathcal{V}, \mathcal{V}_0$ are given in (171). We already discussed the solution of a discrete weak problem (for example refer to (162) and (163)). The linear system matrix $\mathbf{K}$ and force vectors $\mathbf{F}$ were given in (164):

$$\mathbf{K} = \int_{\mathcal{D}} L_m^w([\mathbf{w}]).[L_m(\boldsymbol{\phi})]^{\mathrm{T}} \, \mathrm{dv} \tag{176}$$
$$\mathbf{F} = \int_{\mathcal{D}} \{[\mathbf{w}].\mathbf{r} - L_m^w([\mathbf{w}]).L_m(\boldsymbol{\phi}_p\} \, \mathrm{dv} + \int_{\partial \mathcal{D}_f} [\mathbf{w}].\bar{\mathbf{f}} \, \mathrm{ds}$$

Comparison of two approaches for discretizing the energy method

- In the second approach (weak form obtained from energy functional) as can be seen in (176), we have complete freedom to choose any set of weight functions (for example subdomain, collocation, or Galerkin options).
- While in the first approach (Ritz method) we do not directly deal with weight functions, generally it corresponds to the weak system in (175) with **Galerkin** weight functions, that is $\mathbf{w} = \boldsymbol{\phi}$ and (176) simplifies to

$$\mathbf{K} = \int_{\mathcal{D}} L_m^w([\boldsymbol{\phi}]).[L_m(\boldsymbol{\phi})]^{\mathrm{T}} \,\mathrm{dv}$$

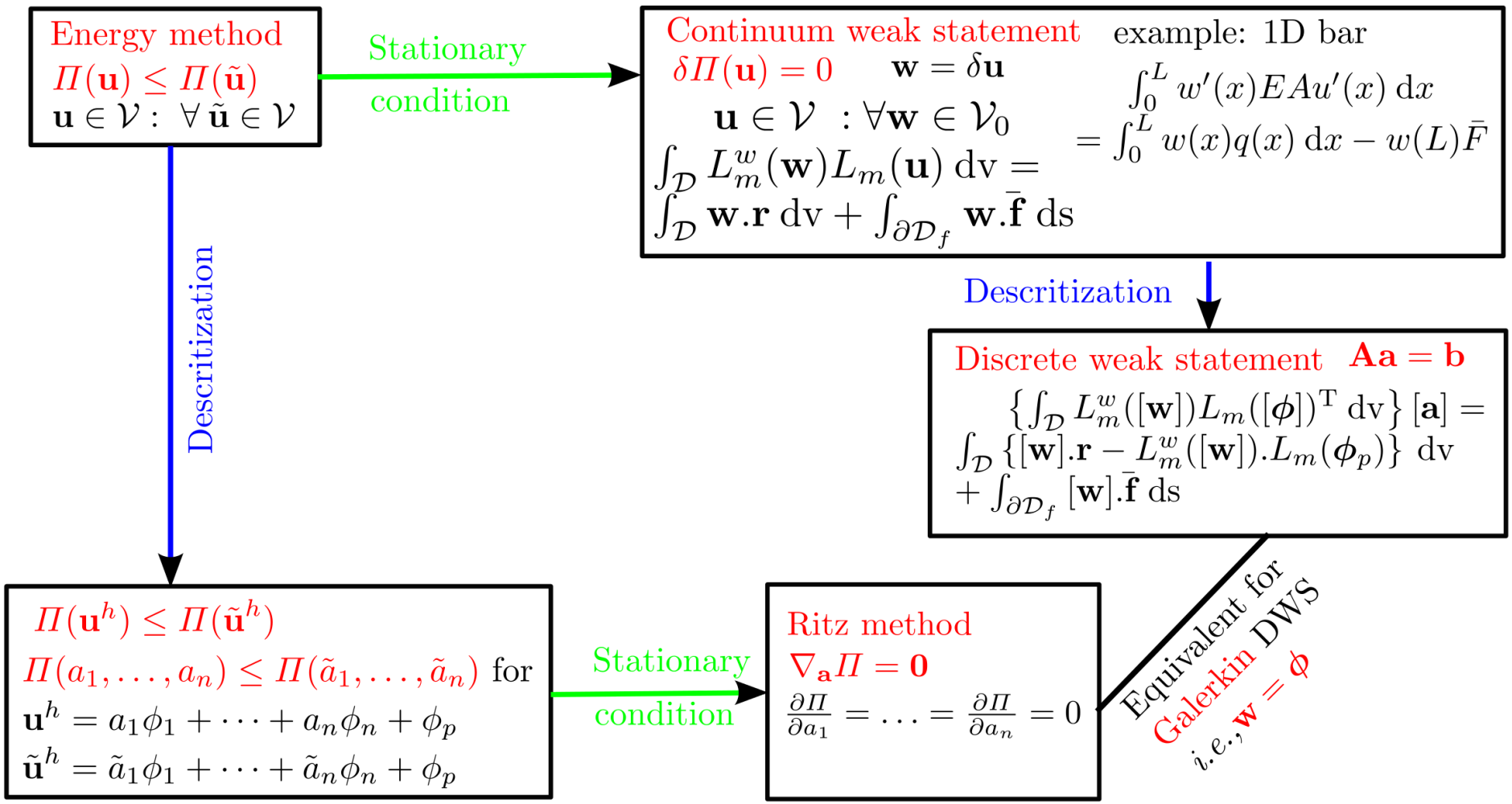
$$\mathbf{F} = \int_{\mathcal{D}} \{[\boldsymbol{\phi}].\mathbf{r} - L_m^w([\boldsymbol{\phi}]).L_m(\boldsymbol{\phi}_p\} \,\mathrm{dv} + \int_{\partial\mathcal{D}_f} [\boldsymbol{\phi}].\bar{\mathbf{f}} \,\mathrm{ds}$$

- This is very similar to least square method which also did not directly include weight functions but we concluded that for linear differential operators L_M and L_f it corresponded to weight functions of the form $\mathbf{w} = L_M(\boldsymbol{\phi})$ and $\mathbf{w}^f = -L_f(\boldsymbol{\phi})$.

Summary

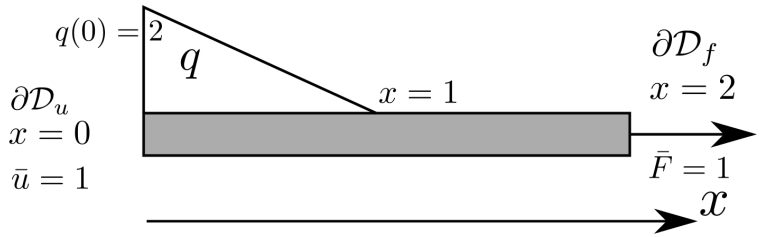
Basically in the first approach (Ritz method) we first discretize then optimize while in the second approach (weak statement) we first optimize then discretize. The commutation of these operations are shown in the next slide.

Relation between Energy Method and Weak Statement



Numerical Example

Problem set up



- For this one problem all the quantities are displayed as scalars.
- Consider the 1D bar example shown with the following boundary value problem:

$$\begin{cases} \frac{\mathrm{d}\sigma(u(x))}{\mathrm{d}x} + q(x) = \frac{\mathrm{d}}{\mathrm{d}x}\left(EA\frac{\mathrm{d}u(x)}{\mathrm{d}x}\right) + q(x) = 0 & \text{Strong form} \\ u(0) = \bar{u} = 1 & \text{Essential BC on } \partial\mathcal{D}_u = \{0\} \\ F(2) = EA\frac{\mathrm{d}u(x)}{\mathrm{d}x} = \bar{F} = 1 & \text{Natural BC on } \partial\mathcal{D}_f = \{2\} \end{cases} \tag{177}$$

- Material and load properties are,

$$q(x) = \begin{cases} 2 - 2x & x < 1 \\ 0 & 1 \leq x \leq 2 \end{cases} \qquad E(x) = 1, \quad A(x) = 1, \quad \bar{u} = 1, \quad \bar{F} = 1$$

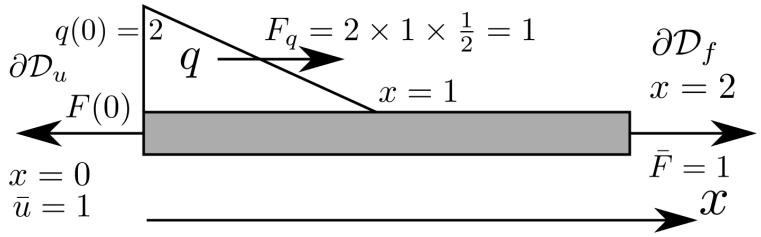
- According to (177), differential operators and r are:

$$L_M(.) = \frac{\mathrm{d}}{\mathrm{d}x}\left(EA\frac{\mathrm{d}(.)}{\mathrm{d}x}\right) \tag{178a}$$

$$L_f(.) = F(.) = EA\frac{\mathrm{d}(.)}{\mathrm{d}x} \tag{178b}$$

$$r(x) = -q(x) \tag{178c}$$

Exact solution



- Axial force at $x = 0$ is obtained from:

$$-F(0) + F_q + \bar{F} = 0 \quad \Rightarrow F(0) = 2 \times 1 \times \frac{1}{2} + 1 \quad \Rightarrow F(0) = 2$$

- For $x < 1$ we have:

$$\left(\frac{\mathrm{d}EA}{\mathrm{d}x}\frac{\mathrm{d}u}{\mathrm{d}x}\right) + q(x) = \frac{\mathrm{d}^2u}{\mathrm{d}x^2} + 2 - 2x = 0 \quad F(x) = EA\frac{\mathrm{d}u}{\mathrm{d}x} = x^2 - 2x + F(0) \quad \Rightarrow$$

$$u(x) = \frac{x^3}{3} - x^2 + 2x + u(0) \quad \Rightarrow \quad u(x) = \frac{x^3}{3} - x^2 + 2x + 1$$

- From this equation we obtain $u(1) = \frac{7}{3}$ and $\frac{\mathrm{d}u}{\mathrm{d}x}(1) = 1$. For $x \leq x \leq 2$ we get:

$$\left(\frac{\mathrm{d}EA}{\mathrm{d}x}\frac{\mathrm{d}u}{\mathrm{d}x}\right) + q(x) = \frac{\mathrm{d}^2u}{\mathrm{d}x^2} + 0 = 0 \quad \Rightarrow \quad \frac{\mathrm{d}u}{\mathrm{d}x} = \frac{\mathrm{d}u}{\mathrm{d}x}(1) = 1 \quad \Rightarrow$$

$$u(x) = x - 1 + u(1) = x - 1 + = \frac{7}{3} \quad \Rightarrow \quad u(x) = x + \frac{4}{3}$$

- The exact solution can be summarized as,

$$u(x) = \begin{cases} \frac{x^3}{3} - x^2 + 2x + 1 & 0 \leq x \leq 1 \\ x + \frac{4}{3} & 1 < x \leq 2 \end{cases} \tag{179}$$

Weighted residual statement and function spaces

- According to derivative orders the weighted residual statement with function spaces is,

$$\text{Find } u \in \mathcal{V} = \{v |\, v \in C^2([0\ 2]), v(0) = \bar{u} = 1\} : \forall w \in \mathcal{W} = C^0([0\ 2])$$

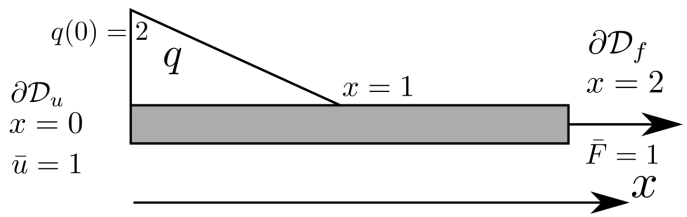
$$\int_0^2 w \left(\frac{\mathrm{d}^2 u}{\mathrm{d}x^2} + q(x) \right) \mathrm{d}x + w^f (1 - \frac{\mathrm{d}u}{\mathrm{d}x}(2)) = 0 \quad (\text{no need for } w(\partial\mathcal{D}_f = \{0\}) = 0) \tag{180}$$

- For the expansion of $u^h = \sum_{j=1}^n a_j \phi_j + \phi_p$ (117a) and choice of weight functions, according to (120) to (123) we need,

$$w_i \in \mathcal{W} \tag{181a}$$

$$\phi_p \in \mathcal{V} \tag{181b}$$

$$\phi_i \in \mathcal{V}_0 = \{v |\, v \in C^2([0\ 2]), v(0) = 0\} \tag{181c}$$



- So from (181b) $\phi_p(0)$ must be equal to 1. A natural choice for this function is,

$$\phi_p(x) = 1, \quad \forall x \in [0\ 2] \tag{182}$$

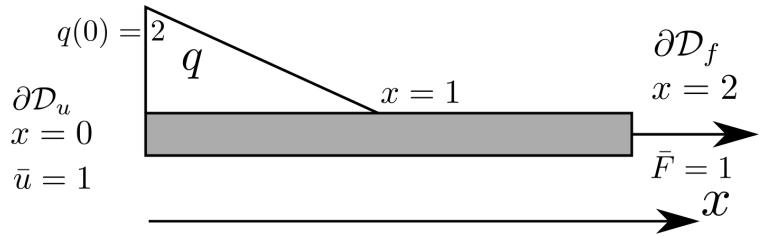
- Trial functions ϕ_i must satisfy the homogeneous version of the essential boundary condition:

$$\forall i \in \{1, 2, \ldots, n\}, \quad \phi_i(x = 0) = 0 \tag{183}$$

  For example $\phi = 1$ is not an acceptable trial functions as it violates this condition.
- Discrete function spaces are:

$$\mathcal{V}_0^h = \text{span}(\phi_1, \ldots, \phi_n), \quad \mathcal{V}^h = \mathcal{V}_0^h + \phi_p, \text{ and } \mathcal{W}^h = \text{span}(w_1, \ldots, w_n). \tag{184}$$

Matrix equation for discrete weighted residual method



- Plugging $u^h = [\boldsymbol{\phi}]^{\mathrm{T}}[\mathbf{a}] + \phi_p = a_j\phi_j + \phi_p$ (117a) into (180) we obtain

$$\mathbf{K} = \int_0^2 [\mathbf{w}].[\frac{\mathrm{d}^2}{\mathrm{d}x^2}(\boldsymbol{\phi})]^{\mathrm{T}}\,\mathrm{d}x - \{[\mathbf{w}^f].\frac{\mathrm{d}[\boldsymbol{\phi}]^{\mathrm{T}}}{\mathrm{d}x}\}|_{x=2} \tag{185a}$$

$$\mathbf{F} = -\int_0^1 [\mathbf{w}].q(x)\,\mathrm{d}x - [\mathbf{w}^f]_{x=2} \tag{185b}$$

or alternatively the individual components are given,

$$K_{ij} = \int_0^2 w_i\frac{\mathrm{d}^2}{\mathrm{d}x^2}(\phi_j)\,\mathrm{d}x - \{w_i^f.\frac{\mathrm{d}\phi_j}{\mathrm{d}x}\}|_{x=2} \tag{186a}$$

$$F_i = -\int_0^1 w_i.q(x)\,\mathrm{d}x - w_i^f|_{x=2} \tag{186b}$$

- Note that the upper limit for $\int_0^1 w_i q(x)$ is 1 as $q(x) = 0$ for $x > 1$ (*cf.* (178)).
- (185), (186) could also be obtained from (132) & (133) by using $L_M = \frac{\mathrm{d}^2}{\mathrm{d}x^2}$, $L_f = \frac{\mathrm{d}}{\mathrm{d}x}$, $r = -q$, $\phi_p = 1$, $\bar{F} = 1$, $\mathcal{D} = [0\ 2]$, $\partial\mathcal{D}_f = \{2\}$, and $w^f = w$ (*cf.* (178) , (178), & (182)).

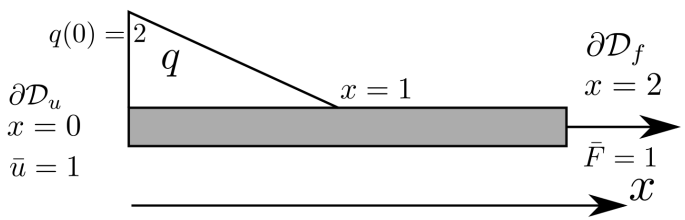
Weak statement and function spaces

- While the weighed residual statement (180) is very useful the high order regularity requirement on u^h ($M = 2$) can be limiting in practice. For example for Finite Element Hat Functions in 1D the trial functions do not possess a second derivative (even in distributional sense) as will be shown later.
- We Obtain the weak form by an integration by part on the weighted residual statement (or as shown before one can directly obtain the weak statement from energy principles):
- From (180) (with $w^f = w$)

$$\int_0^2 w\left(\frac{\mathrm{d}^2 u}{\mathrm{d}x^2} + q(x)\right)\,\mathrm{d}x + w(\bar{F} - \frac{\mathrm{d}u}{\mathrm{d}x}(2)) = 0 \;\Rightarrow$$

$$-\int_0^2 \frac{\mathrm{d}w}{\mathrm{d}x}\frac{\mathrm{d}u}{\mathrm{d}x}\,\mathrm{d}x + \left\{w(2)\frac{\mathrm{d}u}{\mathrm{d}x}(2) - w(0)\frac{\mathrm{d}u}{\mathrm{d}x}(0)\right\}$$

$$+\int_0^2 q(x)\,\mathrm{d}x + w(2)(\bar{F} - \frac{\mathrm{d}u}{\mathrm{d}x}(2)) = 0$$



- Noting that $w(0) = 0$ (strong satisfaction of homogeneous essential boundary condition for w is required in weak statement) we get:

$$\int_0^2 \frac{\mathrm{d}w}{\mathrm{d}x}\frac{\mathrm{d}u}{\mathrm{d}x}\,\mathrm{d}x = \int_0^2 w(x)q(x)\,\mathrm{d}x + w(2)\bar{F}$$

- The LHS is the virtual internal work and the RHS is the virtual external work.

Weak statement and function spaces

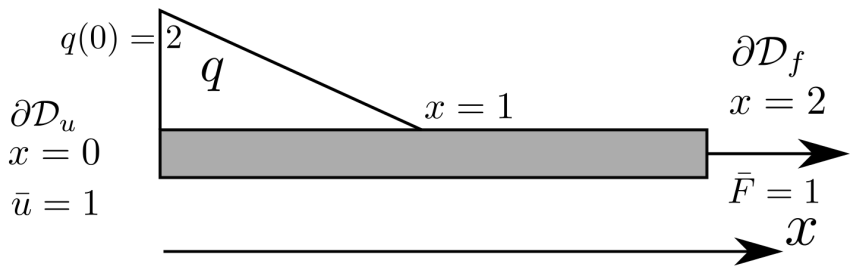
- Observing that maximum derivative orders for both ϕ and w is 1, considering the essential boundary conditions, $\bar{F} = 1$, and (178) our weak statement is,
  we obtain the weak statement:

Find $u \in \mathcal{V} = \{v |\, v \in C^1([0\ 2]), v(0) = \bar{u} = 1\}$:$\forall w \in \mathcal{V}_0 = \{v |\, v \in C^1([0\ 2]), v(0) = \bar{u} = 0\}$

$$\int_0^2 \frac{\mathrm{d}w}{\mathrm{d}x}\frac{\mathrm{d}u}{\mathrm{d}x}\,\mathrm{d}x = \int_0^1 w(x).q(x)\,\mathrm{d}x + w(2) \tag{187}$$

- For the expansion of $u^h = \sum_{j=1}^n a_j\phi_j + \phi_p$ (117a) and choice of weight functions, according to (120) to (123) we need,

$$w_i \in \mathcal{V}_0 \tag{188a}$$
$$\phi_p \in \mathcal{V} \tag{188b}$$
$$\phi_i \in \mathcal{V}_0 \tag{188c}$$



- Similar to weighted residual statement ϕ_p must satisfy the essential boundary condition and $\phi_p = 1$ satisfies $\phi_p(0) = 1$ (188b).
- Discrete function spaces are:

$$\mathcal{V}_0^h = \mathrm{span}(\phi_1, \ldots, \phi_n), \quad \mathcal{V}^h = \mathcal{V}_0^h + \phi_p, \text{ and } \mathcal{W}^h = \mathrm{span}(w_1, \ldots, w_n). \tag{189}$$

Optional: Weak statement: Discussion on function spaces

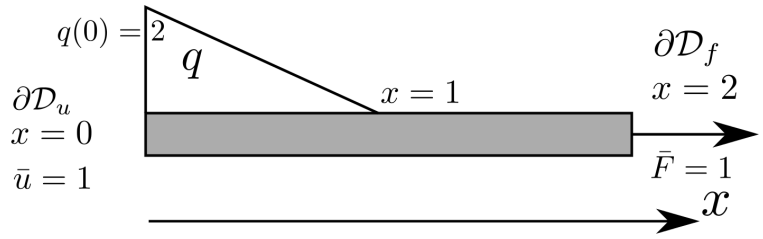


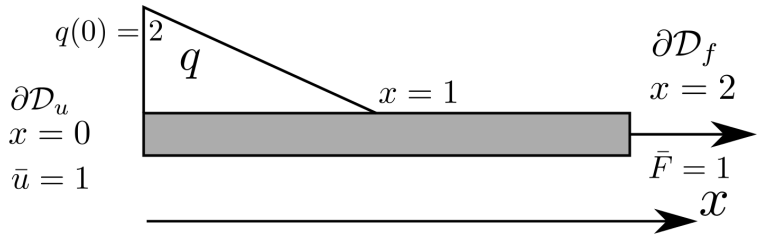
Figure: show an example picture

- Now **both** trial ϕ_i and weight w_i functions must satisfy the homogeneous version of the essential boundary condition:

$$\forall i \in \{1, 2, \ldots, n\}, \quad \phi_i = 0, \; w_i = 0 \tag{190}$$

- $\phi = 1$ is not an acceptable trial functions.
- $w = 1$ is not an acceptable weight functions (**So weight functions of subdomain method cannot be used in weak statement**)
- Also, reduction of the regularity of trial functions ϕ while expands the solution space (*e.g.*, Finite Element trial functions), it reduces weight function space. For example, collocation delta Dirac functions are not acceptable as $\frac{\mathrm{d}\delta(x)}{\mathrm{d}x}$ is not integrable even in distribution sense.

Matrix equation for discrete weak statement



- Plugging $u^h = [\boldsymbol{\phi}]^{\mathrm{T}}[\mathbf{a}] + \phi_p = a_j\phi_j + \phi_p$ (117a) into (187) we obtain

$$\mathbf{K} = \int_0^2 [\frac{\mathrm{d}}{\mathrm{d}x}(\mathbf{w})].[\frac{\mathrm{d}}{\mathrm{d}x}(\boldsymbol{\phi})]^{\mathrm{T}}\,\mathrm{d}x \tag{191a}$$

$$\mathbf{F} = \int_0^1 [\mathbf{w}].q(x)\,\mathrm{d}x + [\mathbf{w}]_{x=2} \tag{191b}$$

or alternatively the individual components are given,

$$K_{ij} = \int_0^2 \frac{\mathrm{d}}{\mathrm{d}x}(w_i)\frac{\mathrm{d}}{\mathrm{d}x}(\phi_j)\,\mathrm{d}x \tag{192a}$$

$$F_i = \int_0^1 w_i.q(x)\,\mathrm{d}x + w_i^f|_{x=2} \tag{192b}$$

- Note that these equations take a simpler form than (185) and (186).
- As before the upper limit for $\int_0^1 w_i q(x)$ is 1 as $q(x) = 0$ for $x > 1$ (*cf.* (178)).
- (191), (192) could also be obtained from (164) and (165) by using $L_m = \frac{\mathrm{d}}{\mathrm{d}x}$, $L_m^w = \frac{\mathrm{d}}{\mathrm{d}x}$, $r = q$, $\phi_p = 1$, $\bar{F} = 1$, $\mathcal{D} = [0\ 2]$, and $\partial\mathcal{D}_f = \{2\}$ (*cf.* (178) , (178), & (182)).

Energy method and function spaces

- As discussed before, if we first obtain the stationary conditions for the energy functional in continuum setting ($\delta \Pi = 0$) we obtain the continuum weak statement which can then be discretized with the procedure in previous slides.
- In this numerical example, we directly work with the Ritz method where we first discretize then obtain the stationary condition,
- From the equation of energy functional for a bar (88) we have,

$$\begin{aligned} \Pi(u) &= \int_0^L \frac{1}{2} EA \left(\frac{\mathrm{d}u}{\mathrm{d}x}\right)^2 \mathrm{d}x - \int_0^L u(x) q(x)\, \mathrm{d}x - u(L)\bar{F} \\ &= \int_0^2 \frac{1}{2} \left(\frac{\mathrm{d}u}{\mathrm{d}x}\right)^2 \mathrm{d}x - \int_0^1 u(x) q(x)\, \mathrm{d}x - u(2) \end{aligned} \tag{193}$$

- In Ritz method we directly discretize the energy functional (193) by using $u^h = a_j \phi_j + \phi_p$ (117a) to obtain,

$$\begin{aligned} \Pi(u^h) &= \Pi([\mathbf{a}]) = \Pi([\{a_1, a_2, \ldots, a_n\}]) \\ &= \int_0^2 \frac{1}{2} \left(\frac{\mathrm{d}\{a_j \phi_j + \phi_p\}(x)}{\mathrm{d}x}\right)^2 \mathrm{d}x - \int_0^1 \{a_j \phi_j + \phi_p\}(x) q(x)\, \mathrm{d}x - \{a_j \phi_j + \phi_p\}(2) \end{aligned} \tag{194}$$

- Now that we have the discrete energy function we apply the stationary condition to obtain the discrete solution,

$$\text{Find } \mathbf{a} \text{ such that } \nabla_{\mathbf{a}} . \Pi = \mathbf{0} \quad \Leftrightarrow \quad \frac{\partial \Pi}{\partial a_i} = 0 \tag{195}$$

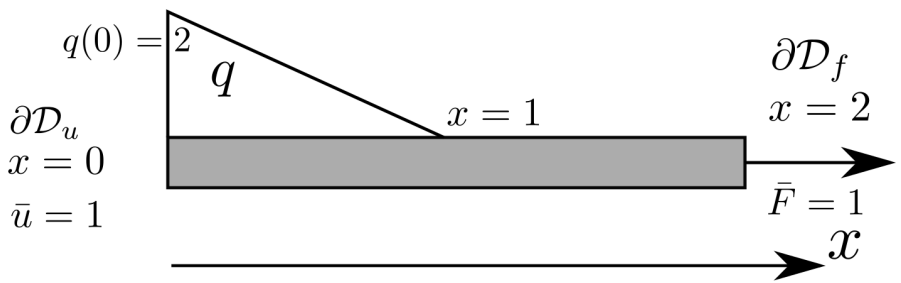
Overview of discrete solutions for the 1D bar Example

- We use various solution schemes,
  1. WRM: subdomain ($w = \chi_{\Omega_i}$, corresponds to discrete balance law)
  2. WRM: Collocation ($w = \delta(x - x_i)$, corresponds to discrete strong form)
  3. Finite Difference, corresponds to discrete strong form
  4. Least square = WRM with ($w = L_M(\phi), w^f = -L_f(\phi)$) = discrete LS
  5. WRM: Galerkin ($w = \phi$) for two forms of ϕ: one using trial functions in (196) and one of Finite Element hat-function type. The latter should be solved with the weak statement as finite element trial functions lack regularity for WRM.
  6. Discrete Energy Method (Ritz method) equivalent to Galerkin method.
- The trial functions are:

$$[\phi] = \{x\} \qquad n = 1 \qquad (196a)$$

$$[\phi] = \{x, x^2\} \qquad n = 2 \qquad (196b)$$

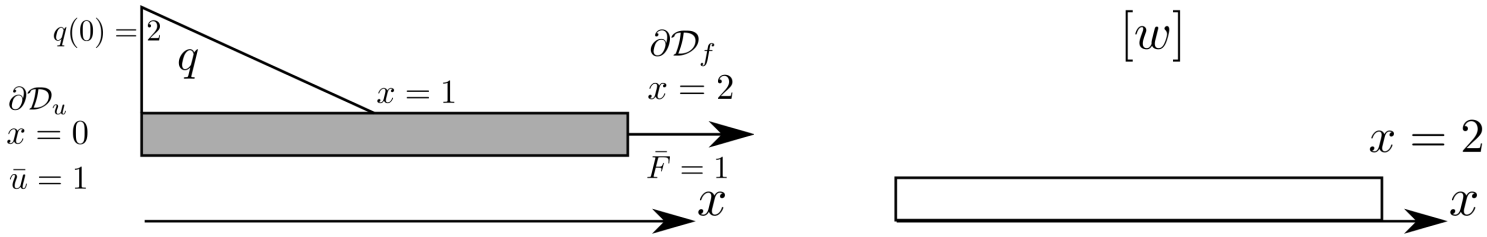
$$[\phi] = \{x, x^2, x^3\} \qquad n = 3 \qquad (196c)$$



for all the cases above except i) Finite Difference (which does not employ trial functions) and finite element type Galerkin method. For the latter we use 1, 2, and 3 FE hat functions (as will be shown later). We will choose 1, 2, and 3 point unknowns for Finite Difference method.

- We will compare the performance and attributes of these methods and within each group observe how increasing n improves the solution.

Bar example, $n = 1$, Subdomain method



- For $n = 1$ our trial function is $[\phi] = \{x\}$ (*cf.* (196a)).
- For subdomain method $n = 1$ weight function is $[w] = [w^f] = \chi_{[0\ 2]}$ (shown in figure).
- We plug $[\phi] = \{x\}$ and $[w] = \chi_{[0\ 2]}$ into (185) to obtain:

$$\mathbf{K} = \int_0^2 [\chi_{[0\ 2]}].\{\frac{\mathrm{d}^2}{\mathrm{d}x^2}[x]\}^{\mathrm{T}}\,\mathrm{d}x - \{[\chi_{[0\ 2]}].\left(\frac{\mathrm{d}[x]}{\mathrm{d}x}\right)^{\mathrm{T}}\}|_{x=2}$$

$$= \int_0^2 [1].[0]^{\mathrm{T}}\,\mathrm{d}x - \{[1].[1]^{\mathrm{T}}\}|_{x=2} = 0 - 1 = -1 \qquad (197a)$$

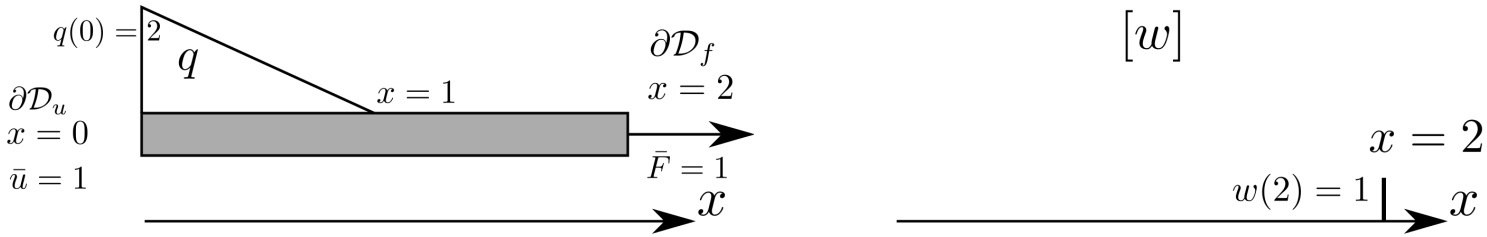
$$\mathbf{F} = -\int_0^1 [\chi_{[0\ 2]}].(2 - 2x)\,\mathrm{d}x - [\chi_{[0\ 2]}]_{x=2}$$

$$= -\int_0^1 [1].(2 - 2x)\,\mathrm{d}x - [\chi_1]_{x=2} = -2 + 1 - 1 = -2 \qquad (197b)$$

- From $\mathbf{Ka} = \mathbf{F}$ (125) and $u^h = a_j\phi_j + \phi_p$ (117a), and $\phi_p = 1$ (182):

$$\boxed{u^h_{S1} = 1 + 2x} \qquad (198)$$

Bar example, $n = 1$, Collocation method



- In collocation method ϕ_p often satisfies both essential and natural boundary conditions and ϕ_i satisfy both homogeneous essential and natural boundary conditions. However, as discussed before we can formulate it such that m collocation points are inside $\mathcal{D}$ and $n - m$ on $\partial\mathcal{D}_f$. Then for $j = 1, \ldots, m$ $\mathcal{R}_i(x_j) = 0$ and for $j = m + 1, n$ $\mathcal{R}_f(x_j) = 0$.
- For this particular problem if we choose $x_1 = 1$ (middle of the bar) we end up with $\mathbf{K} = [0]$ and $\mathbf{F} = [0]$ which clearly cannot be solved. Rather we choose a weight function that only gives us the boundary residual at $x = 2$. One options is,

$$w_1 = \begin{cases} 0 & 0 \leq x < 2 \\ 1 & x = 2 \end{cases}$$

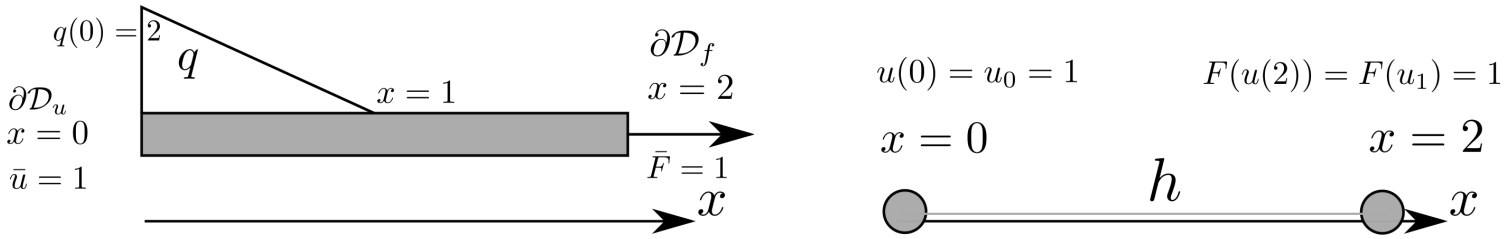
Note **Optional**: choosing $w_1 = \delta(x - 2)$ can also work with the caveat that in $\mathbf{w}_2\mathcal{R}_f(2)$ (weighted residual term on $\partial\mathcal{D}_f$) $\mathcal{R}_f(2)$ should be zero as $\delta(2 - 2) = \delta(0)$ is "infinite".

- With this $\mathbf{w}_1$ our one equation would be $(\int_0^2 w_1\mathcal{R}_i \, \mathrm{d}x + \{w_1\mathcal{R}_f\}|_{x=2} = \{\mathcal{R}_f\}_{x=2})$

$$\{\mathcal{R}_f\}_2 = 1 - \frac{\mathrm{d}u^h}{\mathrm{d}x} = 1 - a_1\frac{\mathrm{d}\phi_1}{\mathrm{d}x}(2) - \frac{\mathrm{d}\phi_p}{\mathrm{d}x}(2) = 1 - a_1\{1\}|_2 - \{0\}_2 = 1 - a_1 = 0 \Rightarrow a_1 = 1$$

$$\boxed{u^h_{C1} = 1 + x} \tag{199}$$

Bar example, $n = 1$, Finite Difference



- In Finite Difference method $n = 1$ we need one unknown.
- The essential boundary condition at $x = 0$ is $u(x = 0) := u_0 = 1$
- As shown in the figure our unknown is $u(x = 2) := u_1$.
- At $x = 1$ the differential equation we need to employ in Finite Difference method is the natural boundary condition ($\partial\mathcal{D}_f = \{1\}$):

$$\{\mathcal{R}_f\}|_{x=2} = \bar{F} - L_f(u) = 1 - \frac{\mathrm{d}u}{\mathrm{d}x}(2) = 0 \tag{200}$$

- In Finite Difference we approximate the differential operator $\frac{\mathrm{d}u}{\mathrm{d}x}(2)$:

$$\frac{\mathrm{d}u}{\mathrm{d}x}(2) \approx \frac{u_1 - u_0}{h} = \frac{u_1 - 1}{2} \tag{201}$$

- So from (200) and (201) we obtain:

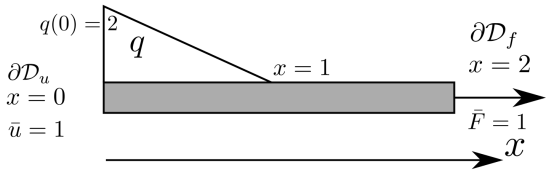
$$u(0) = u_0 = 1 \tag{202a}$$

$$u(2) = u_1 = 3 \tag{202b}$$

- Finite Difference solution only presents solution values at discrete points. Unlike WRM method, in FD we need another level of interpolation to get solution at other points.
- We denote this solution by u^h_{FD1}.

Bar example, $n = 1$, Least Square method

- For $n = 1$ our trial function is $[\phi] = \{x\}$ (*cf.* (196a)).



  - For least square method, the corresponding weight function are (*cf.* (159)):

$$\mathbf{w} = L_M(\boldsymbol{\phi}) \qquad \Rightarrow \qquad \mathbf{w}_1 = \frac{\mathrm{d}^2\boldsymbol{\phi}_1}{\mathrm{d}x^2} = \frac{\mathrm{d}^2 x}{\mathrm{d}x^2} = 0$$

$$\mathbf{w}^f = (-L_f(\boldsymbol{\phi})) \qquad \Rightarrow \qquad \mathbf{w}_1^f = -\frac{\mathrm{d}\boldsymbol{\phi}_1}{\mathrm{d}x} = -\frac{\mathrm{d}x}{\mathrm{d}x} = -1$$

  So for LS method, $n = 1$ weight functions are $[w] = 0, [w^f] = -1$.

- We plug $[\phi] = \{x\}$ into (185) to obtain:

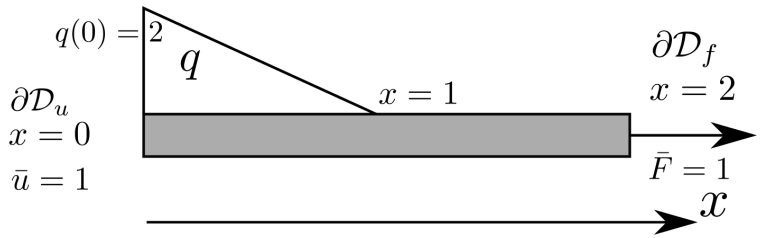
$$\mathbf{K} = \int_0^2 \overbrace{[\,0\,]}^{w_1}.\{\frac{\mathrm{d}^2}{\mathrm{d}x^2}[x]\}^{\mathrm{T}}\,\mathrm{d}x - \{\overbrace{[\,-1\,]}^{\mathbf{w}_1^f}.\left[\frac{\mathrm{d}[x]}{\mathrm{d}x}\right]^{\mathrm{T}}\}|_{x=2} = -\{[-1].[1]^{\mathrm{T}}\}|_{x=2} = -(-1) = 1$$

$$\mathbf{F} = -\int_0^1 \overbrace{[\,0\,]}^{w_1}.(2-2x)\,\mathrm{d}x - \overbrace{[\,-1\,]}^{\mathbf{w}_1^f}_{x=2} = 0 - [-1]_{x=2} = -(-1) = 1 \qquad (203)$$

- From $\mathbf{Ka} = \mathbf{F}$ (125) and $u^h = a_j\phi_j + \phi_p$ (117a), and $\phi_p = 1$ (182):

$$\boxed{u^h_{L1} = 1 + x} \qquad (204)$$

Bar example, $n = 1$, LS method, direct calculation



- To show that the weighted residual method for $\mathbf{w} = L_M(\boldsymbol{\phi})$ and $\mathbf{w}^f = (-L_f(\boldsymbol{\phi}))$ (*cf.* (159)), we directly calculate the R^2.

$$\mathcal{R}_i = L_M(u^h) - r = \frac{\mathrm{d}^2\{a_i\phi i + \phi_p\}}{\mathrm{d}x^2} + q(x) = a_1\frac{\mathrm{d}^2 x}{\mathrm{d}x^2} + \frac{\mathrm{d}^2 1}{\mathrm{d}x^2} + q(x) = q(x)$$

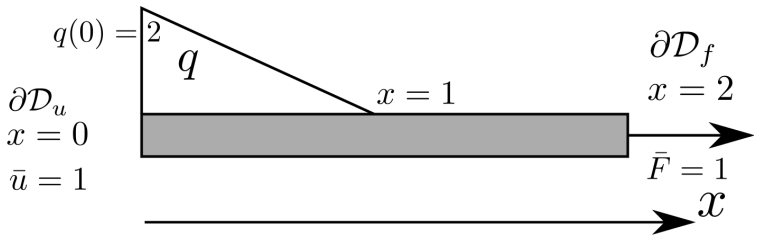
$$\mathcal{R}_f = \bar{f} - L_f(u^h) = \bar{F} - \frac{\mathrm{d}\{a_i\phi i + \phi_p\}}{\mathrm{d}x} = 1 - a_1\frac{\mathrm{d}x}{\mathrm{d}x} - \frac{\mathrm{d}1}{\mathrm{d}x} = 1 - a_1$$

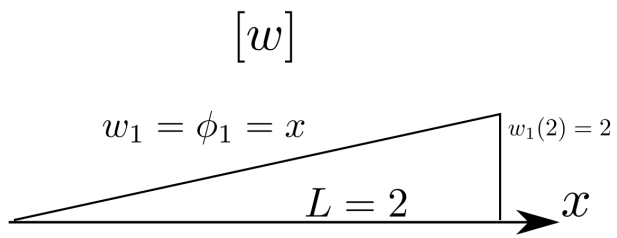
- the R^2 term is then,

$$R^2(a_1) = \int_0^2 \mathcal{R}_i^2\,\mathrm{d}x + \{\mathcal{R}_f(x=2)\}^2 = \int_0^1 (2-2x)^2\,\mathrm{d}x + (1-a_1)^2 = \frac{7}{3} - 2a_1 + {a_1}^2$$

- From the stationary condition $\frac{\partial R^2}{\partial a_1} = 2a_1 - 2 = 0$, we get to the same solution $a_1 = 1$.

Bar example, $n = 1$, Galerkin method





- For $n = 1$ our trial function is $[\phi] = \{x\}$ (*cf.* (196a)).
- For Galerkin method $n = 1$ weight function is $[w] = [w^f] = [\phi]$ (shown in figure).
- We plug $[\phi] = \{x\}$ and $[w] = \{x\}$ into (185) to obtain:

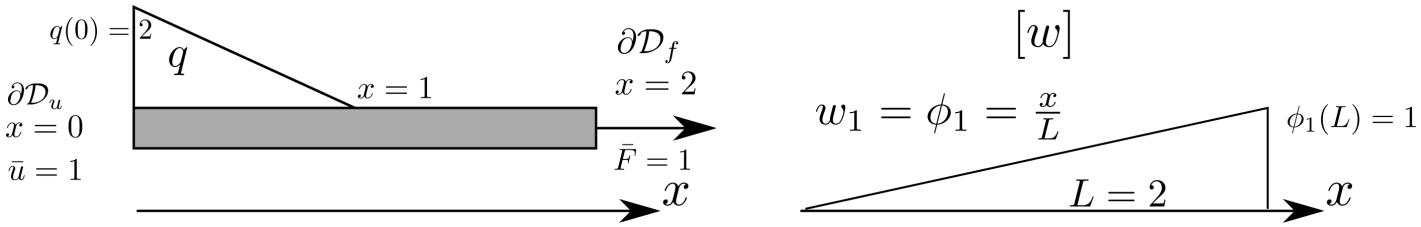
$$\begin{aligned}\mathbf{K} &= \int_0^2 [x].\{\frac{\mathrm{d}^2}{\mathrm{d}x^2}[x]\}^{\mathrm{T}}\,\mathrm{d}x - \{[x].\left(\frac{\mathrm{d}[x]}{\mathrm{d}x}\right)^{\mathrm{T}}\}|_{x=2} \\ &= \int_0^2 [x].\,[0]^{\mathrm{T}}\,\mathrm{d}x - \{[x].\,[1]^{\mathrm{T}}\}|_{x=2} = 0 - 2 = -2 \qquad (205a)\end{aligned}$$

$$\mathbf{F} = -\int_0^1 [x].(2 - 2x)\,\mathrm{d}x - [x]_{x=2} = -1 + \frac{2}{3} - 2 = -\frac{7}{3} \qquad (205b)$$

- From $\mathbf{Ka} = \mathbf{F}$ (125) and $u^h = a_j\phi_j + \phi_p$ (117a), and $\phi_p = 1$ (182):

$$\boxed{u^h_{G1} = 1 + \frac{7}{6}x} \qquad (206)$$

Bar example, $n = 1$, Galerkin Finite Element Method



- In 1D Finite Element method for the bar the trial functions are hat functions that take the value of one at one "node" and zero elsewhere.
- They will form a piece-wise linear solution which is not C^2 needed for the weighted residual method (term $\frac{\mathrm{d}^2\phi_i}{\mathrm{d}x^2}$ in weighted residual cannot be computed at nodes).
- We need to solve the problem with these types of trial functions in weak statement.
- For $n = 1$ as shown in the figure the trial and weight function are $\phi_i = w_1 = x/L = x/2$.
- $\phi_1 = w_1$ equal to $1/2$ of x in last Galerkin method; we expect to get the same solution.
- For $n > 1$ there will be differences in the solution.
- We plug $[\phi] = \{x/2\}$ and $[w] = \{x/2\}$ into (191) (weak statement) to obtain:

$$\mathbf{K} = \int_0^2 [\frac{\mathrm{d}}{\mathrm{d}x}\left(\frac{x}{2}\right)].[\frac{\mathrm{d}}{\mathrm{d}x}\left(\frac{x}{2}\right)]^{\mathrm{T}}\,\mathrm{d}x = \int_0^2 [\frac{1}{2}].[\frac{1}{2}]\,\mathrm{d}x = 2.\frac{1}{4} = \frac{1}{2} \tag{207a}$$

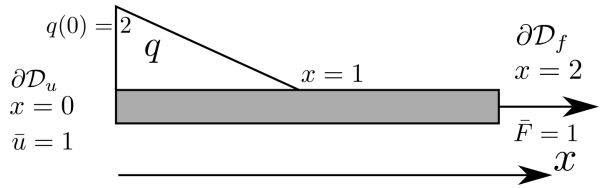
$$\mathbf{F} = \int_0^1 [\left(\frac{x}{2}\right)].(2 - 2x)\,\mathrm{d}x + [\left(\frac{x}{2}\right)]_{x=2} = \frac{1}{2} - \frac{1}{3} + 1 = \frac{7}{6} \tag{207b}$$

- From $\mathbf{Ka} = \mathbf{F}$ we get $a_1 = \frac{7}{3}$. Thus $u^h = a_j\phi_j + \phi_p$ (117a), and $\phi_p = 1$ (182) yield,

$$\boxed{u^h_{FE1} = \frac{7}{3}\left(\frac{x}{2}\right) = 1 + \frac{7}{6}x} \tag{208}$$

Bar example, $n = 1$, Ritz (energy) Method

- As discussed before, the application of stationary condition on energy functional at continuum level directly yields the weak statement.



  - Subsequent discretization on this weak statement is a discrete weak statement.
  - The Galerkin solution of this weak system (*e.g.*, shown for $\phi_1 = w_1 = x$ previously) is equivalent to Ritz method:
  - In Ritz method, we first discretize then apply the stationary condition.
  - For $u^h = a_1\phi_1 + \phi_p = a_1 x + 1$ $(\phi_1 = x)$ from (194) we get,

$$\begin{aligned}
\Pi(a_1) &= \int_0^2 \frac{1}{2}\left(\frac{\mathrm{d}\{a_1 x + 1\}}{\mathrm{d}x}\right)^2 \mathrm{d}x - \int_0^1 \{a_1 x + 1\}(2 - 2x)\,\mathrm{d}x - \{a_1 x + 1\}(2) \\
&= \int_0^2 \frac{1}{2}{a_1}^2\,\mathrm{d}x - a_1\int_0^1 (2x - 2x^2)\,\mathrm{d}x - \int_0^1 (2 - 2x)\,\mathrm{d}x - (2a_1 + 1) \\
&= {a_1}^2 - a_1\left(1 - \frac{2}{3}\right) - (2 - 1) - (2a_1 + 1) = {a_1}^2 - \frac{7}{3}a_1 - 2
\end{aligned}$$

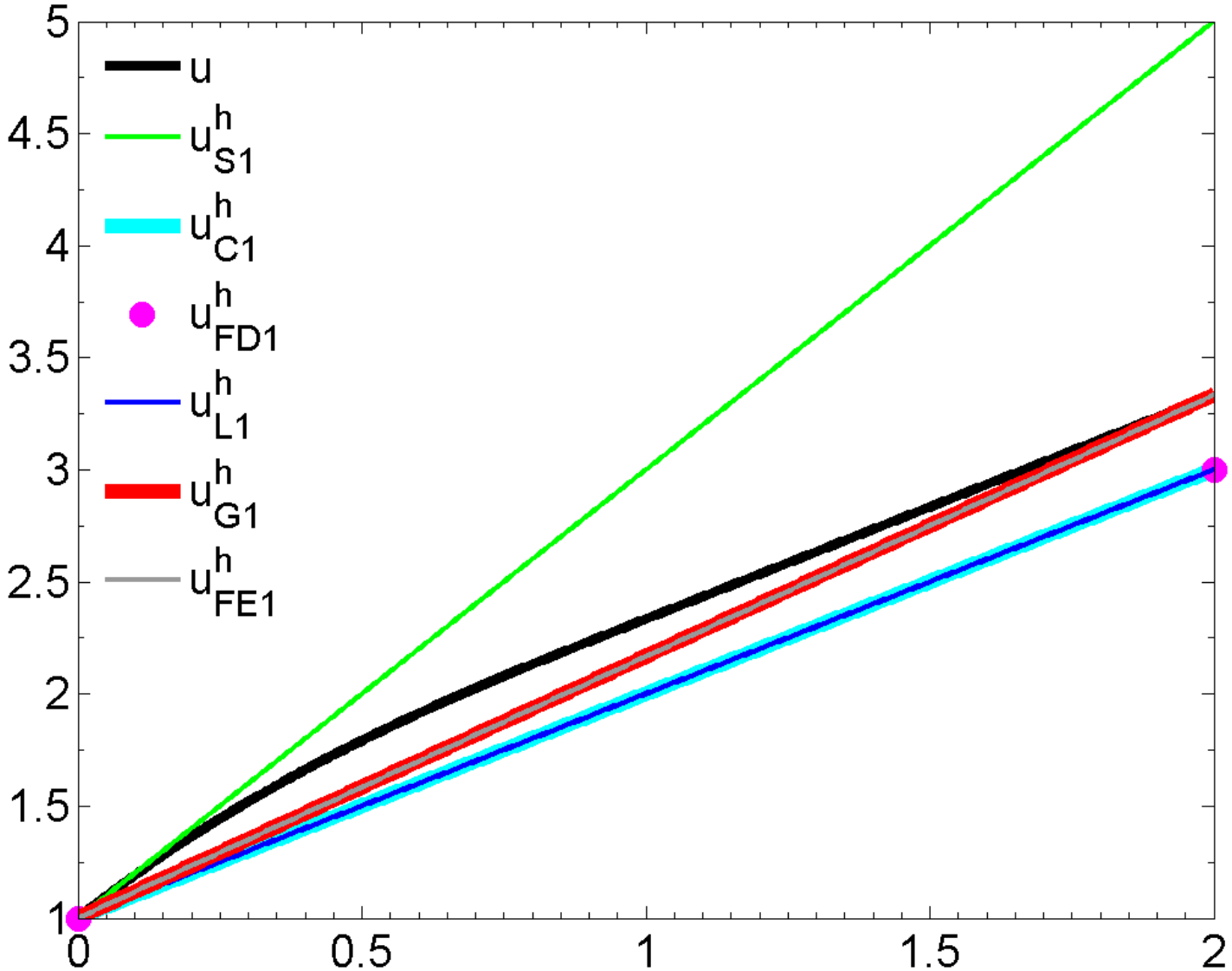
  - From stationary condition (195) the solution should satisfy,

$$\frac{\partial \Pi}{\partial a_1} = 0 \quad \Rightarrow 2a_1 - \frac{7}{3} = 0 \quad \Rightarrow a_1 = \frac{7}{6} \tag{209}$$

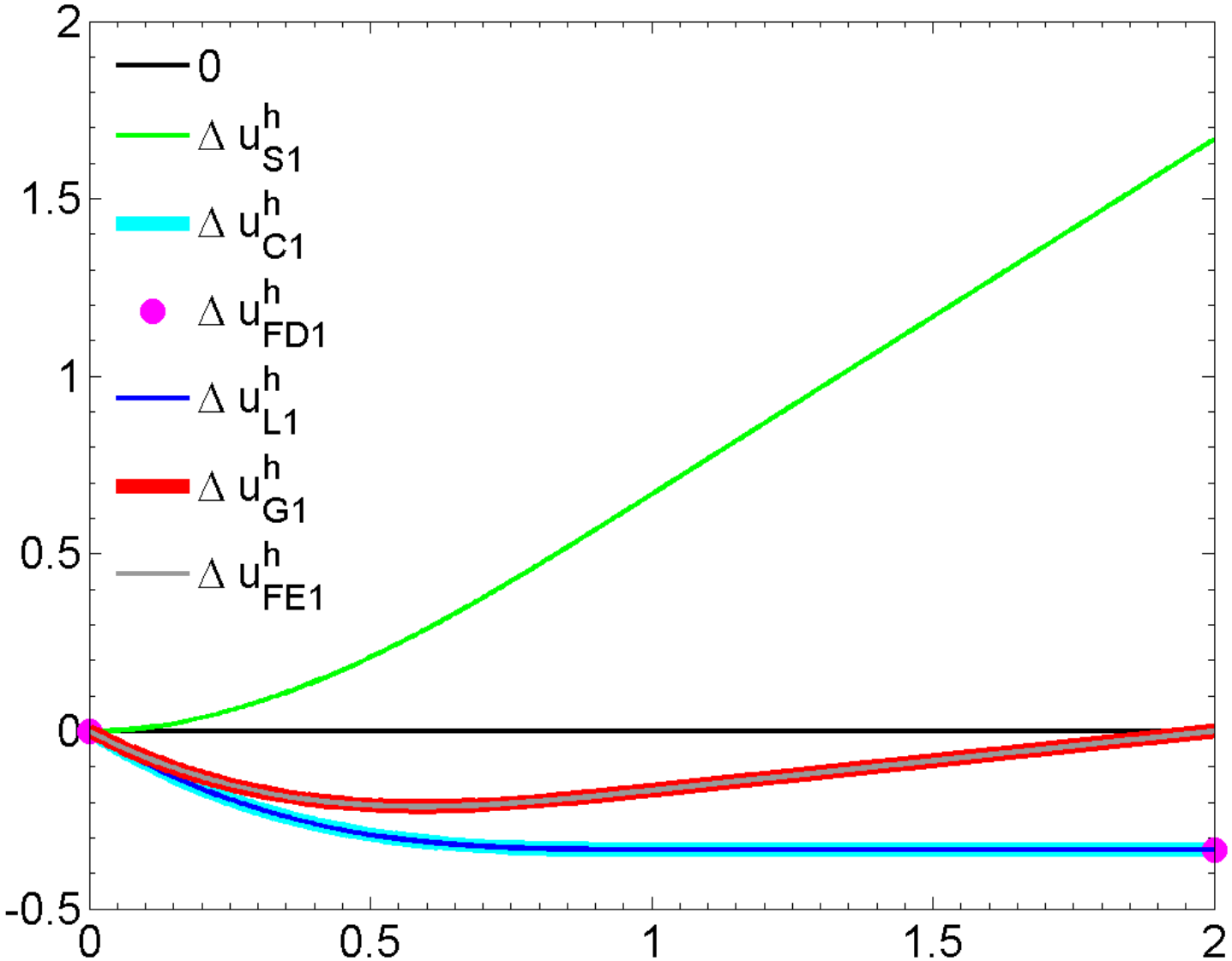
  - As expected the solution is equal to the solution we obtained from Galerkin method with the same trial function space:

$$\boxed{u^h_{E1} = \frac{7}{6}x + 1} \tag{210}$$

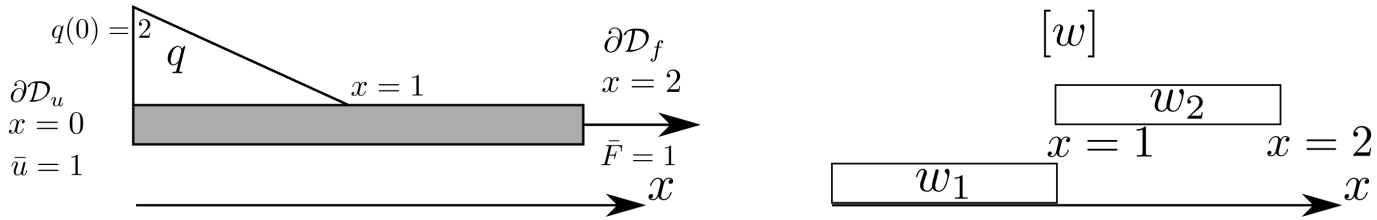
Bar example, $n = 1$, Comparison of solutions



Bar example, $n = 1$, Comparison of solutions



Bar example, $n = 2$, Subdomain method



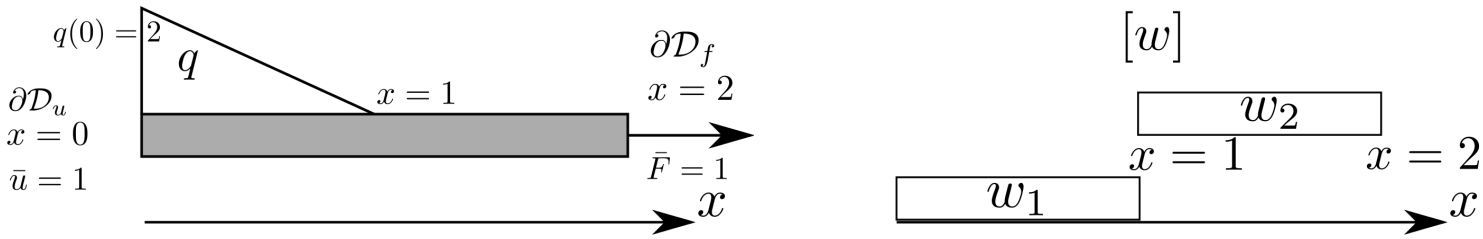
- For $n = 2$ our trial functions are $[\phi] = \{x, x^2\}$ (*cf.* (196b)).
- For subdomain method $n = 2$ weight functions are $[w] = [w^f] = \left[\chi_{[0\ 1]}, \chi_{[1\ 2]}\right]$.
- We plug $[\phi] = \{x, x^2\}$ and $[w] = \left[\chi_{[0\ 1]}, \chi_{[1\ 2]}\right]$ into (185) to obtain:

$$\begin{aligned}\mathbf{K}_{1j} &= \int_0^2 [\chi_{[0\ 1]}].\{\frac{\mathrm{d}^2}{\mathrm{d}x^2}[x\ x^2]\}\,\mathrm{d}x - \{[\chi_{[0\ 1]}].\frac{\mathrm{d}}{\mathrm{d}x}[x\ x^2]\}|_{x=2} \\ &= \int_0^1 [0\ 2]\ \mathrm{d}x - \{[0].[1\ 4]\} = [0\ 2] - [0\ 0] = [0\ 2] \qquad (211a)\\ \mathbf{K}_{2j} &= \int_0^2 [\chi_{[1\ 2]}].\{\frac{\mathrm{d}^2}{\mathrm{d}x^2}[x\ x^2]\}\,\mathrm{d}x - \{[\chi_{[1\ 2]}]\frac{\mathrm{d}}{\mathrm{d}x}[x\ x^2]\}|_{x=2} \\ &= \int_1^2 [0\ 2]\ \mathrm{d}x - \{[1]\,[1\ 4]\} = [0\ 2] - [1\ 4] = [-1\ \ -2] \qquad (211b)\end{aligned}$$

$$F_1 = -\int_0^1 [\chi_{[0\ 1]}](2-2x)\,\mathrm{d}x - [\chi_{[0\ 1]}]_{x=2} = -\int_0^1 (2-2x)\,\mathrm{d}x - 0 = -2+1 = -1$$

$$F_2 = -\int_0^1 [\chi_{[1\ 2]}].(2-2x)\,\mathrm{d}x - [\chi_{[1\ 2]}]_{x=2} = 0 - 1 = 0 - 1 = -1 \qquad (212)$$

Bar example, $n = 2$, Subdomain method



- Equations (211) and (212) yield,

$$\mathbf{K} = \begin{bmatrix} 0 & 2 \\ -1 & -2 \end{bmatrix}, \quad \mathbf{F} = \begin{bmatrix} -1 \\ -1 \end{bmatrix} \tag{213}$$

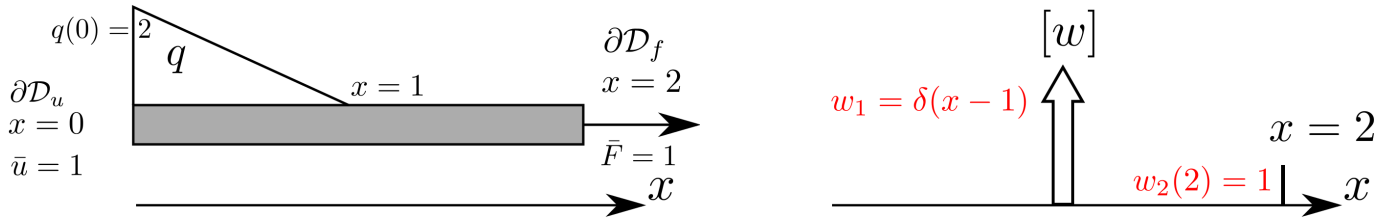
- From $\mathbf{Ka} = \mathbf{F}$ (125) and (213) we get,

$$\mathbf{a} = \begin{bmatrix} 2 \\ -\frac{1}{2} \end{bmatrix} \tag{214}$$

- From $u^h = a_j \phi_j + \phi_p$ (117a), $[\phi] = \{x, x^2\}$ (*cf.* (196b)), and $\phi_p = 1$ (182) we have

$$\boxed{u^h_{S2} = 1 + 2x - \frac{x^2}{2}} \tag{215}$$

Bar example, $n = 2$, Collocation method



- For $n = 2$ our trial functions are $[\phi] = \{x, x^2\}$ (*cf.* (196b)).
- For this collocation method we choose one point inside the domain ($x = 1$); the corresponding weight function color$redw_1 = \delta(x-1)$ only acts on $\mathcal{R}_i$ at $x = 1$.
- The second collocation point is chosen on $\partial\mathcal{D}_f$ ($x = 2$); the corresponding weight function color$redw_2 = \chi_{\{2\}}$ only acts on $\mathcal{R}_f$ at $x = 2$.
- We plug these functions into (185) to obtain (also refer to (148), (149), and (149b))

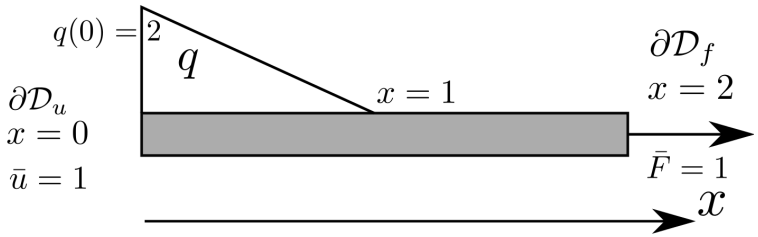
$$\mathbf{K}_{1j} = \int_0^2 [\delta(x-1)].\{\frac{\mathrm{d}^2}{\mathrm{d}x^2}[x\ x^2]\}\,\mathrm{d}x - \{[\delta(x-1)].\frac{\mathrm{d}}{\mathrm{d}x}[x\ x^2]\}|_{x=2} = [0\ 2]_{x=1} - 0 = [0\ 2] \tag{216}$$

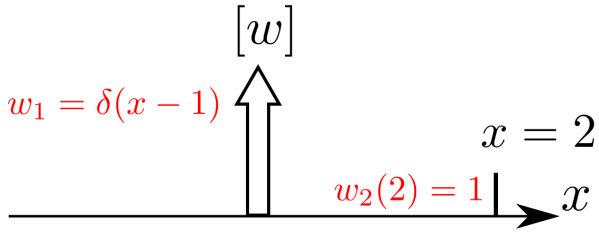
$$\mathbf{K}_{2j} = \int_0^2 [\chi_{\{2\}}].\{\frac{\mathrm{d}^2}{\mathrm{d}x^2}[x\ x^2]\}\,\mathrm{d}x - \{[\chi_{\{2\}}]\frac{\mathrm{d}}{\mathrm{d}x}[x\ x^2]\}|_{x=2} = 0 - \{[1]\,[1\ 4]\} = [-1\ \ -4]$$

$$F_1 = -\int_0^1 [\delta(x-1)](2-2x)\,\mathrm{d}x - [\delta(x-1)]_{x=2} = -(2-2x)_{x=1} - 0 = -2+2 = 0$$

$$F_2 = -\int_0^1 [\chi_{\{2\}}].(2-2x)\,\mathrm{d}x - [\chi_{\{2\}}]_{x=2} = 0 - 1 = -1 \tag{217}$$

Bar example, $n = 2$, Collocation method





- Equations (216) and (217) yield,

$$\mathbf{K} = \begin{bmatrix} 0 & 2 \\ -1 & -4 \end{bmatrix}, \quad \mathbf{F} = \begin{bmatrix} 0 \\ -1 \end{bmatrix} \tag{218}$$

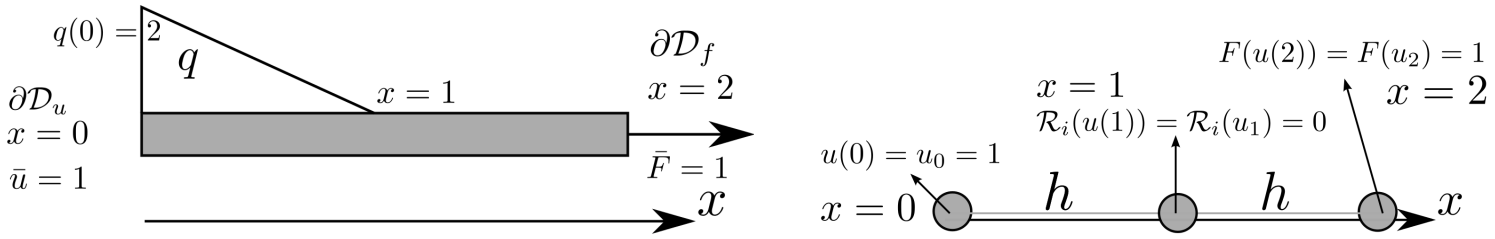
- From $\mathbf{Ka} = \mathbf{F}$ (125) and (218) we get,

$$\mathbf{a} = \begin{bmatrix} 1 \\ 0 \end{bmatrix} \tag{219}$$

- From $u^h = a_j \phi_j + \phi_p$ (117a), $[\phi] = \{x, x^2\}$ (*cf.* (196b)), and $\phi_p = 1$ (182) we have

$$\boxed{u^h_{C2} = 1 + x} \tag{220}$$

Bar example, $n = 2$, Finite Difference



- In Finite Difference method $n = 2$ we need two unknowns.
- The essential boundary condition at $x = 0$ is $u(x = 0) := u_0 = 1$
- 1st unknown is $u_1 = u(x = 1)$; we enforce approximate FD for $\mathcal{R}_i(x = 1) = 0$.
- 2nd unknown is $u_2 = u(x = 2)$; we enforce approximate FD for $\mathcal{R}_f(x = 2) = 0$.
- The equations are:

$$\mathcal{R}_i(x = 1) = \frac{\mathrm{d}^2 u}{\mathrm{d}x^2} + q(x) \approx \frac{u_0 - 2u_1 + u_2}{h^2} + 0 = 1 - 2u_1 + u_2 = 0 \tag{221a}$$

$$\mathcal{R}_f(x = 2) = \bar{F} - \frac{\mathrm{d}u}{\mathrm{d}x} \approx 1 - \frac{u_2 - u_1}{h} = 1 - u_2 + u_1 = 0 \tag{221b}$$

- Unknowns u_i are obtained from (221),

$$\mathbf{K} \begin{bmatrix} u_1 \\ u_2 \end{bmatrix} = \mathbf{F}, \quad for \quad \mathbf{K} = \begin{bmatrix} 2 & -1 \\ -1 & 1 \end{bmatrix}, \quad \mathbf{F} = \begin{bmatrix} 1 \\ 1 \end{bmatrix} \tag{222}$$

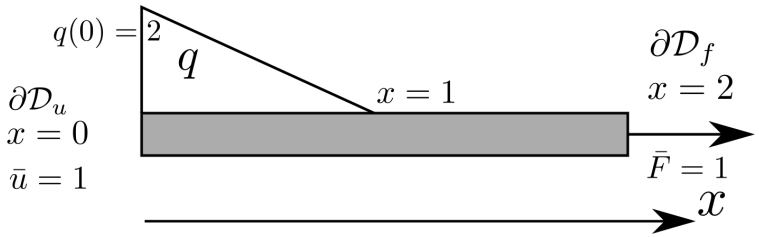
- The solution denoted by u^h_{FD2} is:

$$u(0) = u_0 = 1 \tag{223a}$$

$$u(1) = u_1 = 2 \tag{223b}$$

$$u(2) = u_2 = 3 \tag{223c}$$

Bar example, $n = 2$, Least Square method

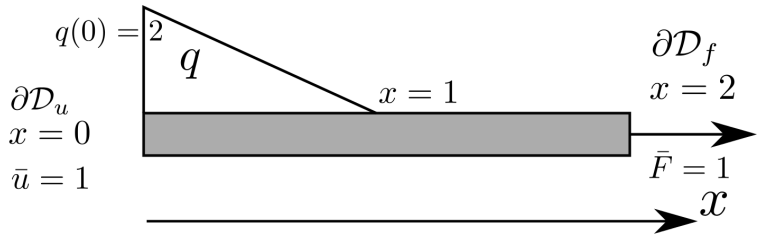


- For $n = 2$ our trial function is $[\phi] = \{x \; x^2\}$ (*cf.* (196b)).
- For least square method, the corresponding weight function are (*cf.* (159)):

$$\mathbf{w} = L_M(\boldsymbol{\phi}) \qquad \Rightarrow \qquad \mathbf{w} = \frac{\mathrm{d}^2[\boldsymbol{\phi}]}{\mathrm{d}x^2} = \frac{\mathrm{d}^2 \begin{bmatrix} x \\ x^2 \end{bmatrix}}{\mathrm{d}x^2} = \begin{bmatrix} 0 \\ 2 \end{bmatrix}$$

$$\mathbf{w}^f = (-L_f(\boldsymbol{\phi})) \qquad \Rightarrow \qquad \mathbf{w}^f = -\frac{\mathrm{d}[\boldsymbol{\phi}]}{\mathrm{d}x} = -\frac{\mathrm{d} \begin{bmatrix} x \\ x^2 \end{bmatrix}}{\mathrm{d}x} = -\begin{bmatrix} 1 \\ 2x \end{bmatrix}$$

Bar example, $n = 2$, Least Square method

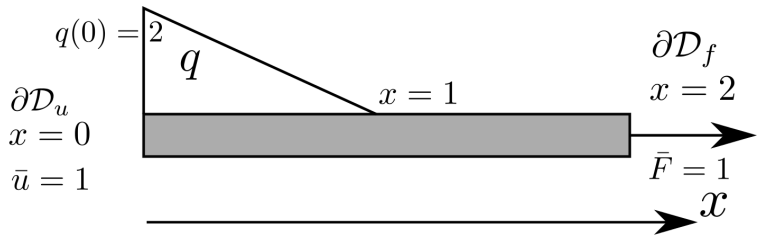


- We plug $[\phi] = \begin{bmatrix} x \\ x^2 \end{bmatrix}$ and (224) into (185) to obtain:

$$\begin{aligned} \mathbf{K} &= \int_0^2 \overbrace{\begin{bmatrix} 0 \\ 2 \end{bmatrix}}^{\mathbf{w}} . \{ \frac{\mathrm{d}^2}{\mathrm{d}x^2} [x \ x^2] \} \, \mathrm{d}x - \{ \overbrace{\begin{bmatrix} -1 \\ -2x \end{bmatrix}}^{\mathbf{w}^f} . \frac{\mathrm{d}[x \ x^2]}{\mathrm{d}x} \} |_{x=2} \\ &= \int_0^2 \begin{bmatrix} 0 \\ 2 \end{bmatrix} . \{ [0 \ 2] \} \, \mathrm{d}x + \{ \begin{bmatrix} 1 \\ 2x \end{bmatrix} . [1 \ 2x] \} |_{x=2} = \begin{bmatrix} 0 & 0 \\ 0 & 8 \end{bmatrix} + \begin{bmatrix} 1 & 4 \\ 4 & 16 \end{bmatrix} = \begin{bmatrix} 1 & 4 \\ 4 & 24 \end{bmatrix} \end{aligned} \tag{225}$$

$$\mathbf{F} = - \int_0^1 \overbrace{\begin{bmatrix} 0 \\ 2 \end{bmatrix}}^{\mathbf{w}} . (2 - 2x) \, \mathrm{d}x - \overbrace{\begin{bmatrix} -1 \\ -2x \end{bmatrix}}^{\mathbf{w}^f} x=2 = - \begin{bmatrix} 0 \\ 4 - 2 \end{bmatrix} - \begin{bmatrix} -1 \\ -4 \end{bmatrix} = \begin{bmatrix} 1 \\ 2 \end{bmatrix} \tag{226}$$

Bar example, $n = 2$, Least Square method



- Equations (225) and (226) yield,

$$\mathbf{K} = \begin{bmatrix} 1 & 4 \\ 4 & 24 \end{bmatrix}, \quad \mathbf{F} = \begin{bmatrix} 1 \\ 2 \end{bmatrix} \tag{227}$$

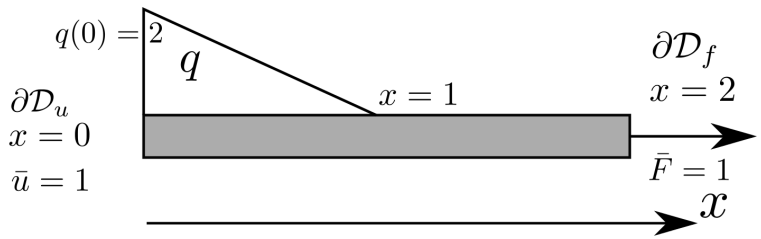
- From $\mathbf{Ka} = \mathbf{F}$ (125) and (227) we get,

$$\mathbf{a} = \begin{bmatrix} 2 \\ -\frac{1}{4} \end{bmatrix} \tag{228}$$

- From $u^h = a_j \phi_j + \phi_p$ (117a), $[\phi] = \{x, x^2\}$ (*cf.* (196b)), and $\phi_p = 1$ (182) we have

$$\boxed{u^h_{L2} = 1 + 2x - \frac{x^2}{4}} \tag{229}$$

Bar example, $n = 2$, LS method, direct calculation



- To show that the weighted residual method for $\mathbf{w} = L_M(\boldsymbol{\phi})$ and $\mathbf{w}^f = (-L_f(\boldsymbol{\phi}))$ (*cf.* (159)), we directly calculate the R^2.

$$\mathcal{R}_i = L_M(u^h) - r = \frac{\mathrm{d}^2\{a_i\phi i + \phi_p\}}{\mathrm{d}x^2} + q = a_1\frac{\mathrm{d}^2 x}{\mathrm{d}x^2} + a_2\frac{\mathrm{d}^2 x^2}{\mathrm{d}x^2} + \frac{\mathrm{d}^2 1}{\mathrm{d}x^2} + q = 2a_2 + q(x)$$

$$\mathcal{R}_f = \bar{f} - L_f(u^h) = \bar{F} - \frac{\mathrm{d}\{a_i\phi i + \phi_p\}}{\mathrm{d}x} = 1 - a_1\frac{\mathrm{d}x}{\mathrm{d}x} - a_2\frac{\mathrm{d}x^2}{\mathrm{d}x} - \frac{\mathrm{d}1}{\mathrm{d}x} = 1 - a_1 - 2a_2 x$$

- R^2 term is then,

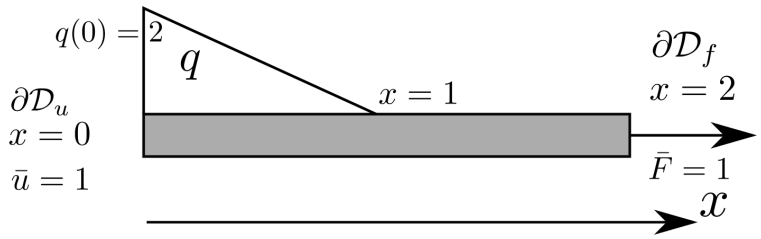
$$R^2(a_1, a_2) = \int_0^2 \mathcal{R}_i^2\,\mathrm{d}x + \{\mathcal{R}_f(x=2)\}^2 = \int_0^1 (2a_2 + 2 - 2x)^2\,\mathrm{d}x + \int_1^2 (2a_2)^2\,\mathrm{d}x$$

$$+ (1 - a_1 - 4a_2)^2 = \int_0^1 \{4(a_2+1)^2 - 8(a_2+1)x + 4x^2\}\,\mathrm{d}x + 4a_2^2$$

$$+ 1 + a_1^2 + 16a_2^2 - 2a_1 - 8a_2 + 8a_1a_2 = (4a_2^2 + 8a_2 + 4 - 4a_2 - 4 + \frac{4}{3})$$

$$+ (1 + a_1^2 + 20a_2^2 - 2a_1 - 8a_2 + 8a_1a_2) \Rightarrow$$

Bar example, $n = 2$, LS method, direct calculation



$$R^2(a_1, a_2) = a_1^2 + 24a_2^2 + 8a_1a_2 - 2a_1 - 4a_2 + \frac{7}{3} \tag{230}$$

- Stationary condition for (230) yields,

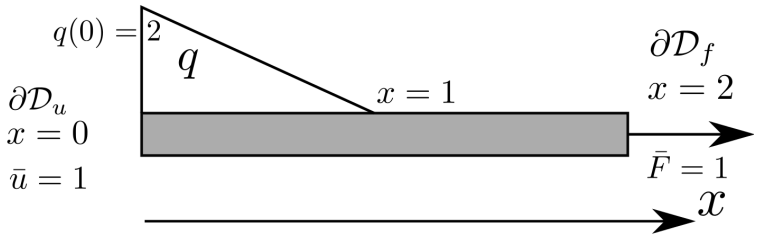
$$\nabla R^2(a_1, a_2) = \begin{bmatrix} \frac{\partial R^2}{\partial a_1} \\ \frac{\partial R^2}{\partial a_2} \end{bmatrix} = \begin{bmatrix} 2a_1 + 8a_2 - 2 \\ 8a_1 + 48a_2 - 4 \end{bmatrix} = \begin{bmatrix} 0 \\ 0 \end{bmatrix} \quad \Rightarrow$$

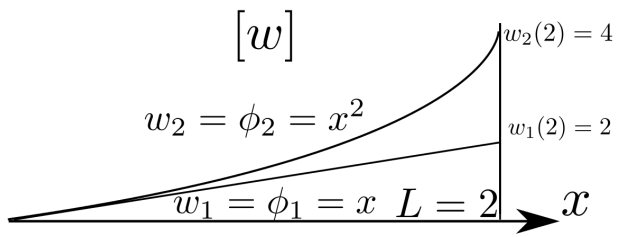
$$\mathbf{Ka} = \mathbf{F} \quad \text{for} \quad \mathbf{K} = \begin{bmatrix} 2 & 8 \\ 8 & 48 \end{bmatrix}, \quad \mathbf{F} = \begin{bmatrix} 2 \\ 4 \end{bmatrix} \quad \Rightarrow \quad \mathbf{a} = \begin{bmatrix} 2 \\ -\frac{1}{4} \end{bmatrix}$$

- This solution agrees with (228) and yields the same solution in (229):

$$u_{L2}^h = 1 + 2x - \frac{x^2}{4}$$

Bar example, $n = 2$, Galerkin method



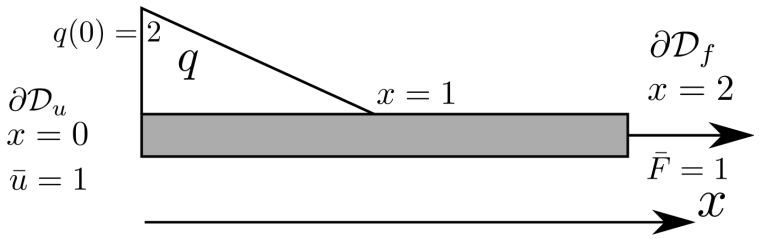


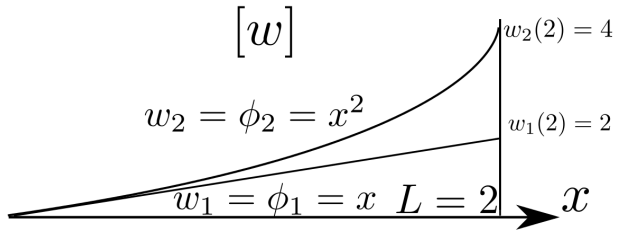
- For $n = 2$ our trial functions are $[\phi] = \{x, x^2\}$ (*cf.* (196b)).
- For Galerkin method $n = 2$ weight functions are $[w] = [w^f] = [\phi] = \{x, x^2\}$.
- We plug $[\phi] = [w] = [w^f] = \{x, x^2\}$ into (185) to obtain:

$$\begin{aligned}\mathbf{K} &= \int_0^2 \begin{bmatrix} x \\ x^2 \end{bmatrix} . \{ \frac{\mathrm{d}^2}{\mathrm{d}x^2} [x \; x^2] \} \, \mathrm{d}x - \{ \begin{bmatrix} x \\ x^2 \end{bmatrix} . \frac{\mathrm{d}}{\mathrm{d}x} [x \; x^2] \} |_{x=2} \\ &= \int_0^2 \begin{bmatrix} 0 & 2x \\ 0 & 2x^2 \end{bmatrix} \, \mathrm{d}x - \{ \begin{bmatrix} x & 2x^2 \\ x^2 & 2x^3 \end{bmatrix} \} |_{x=2} \\ &= \begin{bmatrix} 0 & 4 \\ 0 & \frac{16}{3} \end{bmatrix} - \begin{bmatrix} 2 & 8 \\ 4 & 16 \end{bmatrix} = \begin{bmatrix} -2 & -4 \\ -4 & -\frac{32}{3} \end{bmatrix} \end{aligned} \tag{231}$$

$$\begin{aligned}\mathbf{F} &= - \int_0^1 \begin{bmatrix} x \\ x^2 \end{bmatrix} . (2 - 2x) \, \mathrm{d}x - \begin{bmatrix} x \\ x^2 \end{bmatrix} \Big|_{x=2} \\ &- \int_0^1 \begin{bmatrix} 2x - 2x^2 \\ 2x^2 - 2x^3 \end{bmatrix} \, \mathrm{d}x - \begin{bmatrix} 2 \\ 4 \end{bmatrix} = \begin{bmatrix} -1 + \frac{2}{3} - 2 \\ -\frac{2}{3} + \frac{1}{2} - 4 \end{bmatrix} = \begin{bmatrix} -\frac{7}{3} \\ -\frac{25}{6} \end{bmatrix} \end{aligned} \tag{232}$$

Bar example, $n = 2$, Galerkin method





- Equations (231) and (232) yield,

$$\mathbf{K} = \begin{bmatrix} -2 & -4 \\ -4 & -\frac{32}{3} \end{bmatrix}, \quad \mathbf{F} = \begin{bmatrix} -\frac{7}{3} \\ -\frac{25}{6} \end{bmatrix} \tag{233}$$

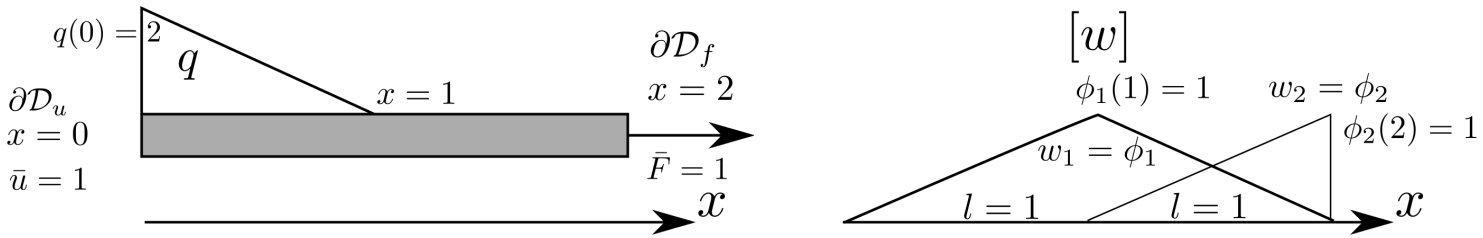
- From $\mathbf{Ka} = \mathbf{F}$ (125) and (233) we get,

$$\mathbf{a} = \begin{bmatrix} \frac{37}{24} \\ -\frac{3}{16} \end{bmatrix} \tag{234}$$

- From $u^h = a_j \phi_j + \phi_p$ (117a), $[\phi] = \{x, x^2\}$ (*cf.* (196b)), and $\phi_p = 1$ (182) we have

$$\boxed{u^h_{G2} = 1 + \frac{37}{24}x - \frac{3}{16}x^2} \tag{235}$$

Bar example, $n = 2$, Galerkin Finite Element Method



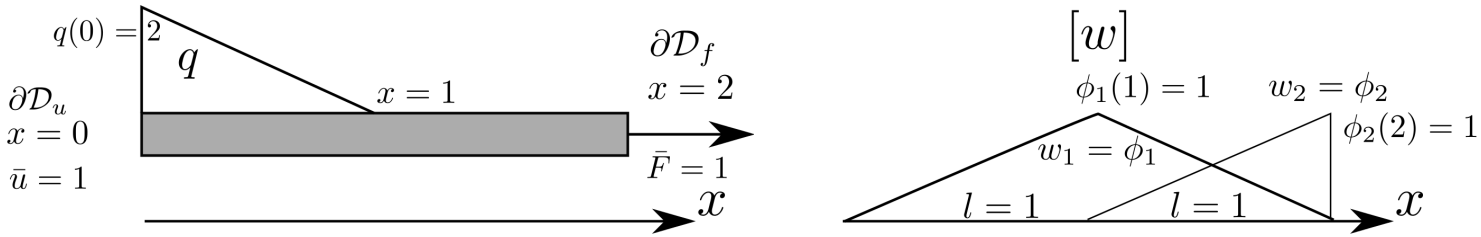
- In 1D Finite Element method for the bar the trial functions are hat functions that take the value of one at one "node" and zero elsewhere.
- They will form a piece-wise linear solution which is not C^2 needed for the weighted residual method (terms $\frac{\mathrm{d}^2\phi_i}{\mathrm{d}x^2}$ in weighted residual are not functions).
- For example $\frac{\mathrm{d}^2\phi_1}{\mathrm{d}x^2}(1) = -2\delta(x-1)$.
- Even $\frac{\mathrm{d}\phi_1}{\mathrm{d}x}(1)$ is not defined, but all the integrals in the weak statement can be evaluated.
- We need to solve the problem with these types of trial functions in weak statement.
- For $n = 2$ as shown in the figure the trial and weight functions and their derivatives are :

$$\phi_1 = w_1 = \begin{cases} x & 0 \le x < 1 \\ 2 - x & 1 \le x \le 2 \end{cases} \quad \Rightarrow \quad \phi_1' = w_1' = \begin{cases} 1 & 0 \le x < 1 \\ -1 & 1 \le x \le 2 \end{cases} \tag{236a}$$

$$\phi_2 = w_2 = \begin{cases} 0 & 0 \le x < 1 \\ x - 1 & 1 \le x \le 2 \end{cases} \quad \Rightarrow \quad \phi_2' = w_2' = \begin{cases} 0 & 0 \le x < 1 \\ 1 & 1 \le x \le 2 \end{cases} \tag{236b}$$

- Note that each trial function takes the value of 1 at its corresponding point: $\phi_1(1) = \phi_2(2) = 1$.

Bar example, $n = 2$, Galerkin Finite Element Method

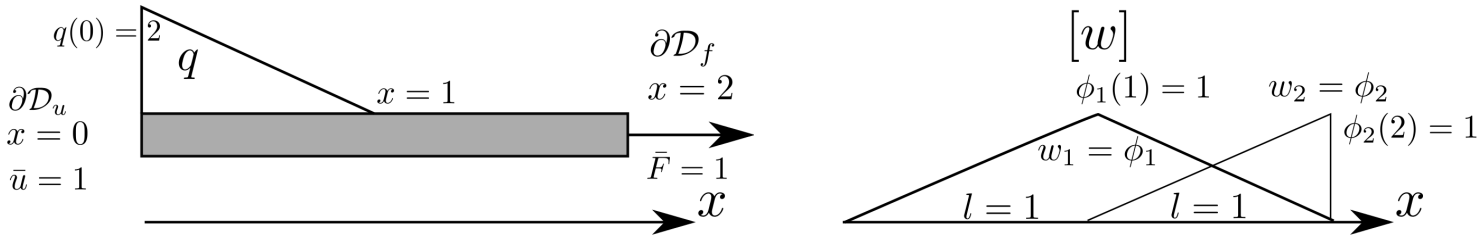


- We plug (236) into (191) (weak statement) to obtain:

$$\begin{aligned}
\mathbf{K} &= \int_0^2 \frac{\mathrm{d}}{\mathrm{d}x}\begin{bmatrix}\phi_1\\ \phi_2\end{bmatrix}.\frac{\mathrm{d}}{\mathrm{d}x}\begin{bmatrix}\phi_1 & \phi_2\end{bmatrix}\mathrm{d}x = \int_0^2 \begin{bmatrix}\phi_1'\phi_1' & \phi_1'\phi_2'\\ \phi_2'\phi_1' & \phi_2'\phi_2'\end{bmatrix}\mathrm{d}x \\
&= \int_0^1 \begin{bmatrix}(1).(1) & (1).(0)\\ (0).(1) & (0).(0)\end{bmatrix}\mathrm{d}x + \int_0^1 \begin{bmatrix}(-1).(-1) & (-1).(1)\\ (1).(-1) & (1).(1)\end{bmatrix}\mathrm{d}x \\
&= \begin{bmatrix}1 & 0\\ 0 & 0\end{bmatrix} + \begin{bmatrix}1 & -1\\ -1 & 1\end{bmatrix} = \begin{bmatrix}2 & -1\\ -1 & 1\end{bmatrix}
\end{aligned} \tag{237}$$

$$\begin{aligned}
\mathbf{F} &= \int_0^2 \begin{bmatrix}\phi_1\\ \phi_2\end{bmatrix}.q(x)\,\mathrm{d}x + \left\{\begin{bmatrix}\phi_1\\ \phi_2\end{bmatrix}\right\}_{x=2} \\
&= \int_0^1 \begin{bmatrix}x\\ 0\end{bmatrix}.(2-2x)\,\mathrm{d}x + \int_1^2 \begin{bmatrix}2-x\\ x-1\end{bmatrix}.0\,\mathrm{d}x + \begin{bmatrix}2-2\\ 2-1\end{bmatrix} \\
&= \int_0^1 \begin{bmatrix}2x-2x^2\\ 0\end{bmatrix}\mathrm{d}x + \begin{bmatrix}0\\ 1\end{bmatrix} = \begin{bmatrix}1-\frac{2}{3}\\ 1\end{bmatrix} = \begin{bmatrix}\frac{1}{3}\\ 1\end{bmatrix}
\end{aligned} \tag{238}$$

Bar example, $n = 2$, Galerkin Finite Element Method



- Equations (237) and (238) yield,

$$\mathbf{K} = \begin{bmatrix} 2 & -1 \\ -1 & 1 \end{bmatrix}, \quad \mathbf{F} = \begin{bmatrix} \frac{1}{3} \\ 1 \end{bmatrix} \tag{239}$$

- From $\mathbf{Ka} = \mathbf{F}$ (125) and (239) we get,

$$\mathbf{a} = \begin{bmatrix} \frac{4}{3} \\ \frac{7}{3} \end{bmatrix} \tag{240}$$

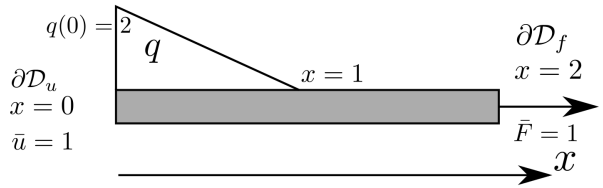
- From $u^h = a_j \phi_j + \phi_p$ (117a), and $\phi_p = 1$ (182) we have

$$\boxed{u^h_{FE2} = 1 + \frac{4}{3}\phi_1 + \frac{7}{3}\phi_2} \tag{241}$$

- What is $u^h_{FE2}(1)$?
- What is $u^h_{FE2}(2)$?

Bar example, $n = 2$, Ritz (energy) Method

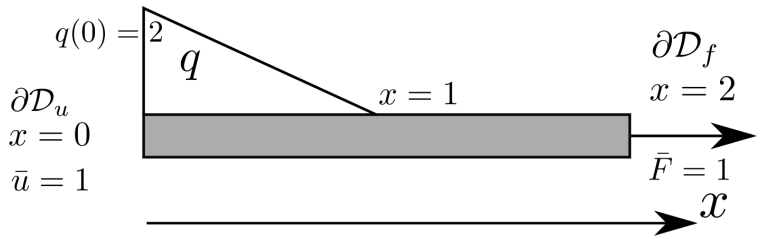
- As discussed before, the application of stationary condition on energy functional at continuum level directly yields the weak statement.



  - Subsequent discretization on this weak statement is a discrete weak statement.
  - The Galerkin solution of this weak system (*e.g.*, shown for $\phi = \mathbf{w} = \begin{bmatrix} x & x^2 \end{bmatrix}^{\mathrm{T}}$ previously) is equivalent to Ritz method:
  - In Ritz method, we first discretize then apply the stationary condition.
  - For $u^h = a_i \phi_i + \phi_p = a_1 x + a_2 x^2 + 1$ from (194) we get,

$$
\begin{aligned}
\Pi(a_1, a_2) &= \int_0^2 \frac{1}{2} \left(\frac{\mathrm{d}\{a_1 x + a_2 x^2 + 1\}}{\mathrm{d}x} \right)^2 \mathrm{d}x - \int_0^1 \{a_1 x + a_2 x^2 + 1\}(2 - 2x)\, \mathrm{d}x \\
&\quad - \{a_1 x + a_2 x^2 + 1\}(2) = \int_0^2 \frac{1}{2} (a_1 + 2a_2 x)^2 \, \mathrm{d}x - a_1 \int_0^1 (2x - 2x^2)\, \mathrm{d}x \\
&\quad - a_2 \int_0^1 (2x^2 - 2x^3)\, \mathrm{d}x - \int_0^1 (2 - 2x)\, \mathrm{d}x - (2a_1 + 4a_2 + 1) \\
&\quad = \int_0^2 \left\{ \frac{1}{2} a_1^2 + 2a_1 a_2 x + 2a_2^2 x^2 \right\} \mathrm{d}x - a_1 (1 - \frac{2}{3}) - a_2 (\frac{2}{3} - \frac{1}{2}) - (2 - 1) \\
&\quad - (2a_1 + 4a_2 + 1) = \left(a_1^2 + 4a_1 a_2 + \frac{16}{3} a_2^2 \right) - \left(\frac{7}{3} a_1 + \frac{25}{6} a_2 + 2 \right) \quad \Rightarrow
\end{aligned}
$$

Bar example, $n = 2$, Ritz (energy) Method



$$\Pi(a_1, a_2) = a_1^2 + 4a_1a_2 + \frac{16}{3}a_2^2 - \frac{7}{3}a_1 - \frac{25}{6}a_2 - 2 \tag{242}$$

- From stationary condition (195) the solution should satisfy,

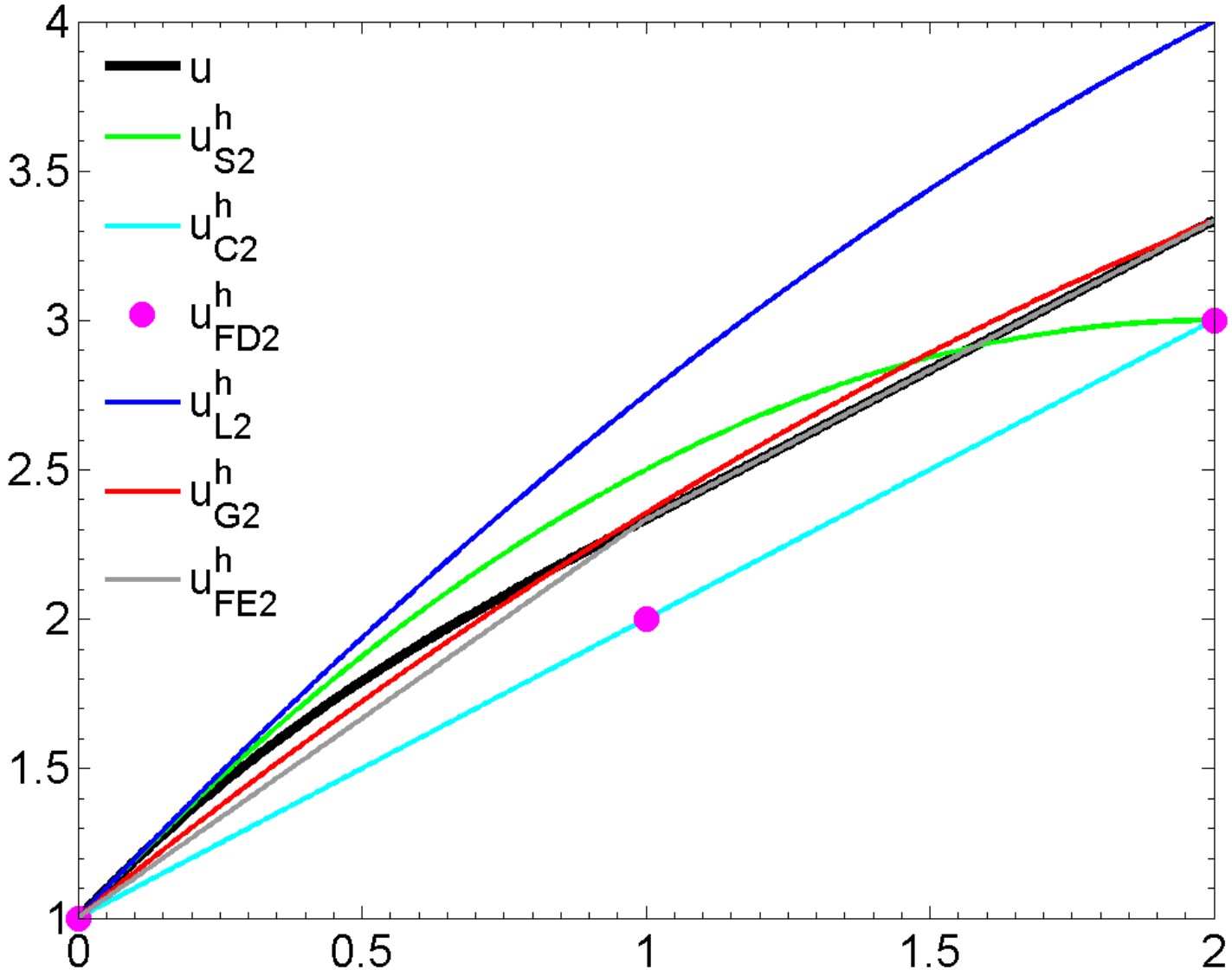
$$\nabla\Pi(a_1, a_2) = \begin{bmatrix} \frac{\partial\Pi}{\partial a_1} \\ \frac{\partial\Pi}{\partial a_2} \end{bmatrix} = \begin{bmatrix} 2a_1 + 4a_2 - \frac{7}{3} \\ 4a_1 + \frac{32}{3}a_2 - \frac{25}{6} \end{bmatrix} = \mathbf{0} \quad \Rightarrow \tag{243}$$

$$\mathbf{Ka} = \mathbf{F} \quad \text{for} \quad \mathbf{K} = \begin{bmatrix} 2 & 4 \\ 4 & \frac{32}{3} \end{bmatrix}, \quad \mathbf{F} = \begin{bmatrix} \frac{7}{3} \\ \frac{25}{6} \end{bmatrix} \quad \Rightarrow \quad \mathbf{a} = \begin{bmatrix} \frac{37}{24} \\ -\frac{3}{16} \end{bmatrix}$$

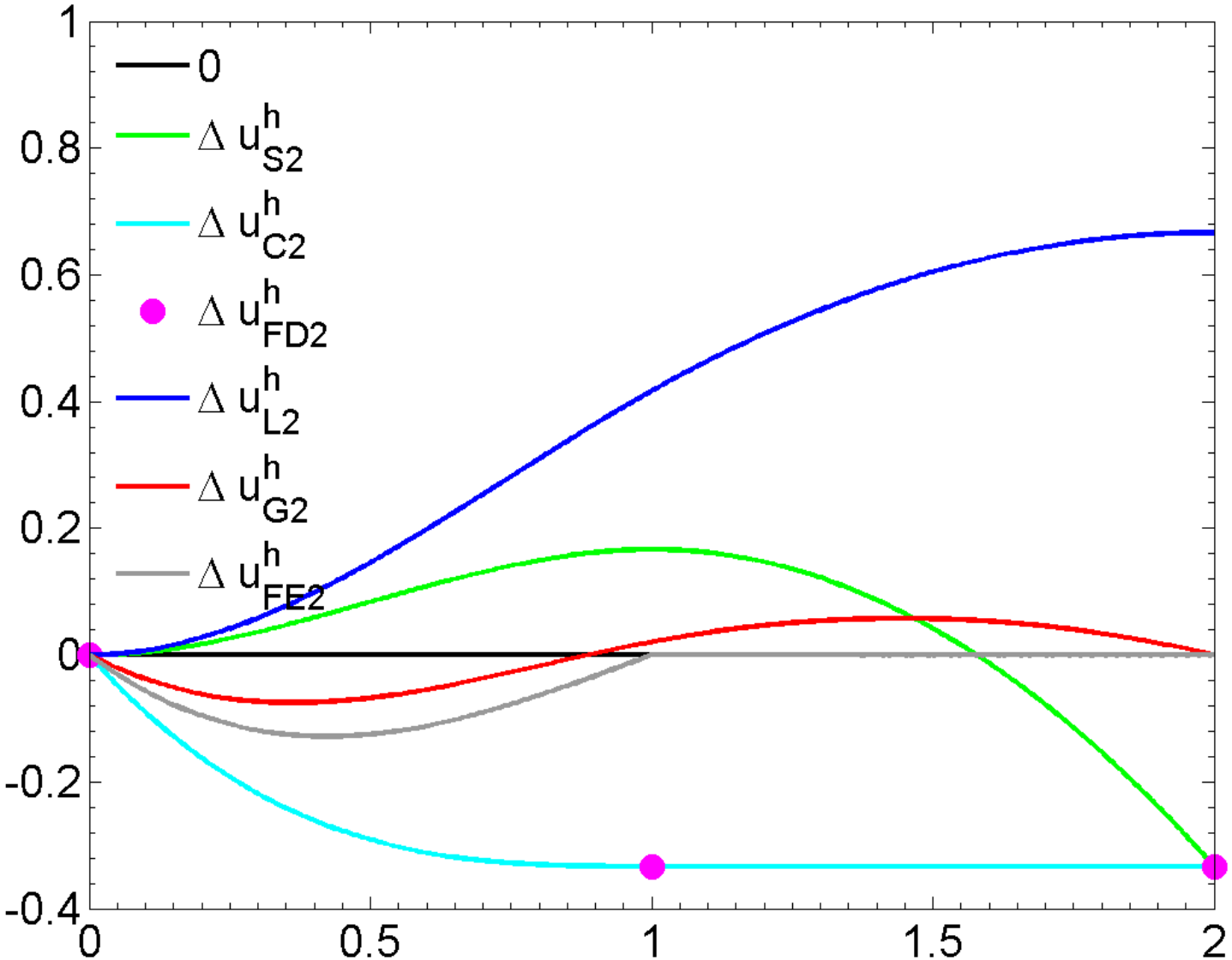
- As expected, this solution agrees with (234) and yields the same solution in (235):

$$u_{G2}^h = 1 + \frac{37}{24}x - \frac{3}{16}x^2$$

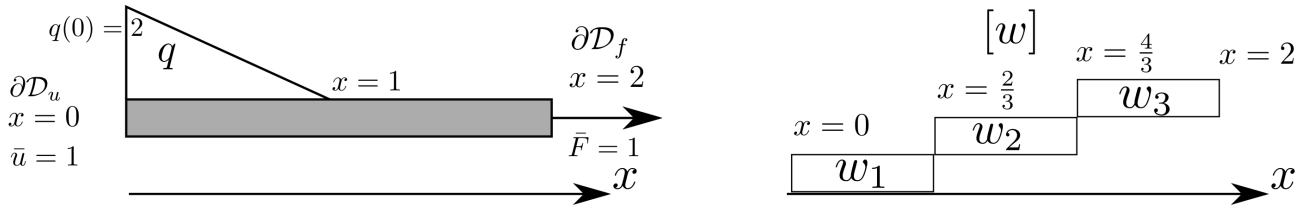
Bar example, $n = 2$, Comparison of solutions



Bar example, $n = 2$, Comparison of solutions



Bar example, $n = 3$, Subdomain method



- For $n = 3$ our trial functions are $[\phi] = \{x, x^2, x^3\}$ (*cf.* (196c)).
- For subdomain method $n = 3$ weight functions are $[w] = [w^f] = \left[\chi_{[0\ \frac{2}{3}]}, \chi_{[\frac{2}{3}\ \frac{4}{3}]}\chi_{[\frac{4}{3}\ 2]}\right]$.
- We plug $[\phi] = \{x, x^2, x^3\}$ and $[w] = [w^f] = \left[\chi_{[0\ \frac{2}{3}]}, \chi_{[\frac{2}{3}\ \frac{4}{3}]}\chi_{[\frac{4}{3}\ 2]}\right]$ into (185) to obtain:

$$\mathbf{K}_{1j} = \int_0^2 [\chi_{[0\ \frac{2}{3}]}].\{\frac{\mathrm{d}^2}{\mathrm{d}x^2}[x\ x^2\ x^3]\}\,\mathrm{d}x - \{[\chi_{[0\ \frac{2}{3}]}].\frac{\mathrm{d}}{\mathrm{d}x}[x\ x^2\ x^3]\}|_{x=2}$$

$$= \int_0^{\frac{2}{3}} [0\ 2\ 6x]\ \mathrm{d}x - \{[0].[1\ 4\ 12]\} = [0\ \frac{4}{3}\ \frac{4}{3}] - [0\ 0\ 0] = [0\ \frac{4}{3}\ \frac{4}{3}]$$

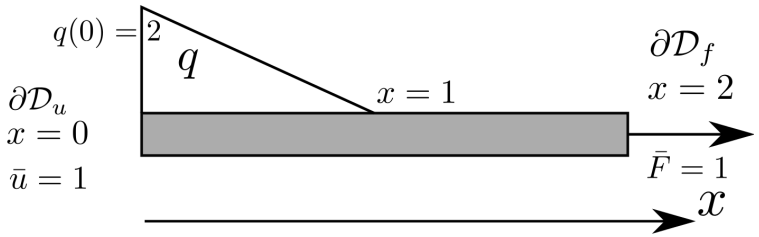
$$\mathbf{K}_{2j} = \int_0^2 [\chi_{[\frac{2}{3}\ \frac{4}{3}]}].\{\frac{\mathrm{d}^2}{\mathrm{d}x^2}[x\ x^2\ x^3]\}\,\mathrm{d}x - \{[\chi_{[\frac{2}{3}\ \frac{4}{3}]}].\frac{\mathrm{d}}{\mathrm{d}x}[x\ x^2\ x^3]\}|_{x=2}$$

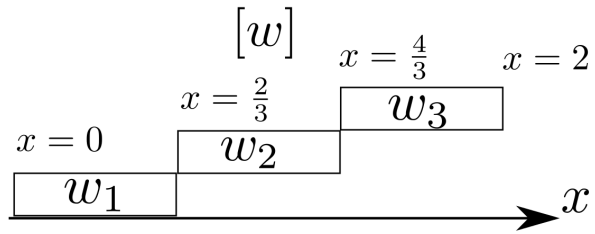
$$= \int_{\frac{2}{3}}^{\frac{4}{3}} [0\ 2\ 6x]\ \mathrm{d}x - \{[0].[1\ 4\ 12]\} = [0\ \frac{4}{3}\ 4] - [0\ 0\ 0] = [0\ \frac{4}{3}\ 4] \qquad (244)$$

$$\mathbf{K}_{3j} = \int_0^2 [\chi_{[\frac{4}{3}\ 2]}].\{\frac{\mathrm{d}^2}{\mathrm{d}x^2}[x\ x^2\ x^3]\}\,\mathrm{d}x - \{[\chi_{[\frac{4}{3}\ 2]}].\frac{\mathrm{d}}{\mathrm{d}x}[x\ x^2\ x^3]\}|_{x=2}$$

$$= \int_{\frac{4}{3}}^2 [0\ 2\ 6x]\ \mathrm{d}x - \{[1].[1\ 4\ 12]\} = [0\ \frac{4}{3}\ \frac{20}{3}] - [1\ 4\ 12] = [-1\ -\frac{8}{3}\ -\frac{16}{3}]$$

Bar example, $n = 3$, Subdomain method





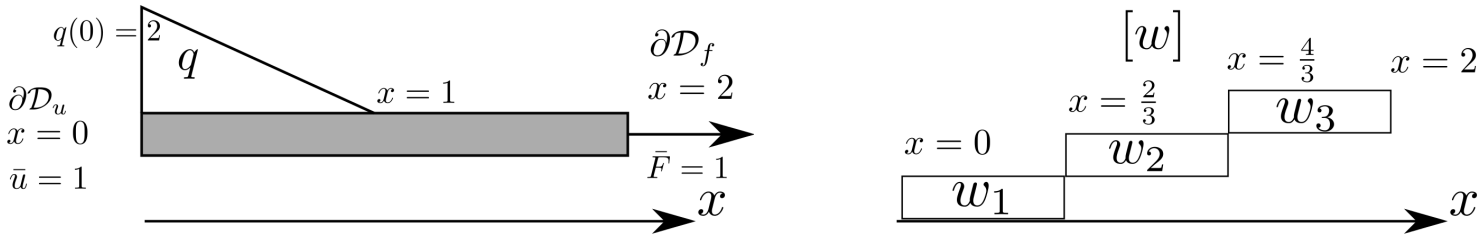
$$F_1 = -\int_0^1 [\chi_{[0\ \ \frac{2}{3}]}](2-2x)\,\mathrm{d}x - [\chi_{[0\ \ \frac{2}{3}]}]_{x=2} = -\int_0^{\frac{2}{3}} (2-2x)\,\mathrm{d}x - 0 = (-2).\frac{2}{3} + \left(\frac{2}{3}\right)^2 = -\frac{8}{9}$$

$$F_2 = -\int_0^1 [\chi_{[\frac{2}{3}\ \frac{4}{3}]}](2-2x)\,\mathrm{d}x - [\chi_{[\frac{2}{3}\ \ \frac{4}{3}]}]_{x=2} = -\int_{\frac{2}{3}}^{\min(1,\frac{4}{3})} (2-2x)\,\mathrm{d}x - 0$$

$$= -\frac{2}{3} + 1 - \left(\frac{2}{3}\right)^2 = -\frac{1}{9} \tag{245}$$

$$F_3 = -\int_0^1 [\chi_{[\frac{4}{3}\ \ 2]}](2-2x)\,\mathrm{d}x - [\chi_{[\frac{4}{3}\ \ 2]}]_{x=2} = 0 - 1 = -1$$

Bar example, $n = 3$, Subdomain method



- Equations (244) and (245) yield,

$$\mathbf{K} = \begin{bmatrix} 0 & \frac{4}{3} & \frac{4}{3} \\ 0 & \frac{4}{3} & 4 \\ -1 & -\frac{8}{3} & -\frac{16}{3} \end{bmatrix}, \quad \mathbf{F} = \begin{bmatrix} -\frac{8}{9} \\ -\frac{1}{9} \\ -1 \end{bmatrix} \tag{246}$$

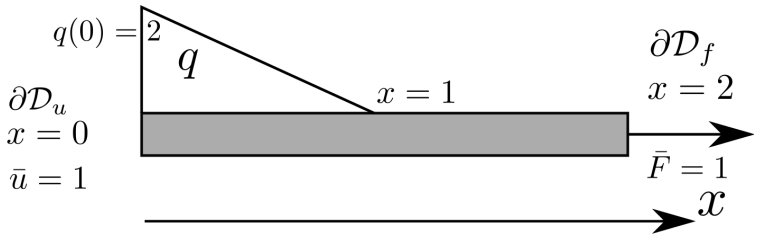
- From $\mathbf{Ka} = \mathbf{F}$ (125) and (246) we get,

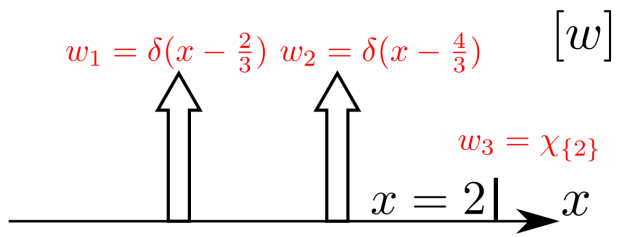
$$\mathbf{a} = \begin{bmatrix} 2 \\ -\frac{23}{24} \\ \frac{7}{24} \end{bmatrix} \tag{247}$$

- From $u^h = a_j \phi_j + \phi_p$ (117a), $[\phi] = \{x, x^2, x^3\}$ (*cf.* (196c)), and $\phi_p = 1$ (182) we have

$$\boxed{u^h_{S3} = 1 + 2x - \frac{23}{24}x^2 + \frac{7}{24}x^3} \tag{248}$$

Bar example, $n = 3$, Collocation method

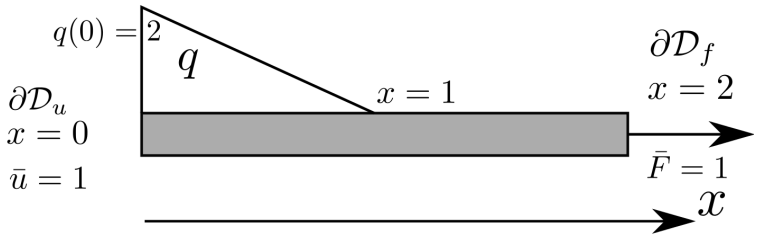


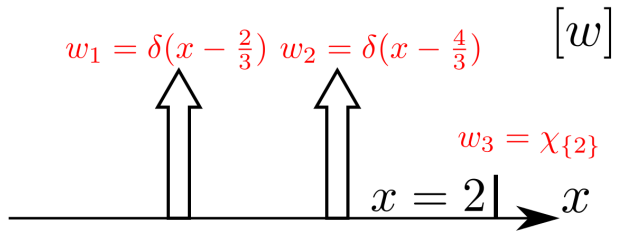


- For $n = 3$ our trial functions are $[\phi] = \{x, x^2, x^3\}$ (*cf.* (196c)).
- For this collocation method we choose two points inside the domain ($x = \frac{2}{3}, \frac{4}{3}$); the corresponding weight functions $w_1 = \delta(x - \frac{2}{3})$ and $w_2 = \delta(x - \frac{4}{3})$ act on $\mathcal{R}_i(\frac{2}{3})$, $\mathcal{R}_i(\frac{4}{3})$.
- The second collocation point is chosen on $\partial\mathcal{D}_f$ ($x = 2$); the corresponding weight function $w_3 = \chi_{\{2\}}$ only acts on $\mathcal{R}_f$ at $x = 2$.
- We plug these functions into (185) to obtain (also refer to (148), (149), and (149b))

$$
\begin{aligned}
\mathbf{K}_{1j} &= \int_0^2 [\delta(x - \frac{2}{3})].\{\frac{\mathrm{d}^2}{\mathrm{d}x^2}[x\ x^2\ x^3]\}\,\mathrm{d}x - \{[\delta(x - \frac{2}{3})].\frac{\mathrm{d}}{\mathrm{d}x}[x\ x^2\ x^3]\}|_{x=2} \\
&= [0\ 2\ 6x]_{x=\frac{2}{3}} - [0\ 0\ 0] = [0\ 2\ 4] \\
\mathbf{K}_{2j} &= \int_0^2 [\delta(x - \frac{4}{3})].\{\frac{\mathrm{d}^2}{\mathrm{d}x^2}[x\ x^2\ x^3]\}\,\mathrm{d}x - \{[\delta(x - \frac{4}{3})].\frac{\mathrm{d}}{\mathrm{d}x}[x\ x^2\ x^3]\}|_{x=2} \\
&= [0\ 2\ 6x]_{x=\frac{4}{3}} - [0\ 0\ 0] = [0\ 2\ 8] \qquad (249) \\
\mathbf{K}_{3j} &= \int_0^2 [\chi_{\{2\}}].\{\frac{\mathrm{d}^2}{\mathrm{d}x^2}[x\ x^2\ x^3]\}\,\mathrm{d}x - \{[\chi_{\{2\}}]\frac{\mathrm{d}}{\mathrm{d}x}[x\ x^2\ x^3]\}|_{x=2} = 0 - \{[1]\,[1\ 4\ 12]\} \\
&= [-1\ \ -4\ \ -12]
\end{aligned}
$$

Bar example, $n = 3$, Collocation method





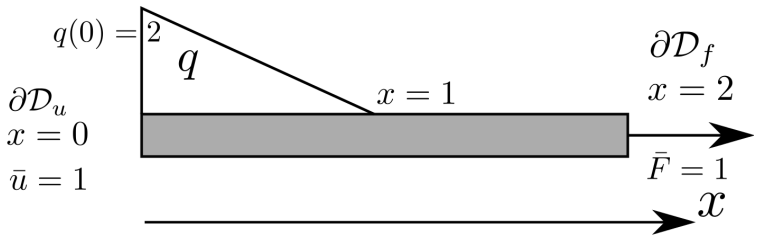
$$F_1 = -\int_0^1 [\delta(x - \frac{2}{3})](2 - 2x)\,\mathrm{d}x - [\delta(x - \frac{2}{3})]_{x=2} = -(2 - 2x)_{x=\frac{2}{3}} - 0 = -2 + (2).\frac{2}{3} = -\frac{2}{3}$$

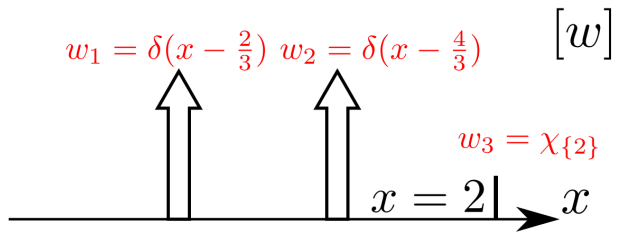
$$F_2 = -\int_0^1 [\delta(x - \frac{4}{3})](2 - 2x)\,\mathrm{d}x - [\delta(x - \frac{4}{3})]_{x=2} = 0 - 0 = 0 = 0$$

$$F_3 = -\int_0^1 [\chi_{\{2\}}].(2 - 2x)\,\mathrm{d}x - [\chi_{\{2\}}]_{x=2} = 0 - 1 = -1 \tag{250}$$

- For $n = 3$ our trial functions are $[\phi] = \{x, x^2, x^3\}$ (*cf.* (196c)).
- For this collocation method we choose two points inside the domain ($x = \frac{2}{3}, \frac{4}{3}$); the corresponding weight functions $w_1 = \delta(x - \frac{2}{3})$ and $w_2 = \delta(x - \frac{4}{3})$ act on $\mathcal{R}_i(\frac{2}{3})$, $\mathcal{R}_i(\frac{4}{3})$.
- The second collocation point is chosen on $\partial\mathcal{D}_f$ ($x = 2$); the corresponding weight function $w_3 = \chi_{\{2\}}$ only acts on $\mathcal{R}_f$ at $x = 2$.
- We plug these functions into (185) to obtain (also refer to (148), (149), and (149b))

Bar example, $n = 3$, Collocation method





- Equations (249) and (250) yield,

$$\mathbf{K} = \begin{bmatrix} 0 & 2 & 4 \\ 0 & 2 & 8 \\ -1 & -4 & -12 \end{bmatrix}, \quad \mathbf{F} = \begin{bmatrix} -\frac{2}{3} \\ 0 \\ -1 \end{bmatrix} \tag{251}$$

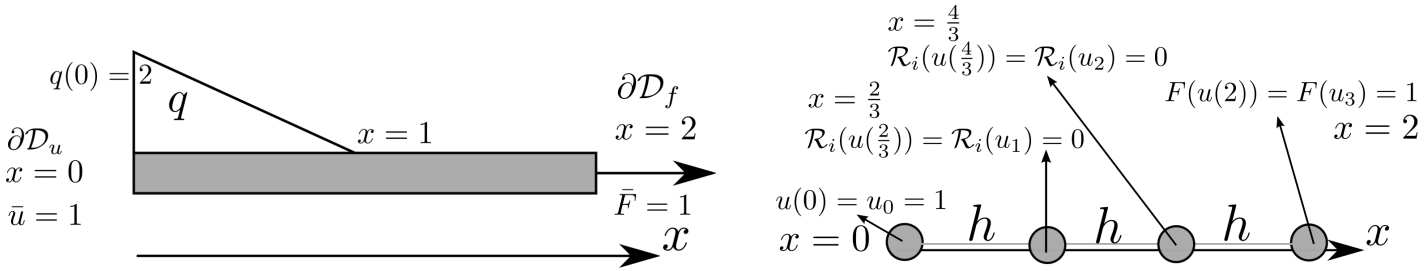
- From $\mathbf{Ka} = \mathbf{F}$ (125) and (251) we get,

$$\mathbf{a} = \begin{bmatrix} \frac{5}{3} \\ -\frac{2}{3} \\ \frac{1}{6} \end{bmatrix} \tag{252}$$

- From $u^h = a_j \phi_j + \phi_p$ (117a), $[\phi] = \{x, x^2, x^3\}$ (*cf.* (196c)), and $\phi_p = 1$ (182) we have

$$\boxed{u^h_{C3} = 1 + \frac{5}{3}x - \frac{2}{3}x^2 + \frac{1}{6}x^3} \tag{253}$$

Bar example, $n = 3$, Finite Difference



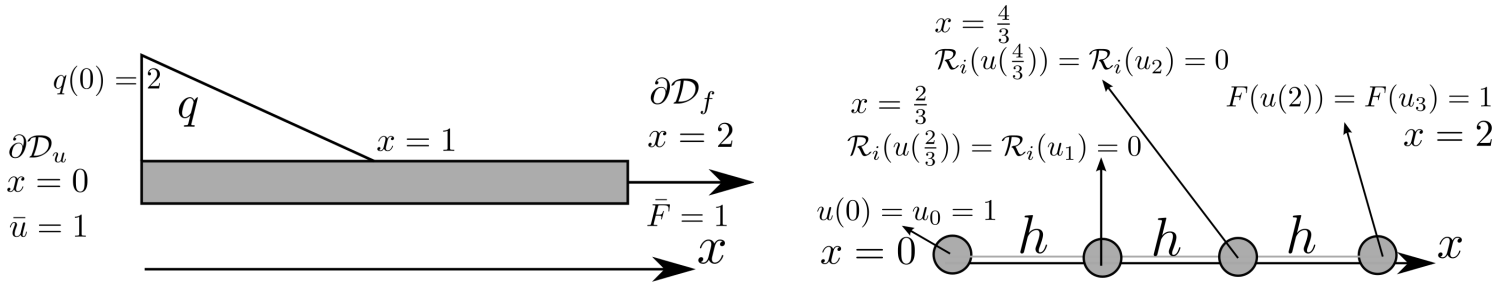
- In Finite Difference method $n = 3$ we need three unknowns.
- The essential boundary condition at $x = 0$ is $u(x = 0) := u_0 = 1$
- For first and second unknowns $u_1 = u(x = \frac{2}{3})$ and $u_2 = u(x = \frac{4}{3})$ we enforce approximate FD for $\mathcal{R}_i(x = \frac{2}{3}) = 0$ and $\mathcal{R}_i(x = \frac{42}{3}) = 0$.
- Third unknown is $u_3 = u(x = 2)$; we enforce approximate FD for $\mathcal{R}_f(x = 2) = 0$.
- The equations are:

$$\mathcal{R}_i(x = \frac{2}{3}) = \frac{\mathrm{d}^2 u}{\mathrm{d}x^2} + q(\frac{2}{3}) \approx \frac{u_0 - 2u_1 + u_2}{h^2} + (2 - \frac{4}{3})$$
$$= \frac{u_0 - 2u_1 + u_2}{(2/3)^2} + \frac{2}{3} = \frac{9}{4}(1) - \frac{9}{2}u_1 + \frac{9}{4}u_2 + \frac{2}{3} = 0 \qquad (254a)$$

$$\mathcal{R}_i(x = \frac{4}{3}) = \frac{\mathrm{d}^2 u}{\mathrm{d}x^2} + q(\frac{4}{3}) \approx \frac{u_1 - 2u_2 + u_3}{h^2} + 0$$
$$= \frac{u_1 - 2u_2 + u_3}{(2/3)^2} = \frac{9}{4}u_1 - \frac{9}{2}u_2 + \frac{9}{4}u_3 = 0 \qquad (254b)$$

$$\mathcal{R}_f(x = 2) = \bar{F} - \frac{\mathrm{d}u}{\mathrm{d}x} \approx 1 - \frac{u_3 - u_2}{h} = 1 - \frac{u_3 - u_2}{(2/3)} = 1 - \frac{3}{2}u_3 + \frac{3}{2}u_2 = 0 \qquad (254c)$$

Bar example, $n = 3$, Finite Difference



- Unknowns u_i are obtained from (254),

$$\mathbf{K}\begin{bmatrix} u_1 \\ u_2 \\ u_3 \end{bmatrix} = \mathbf{F}, \quad for \quad \mathbf{K} = \begin{bmatrix} -\frac{9}{2} & \frac{9}{4} & 0 \\ \frac{9}{4} & -\frac{9}{2} & \frac{9}{4} \\ 0 & \frac{3}{2} & -\frac{3}{2} \end{bmatrix}, \quad \mathbf{F} = \begin{bmatrix} -\frac{35}{12} \\ 0 \\ -1 \end{bmatrix} \tag{255}$$

- The solution denoted by u^h_{FD3} is:

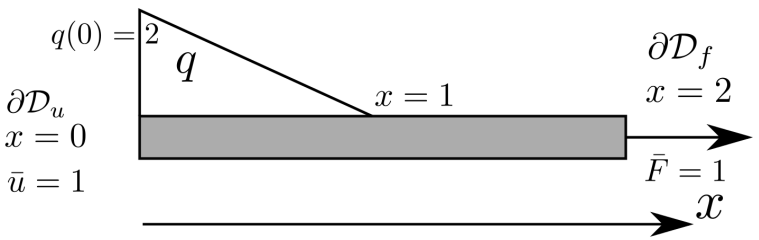
$$u(0) = u_0 = 1 \tag{256a}$$

$$u(\frac{2}{3}) = u_1 = \frac{53}{27} \tag{256b}$$

$$u(\frac{4}{3}) = u_2 = \frac{71}{27} \tag{256c}$$

$$u(2) = u_3 = \frac{89}{27} \tag{256d}$$

Bar example, $n = 3$, Least Square method

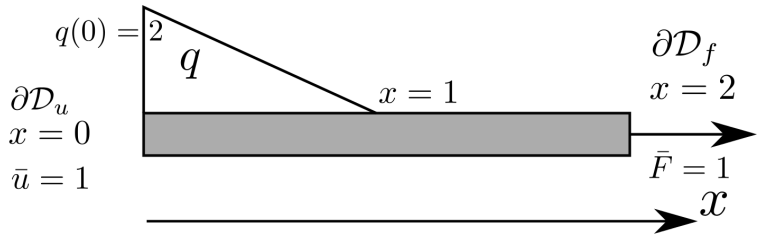


- For $n = 3$ our trial function is $[\phi] = \{x \; x^2 \; x^3\}$ (*cf.* (196c)).
- For least square method, the corresponding weight function are (*cf.* (159)):

$$\mathbf{w} = L_M(\boldsymbol{\phi}) \qquad \Rightarrow \qquad \mathbf{w} = \frac{\mathrm{d}^2[\boldsymbol{\phi}]}{\mathrm{d}x^2} = \frac{\mathrm{d}^2 \begin{bmatrix} x \\ x^2 \\ x^3 \end{bmatrix}}{\mathrm{d}x^2} = \begin{bmatrix} 0 \\ 2 \\ 6x \end{bmatrix}$$

$$\mathbf{w}^f = (-L_f(\boldsymbol{\phi})) \qquad \Rightarrow \qquad \mathbf{w}^f = -\frac{\mathrm{d}[\boldsymbol{\phi}]}{\mathrm{d}x} = -\frac{\mathrm{d} \begin{bmatrix} x \\ x^2 \\ x^3 \end{bmatrix}}{\mathrm{d}x} = - \begin{bmatrix} 1 \\ 2x \\ 3x^2 \end{bmatrix}$$

Bar example, $n = 3$, Least Square method

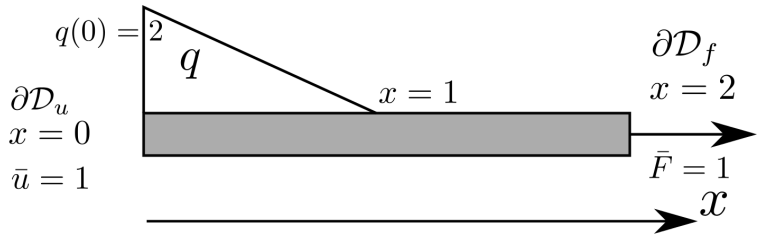


- We plug $[\phi] = \begin{bmatrix} x \\ x^2 \\ x^3 \end{bmatrix}$ and (257) into (185) to obtain:

$$\mathbf{K} = \int_0^2 \overbrace{\begin{bmatrix} 0 \\ 2 \\ 6x \end{bmatrix}}^{\mathbf{w}} .\{\frac{\mathrm{d}^2}{\mathrm{d}x^2}[x\ x^2\ x^3]\}\,\mathrm{d}x - \{\overbrace{\begin{bmatrix} -1 \\ -2x \\ -3x^2 \end{bmatrix}}^{\mathbf{w}^f} .\frac{\mathrm{d}[x\ x^2\ x^3]}{\mathrm{d}x}\}|_{x=2} = \int_0^2 \begin{bmatrix} 0 \\ 2 \\ 6x \end{bmatrix} .\{[0\ 2\ 6x]\}\,\mathrm{d}x$$
$$+ \{\begin{bmatrix} 1 \\ 2x \\ 3x^2 \end{bmatrix} .[1\ 2x\ 3x^2]\}|_{x=2} = \begin{bmatrix} 0 & 0 & 0 \\ 0 & 8 & 24 \\ 0 & 24 & 96 \end{bmatrix} + \begin{bmatrix} 1 & 4 & 12 \\ 4 & 16 & 48 \\ 12 & 48 & 144 \end{bmatrix} = \begin{bmatrix} 1 & 4 & 12 \\ 4 & 24 & 72 \\ 12 & 72 & 240 \end{bmatrix} \tag{258}$$

$$\mathbf{F} = -\int_0^1 \overbrace{\begin{bmatrix} 0 \\ 2 \\ 6x \end{bmatrix}}^{\mathbf{w}} .(2 - 2x)\,\mathrm{d}x - \overbrace{\begin{bmatrix} -1 \\ -2x \\ -3x^2 \end{bmatrix}}^{\mathbf{w}^f}_{x=2} = -\begin{bmatrix} 0 \\ 4-2 \\ 6-4 \end{bmatrix} - \begin{bmatrix} -1 \\ -4 \\ -12 \end{bmatrix} = \begin{bmatrix} 1 \\ 2 \\ 10 \end{bmatrix} \tag{259}$$

Bar example, $n = 3$, Least Square method



- Equations (258) and (259) yield,

$$\mathbf{K} = \begin{bmatrix} 1 & 4 & 12 \\ 4 & 24 & 72 \\ 12 & 72 & 240 \end{bmatrix}, \quad \mathbf{F} = \begin{bmatrix} 1 \\ 2 \\ 10 \end{bmatrix} \tag{260}$$

- From $\mathbf{Ka} = \mathbf{F}$ (125) and (260) we get,

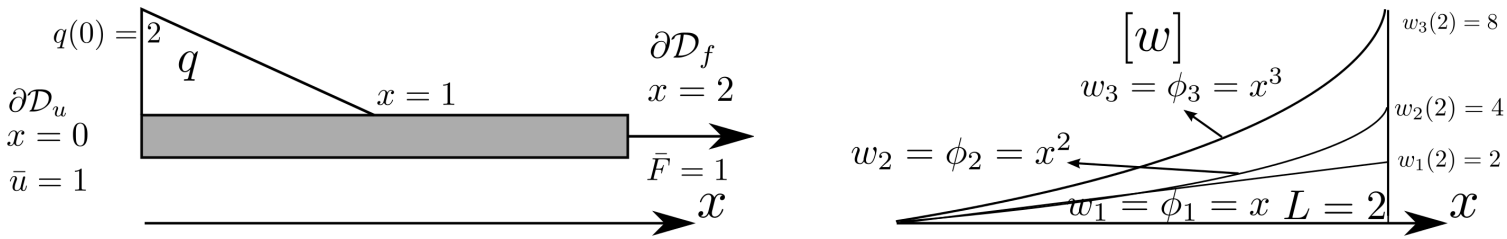
$$\mathbf{a} = \begin{bmatrix} 2 \\ -\frac{3}{4} \\ \frac{1}{6} \end{bmatrix} \tag{261}$$

- From $u^h = a_j \phi_j + \phi_p$ (117a), $[\phi] = \{x, x^2, x^3\}$ (*cf.* (196c)), and $\phi_p = 1$ (182) we have

$$\boxed{u^h_{L3} = 1 + 2x - \frac{3}{4}x^2 + \frac{1}{6}x^3} \tag{262}$$

- As we demonstrated that direct stationary condition on R^2 yields the same solution for $n = 1$ and $n = 2$, we don't verify it further for $n = 3$.

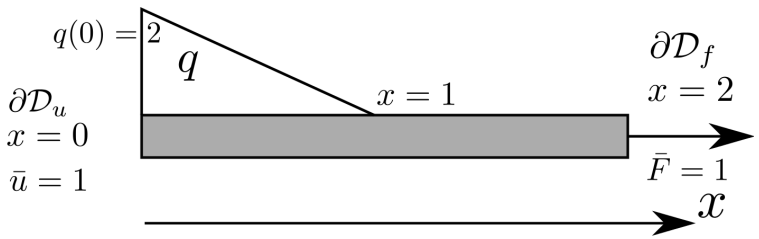
Bar example, $n = 3$, Galerkin method

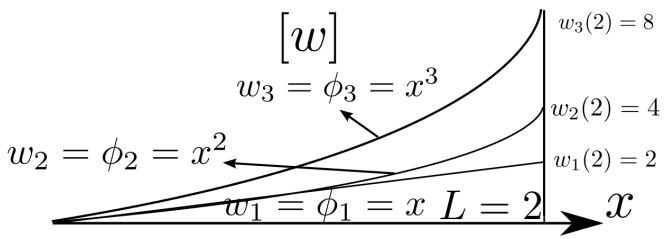


- For $n = 3$ our trial functions are $[\phi] = \{x, x^2, x^3\}$ (*cf.* (196c)).
- For Galerkin method $n = 3$ weight functions are $[w] = [w^f] = [\phi] = \{x, x^2, x^3\}$.
- We plug $[\phi] = [w] = [w^f] = \{x, x^2, x^3\}$ into (185) to obtain:

$$
\begin{aligned}
\mathbf{K} &= \int_0^2 \begin{bmatrix} x \\ x^2 \\ x^3 \end{bmatrix} . \{ \frac{\mathrm{d}^2}{\mathrm{d}x^2} [x \; x^2 \; x^3] \} \, \mathrm{d}x - \{ \begin{bmatrix} x \\ x^2 \\ x^3 \end{bmatrix} . \frac{\mathrm{d}}{\mathrm{d}x} [x \; x^2 \; x^3] \} |_{x=2} \\
&= \int_0^2 \begin{bmatrix} 0 & 2x & 6x^2 \\ 0 & 2x^2 & 6x^3 \\ 0 & 12x^3 & 6x^4 \end{bmatrix} \, \mathrm{d}x - \{ \begin{bmatrix} x & 2x^2 & 3x^3 \\ x^2 & 2x^3 & 3x^4 \\ x^3 & 2x^4 & 3x^5 \end{bmatrix} \} |_{x=2} \\
&= \begin{bmatrix} 0 & 4 & 16 \\ 0 & \frac{16}{3} & 24 \\ 0 & 8 & \frac{192}{5} \end{bmatrix} - \begin{bmatrix} 2 & 8 & 24 \\ 4 & 16 & 48 \\ 8 & 32 & 96 \end{bmatrix} = \begin{bmatrix} -2 & -4 & -8 \\ -4 & -\frac{32}{3} & -24 \\ -8 & -24 & -\frac{288}{5} \end{bmatrix}
\end{aligned}
\tag{263}
$$

Bar example, $n = 3$, Galerkin method

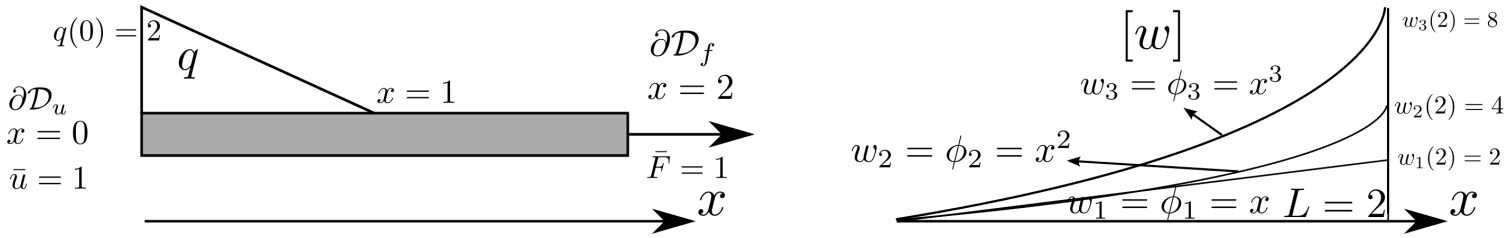




$$\mathbf{F} = -\int_0^1 \begin{bmatrix} x \\ x^2 \\ x^3 \end{bmatrix} .(2 - 2x)\, \mathrm{d}x - \left. \begin{bmatrix} x \\ x^2 \\ x^3 \end{bmatrix} \right|_{x=2}$$

$$-\int_0^1 \begin{bmatrix} 2x - 2x^2 \\ 2x^2 - 2x^3 \\ 2x^3 - 2x^4 \end{bmatrix} \mathrm{d}x - \begin{bmatrix} 2 \\ 4 \\ 8 \end{bmatrix} = \begin{bmatrix} -1 + \frac{2}{3} - 2 \\ -\frac{2}{3} + \frac{1}{2} - 4 \\ -\frac{1}{2} + \frac{2}{5} - 8 \end{bmatrix} = \begin{bmatrix} -\frac{7}{3} \\ -\frac{25}{6} \\ -\frac{81}{10} \end{bmatrix} \tag{264}$$

Bar example, $n = 3$, Galerkin method



- Equations (263) and (264) yield,

$$\mathbf{K} = \begin{bmatrix} -2 & -4 & -8 \\ -4 & -\frac{32}{3} & -24 \\ -8 & -24 & -\frac{288}{5} \end{bmatrix}, \quad \mathbf{F} = \begin{bmatrix} -\frac{7}{3} \\ -\frac{25}{6} \\ -\frac{81}{10} \end{bmatrix} \tag{265}$$

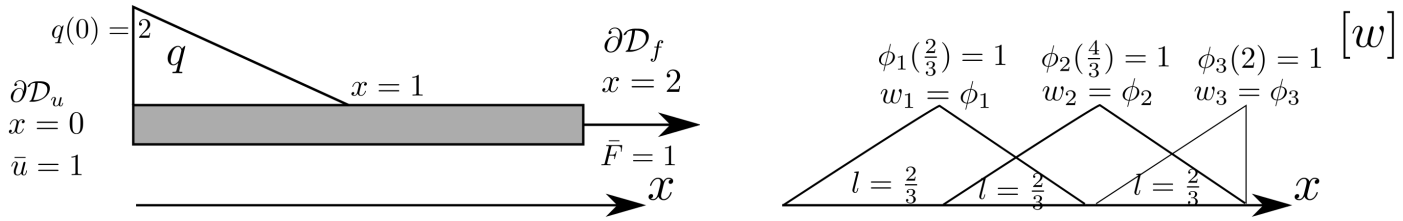
- From $\mathbf{Ka} = \mathbf{F}$ (125) and (265) we get,

$$\mathbf{a} = \begin{bmatrix} \frac{15}{8} \\ -\frac{11}{16} \\ \frac{1}{6} \end{bmatrix} \tag{266}$$

- From $u^h = a_j \phi_j + \phi_p$ (117a), $[\phi] = \{x, x^2, x^3\}$ (*cf.* (196c)), and $\phi_p = 1$ (182) we have

$$\boxed{u^h_{G3} = 1 + \frac{15}{8}x - \frac{11}{16}x^2 + \frac{1}{6}x^3} \tag{267}$$

Bar example, $n = 3$, Galerkin Finite Element Method



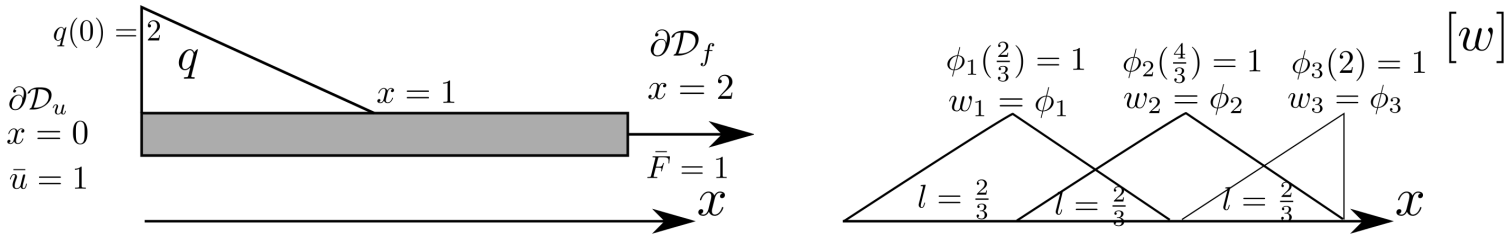
- The 1D hat shape functions will form a piece-wise linear solution which is not C^2 needed for the weighted residual method (terms $\frac{\mathrm{d}^2\phi_i}{\mathrm{d}x^2}$ in weighted residual are not functions).
- For example $\frac{\mathrm{d}^2\phi_1}{\mathrm{d}x^2}(\frac{2}{3}) = -3\delta(x - \frac{2}{3})$.
- Even $\frac{\mathrm{d}\phi_1}{\mathrm{d}x}(\frac{2}{3})$ is not defined, but all the integrals in the weak statement can be evaluated.
- We need to solve the problem with these types of trial functions in weak statement.
- For $n = 3$ as shown in the figure the trial and weight functions and their derivatives are :

$$\phi_1 = w_1 = \begin{cases} \frac{3}{2}x & 0 \le x < \frac{2}{3} \\ 2 - \frac{3}{2}x & \frac{2}{3} \le x < \frac{4}{3} \\ 0 & \frac{4}{3} \le x < 2 \end{cases} \quad \Rightarrow \quad \phi_1' = w_1' = \begin{cases} \frac{3}{2} & 0 \le x < \frac{2}{3} \\ -\frac{3}{2} & \frac{2}{3} \le x < \frac{4}{3} \\ 0 & \frac{4}{3} \le x < 2 \end{cases} \tag{268a}$$

$$\phi_2 = w_2 = \begin{cases} 0 & 0 \le x < \frac{2}{3} \\ \frac{3}{2}x - 1 & \frac{2}{3} \le x < \frac{4}{3} \\ 3 - \frac{3}{2}x & \frac{4}{3} \le x < 2 \end{cases} \quad \Rightarrow \quad \phi_2' = w_2' = \begin{cases} 0 & 0 \le x < \frac{2}{3} \\ \frac{3}{2} & \frac{2}{3} \le x < \frac{4}{3} \\ -\frac{3}{2} & \frac{4}{3} \le x < 2 \end{cases} \tag{268b}$$

$$\phi_3 = w_3 = \begin{cases} 0 & 0 \le x < \frac{2}{3} \\ 0 & \frac{2}{3} \le x < \frac{4}{3} \\ \frac{3}{2}x - 2 & \frac{4}{3} \le x < 2 \end{cases} \quad \Rightarrow \quad \phi_3' = w_3' = \begin{cases} 0 & 0 \le x < \frac{2}{3} \\ 0 & \frac{2}{3} \le x < \frac{4}{3} \\ \frac{3}{2} & \frac{4}{3} \le x < 2 \end{cases} \tag{268c}$$

Bar example, $n = 3$, Galerkin Finite Element Method



- Note that each trial function takes the value of 1 at its corresponding point: $\phi_1(1) = \phi_2(2) = \phi_3(3) = 1$ (*cf.* (268))
- We plug (268) into (191) (weak statement) to obtain:

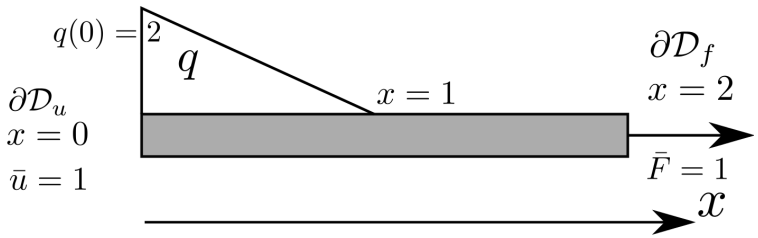
$$\mathbf{K} = \int_0^2 \frac{\mathrm{d}}{\mathrm{d}x}\begin{bmatrix}\phi_1\\ \phi_2\\ \phi_3\end{bmatrix} \cdot \frac{\mathrm{d}}{\mathrm{d}x}\begin{bmatrix}\phi_1 & \phi_2 & \phi_3\end{bmatrix} \mathrm{d}x = \int_0^2 \begin{bmatrix}\phi_1'\phi_1' & \phi_1'\phi_2' & \phi_1'\phi_3'\\ \phi_2'\phi_1' & \phi_2'\phi_2' & \phi_2'\phi_3'\\ \phi_3'\phi_1' & \phi_3'\phi_2' & \phi_3'\phi_3'\end{bmatrix} \mathrm{d}x$$

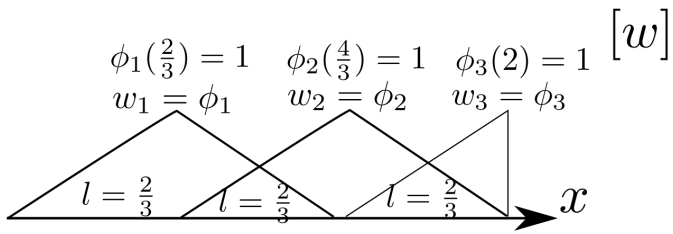
$$= \int_0^{\frac{2}{3}} \begin{bmatrix}\frac{3}{2}\cdot\frac{3}{2} & \frac{3}{2}\cdot(0) & \frac{3}{2}\cdot(0)\\ (0)\cdot\frac{3}{2} & (0)\cdot(0) & (0)\cdot(0)\\ (0)\cdot\frac{3}{2} & (0)\cdot(0) & (0)\cdot(0)\end{bmatrix} \mathrm{d}x + \int_{\frac{2}{3}}^{\frac{4}{3}} \begin{bmatrix}-\frac{3}{2}\cdot-\frac{3}{2} & -\frac{3}{2}\cdot\frac{3}{2} & -\frac{3}{2}\cdot(0)\\ \frac{3}{2}\cdot-\frac{3}{2} & \frac{3}{2}\cdot\frac{3}{2} & \frac{3}{2}\cdot(0)\\ (0)\cdot-\frac{3}{2} & (0)\cdot\frac{3}{2} & (0)\cdot(0)\end{bmatrix} \mathrm{d}x$$

$$+ \int_{\frac{4}{3}}^{2} \begin{bmatrix}(0)\cdot(0) & (0)\cdot-\frac{3}{2} & (0)\cdot\frac{3}{2}\\ -\frac{3}{2}\cdot(0) & -\frac{3}{2}\cdot-\frac{3}{2} & -\frac{3}{2}\cdot\frac{3}{2}\\ \frac{3}{2}\cdot(0) & \frac{3}{2}\cdot-\frac{3}{2} & \frac{3}{2}\cdot\frac{3}{2}\end{bmatrix} \mathrm{d}x = \begin{bmatrix}\frac{3}{2} & 0 & 0\\ 0 & 0 & 0\\ 0 & 0 & 0\end{bmatrix} + \begin{bmatrix}\frac{3}{2} & -\frac{3}{2} & 0\\ -\frac{3}{2} & \frac{3}{2} & 0\\ 0 & 0 & 0\end{bmatrix}$$

$$+ \begin{bmatrix}0 & 0 & 0\\ 0 & \frac{3}{2} & -\frac{3}{2}\\ 0 & -\frac{3}{2} & \frac{3}{2}\end{bmatrix} = \begin{bmatrix}3 & -\frac{3}{2} & 0\\ -\frac{3}{2} & 3 & -\frac{3}{2}\\ 0 & -\frac{3}{2} & \frac{3}{2}\end{bmatrix} \tag{269}$$

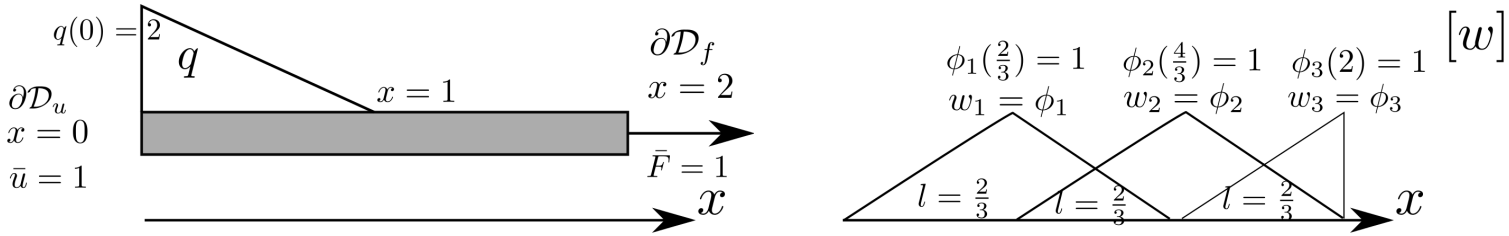
Bar example, $n = 3$, Galerkin Finite Element Method





$$
\begin{aligned}
\mathbf{F} &= \int_0^2 \begin{bmatrix} \phi_1 \\ \phi_2 \\ \phi_3 \end{bmatrix} .q(x)\,\mathrm{d}x + \left\{ \begin{bmatrix} \phi_1 \\ \phi_2 \\ \phi_3 \end{bmatrix} \right\}_{x=2} \\
&= \int_0^{\frac{2}{3}} \begin{bmatrix} \frac{3}{2}x \\ 0 \\ 0 \end{bmatrix} .(2-2x)\,\mathrm{d}x + \int_{\frac{2}{3}}^{\min(1,\frac{4}{3})} \begin{bmatrix} 2-\frac{3}{2}x \\ \frac{3}{2}x-1 \\ 0 \end{bmatrix} .(2-2x)\,\mathrm{d}x + \int_{\frac{4}{3}}^{2} \begin{bmatrix} 0 \\ 3-\frac{3}{2}x \\ \frac{3}{2}x-2 \end{bmatrix} .0\,\mathrm{d}x + \begin{bmatrix} 0 \\ 0 \\ 1 \end{bmatrix} \\
&= \int_0^{\frac{2}{3}} \begin{bmatrix} 3x-3x^2 \\ 0 \\ 0 \end{bmatrix} \mathrm{d}x + \int_{\frac{2}{3}}^{1} \begin{bmatrix} 3x^2-7x+4 \\ -3x^2+5x-2 \\ 0 \end{bmatrix} \mathrm{d}x + \begin{bmatrix} 0 \\ 0 \\ 1 \end{bmatrix} \\
&= \begin{bmatrix} \frac{2}{3}-\frac{8}{27} \\ 0 \\ 0 \end{bmatrix} + \begin{bmatrix} \frac{19}{27}-\frac{35}{18}+\frac{4}{3} \\ -\frac{19}{27}+\frac{25}{18}-\frac{2}{3} \\ 0 \end{bmatrix} + \begin{bmatrix} 0 \\ 0 \\ 1 \end{bmatrix} = \begin{bmatrix} \frac{25}{54} \\ \frac{1}{54} \\ 1 \end{bmatrix}
\end{aligned} \tag{270}
$$

Bar example, $n = 3$, Galerkin Finite Element Method



- Equations (269) and (270) yield,

$$\mathbf{K} = \begin{bmatrix} 3 & -\frac{3}{2} & 0 \\ -\frac{3}{2} & 3 & -\frac{3}{2} \\ 0 & -\frac{3}{2} & \frac{3}{2} \end{bmatrix}, \quad \mathbf{F} = \begin{bmatrix} \frac{25}{54} \\ \frac{1}{54} \\ 1 \end{bmatrix} \tag{271}$$

- From $\mathbf{Ka} = \mathbf{F}$ (125) and (271) we get,

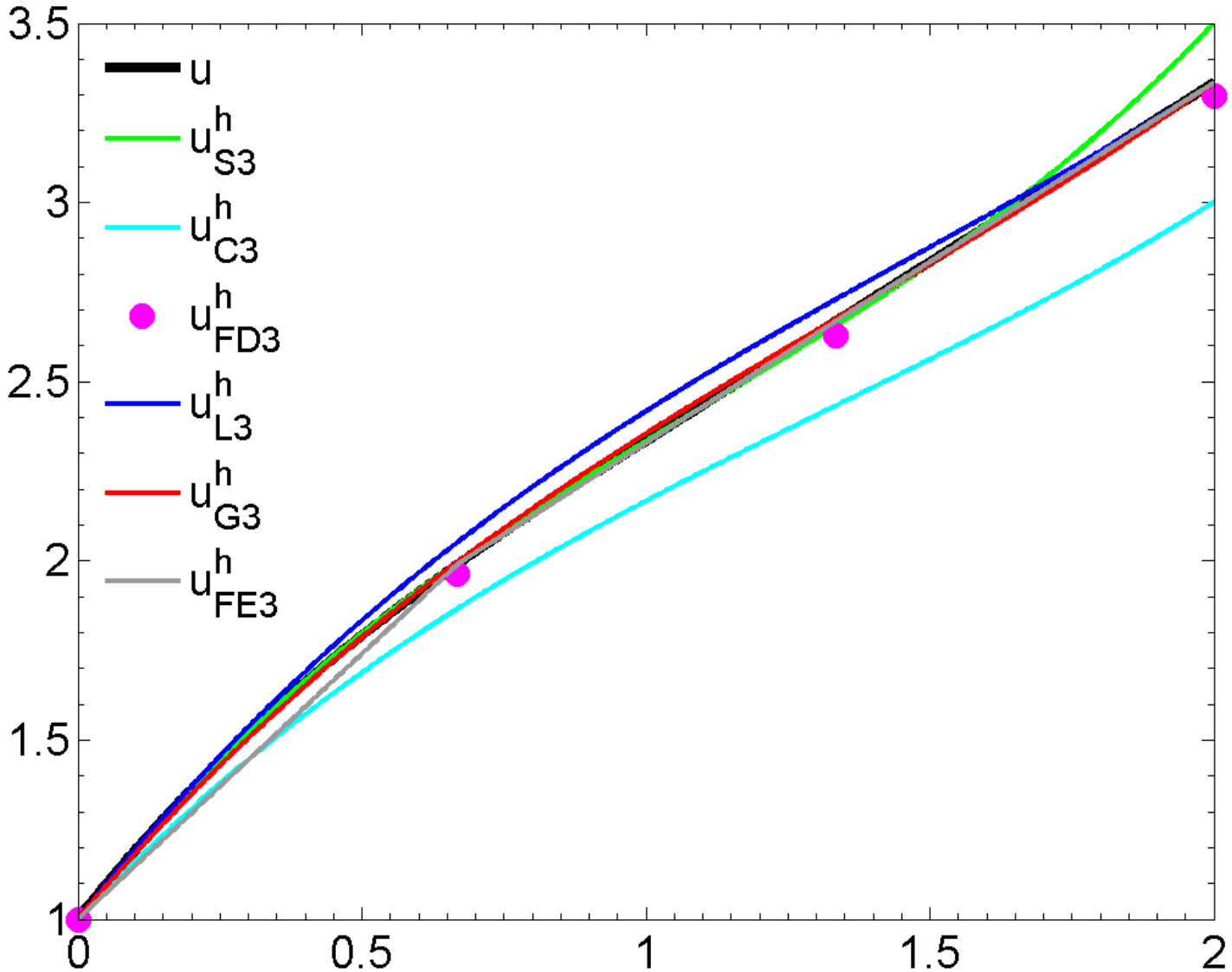
$$\mathbf{a} = \begin{bmatrix} \frac{81}{80} \\ \frac{5}{3} \\ \frac{7}{3} \end{bmatrix} \tag{272}$$

- From $u^h = a_j \phi_j + \phi_p$ (117a) and $\phi_p = 1$ (182) we have

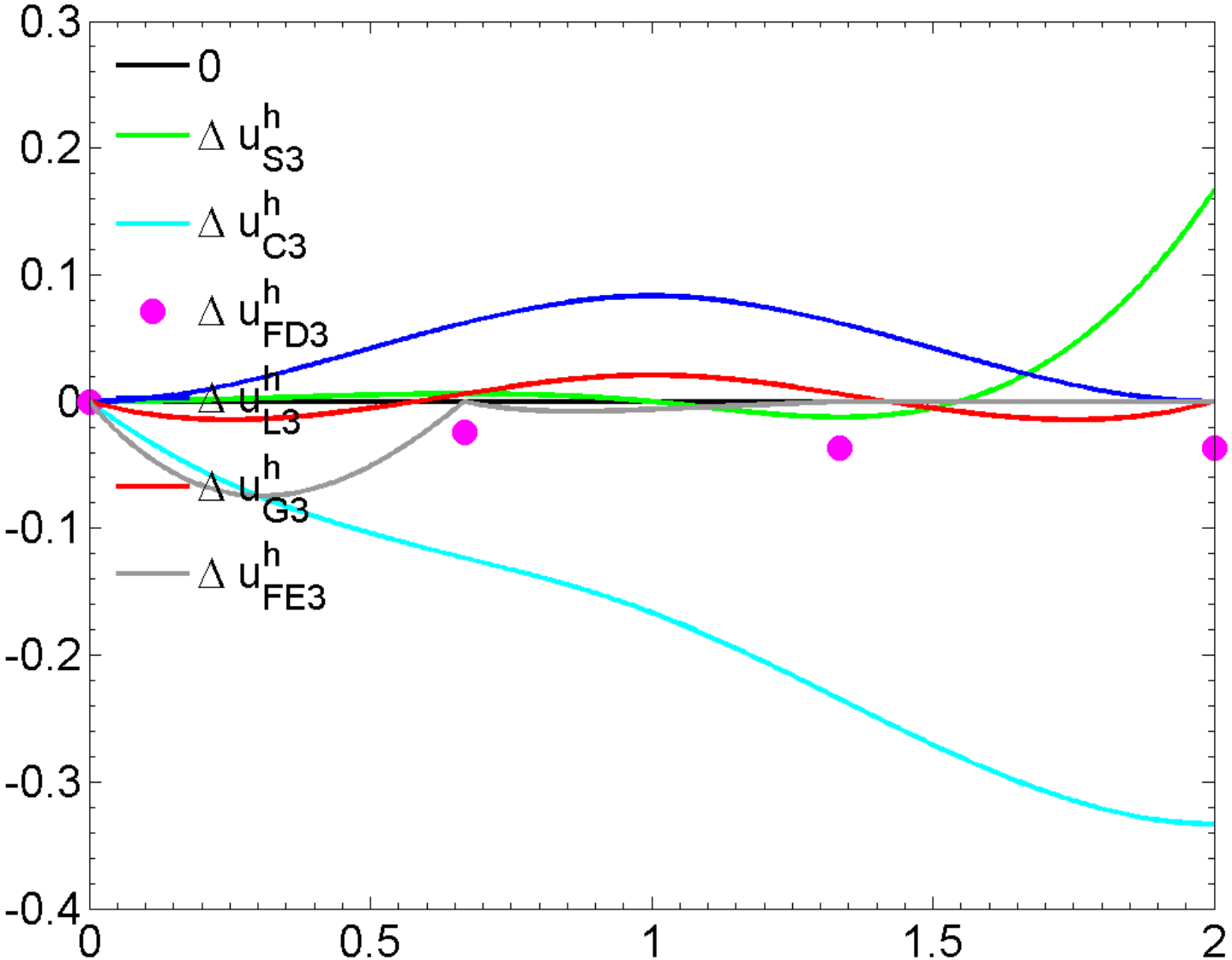
$$\boxed{u^h_{FE3} = 1 + \frac{80}{81}\phi_1 + \frac{5}{3}\phi_2 + \frac{7}{3}\phi_3} \tag{273}$$

- What is $u^h_{FE3}(\frac{2}{3})$?
- What is $u^h_{FE3}(\frac{4}{3})$?
- What is $u^h_{FE3}(2)$?

Bar example, $n = 3$, Comparison of solutions



Bar example, $n = 3$, Comparison of solutions

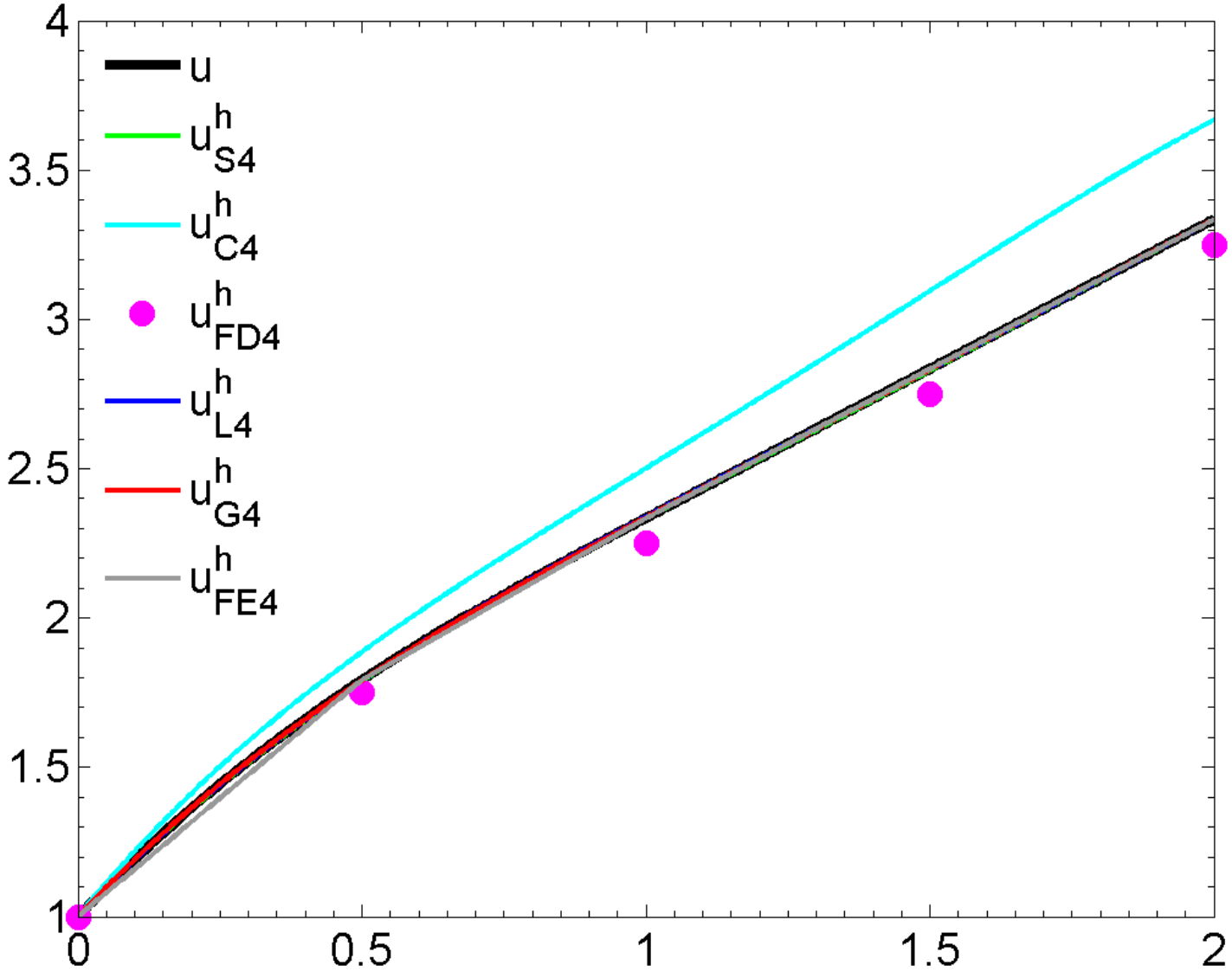


Bar example, $n = 4$

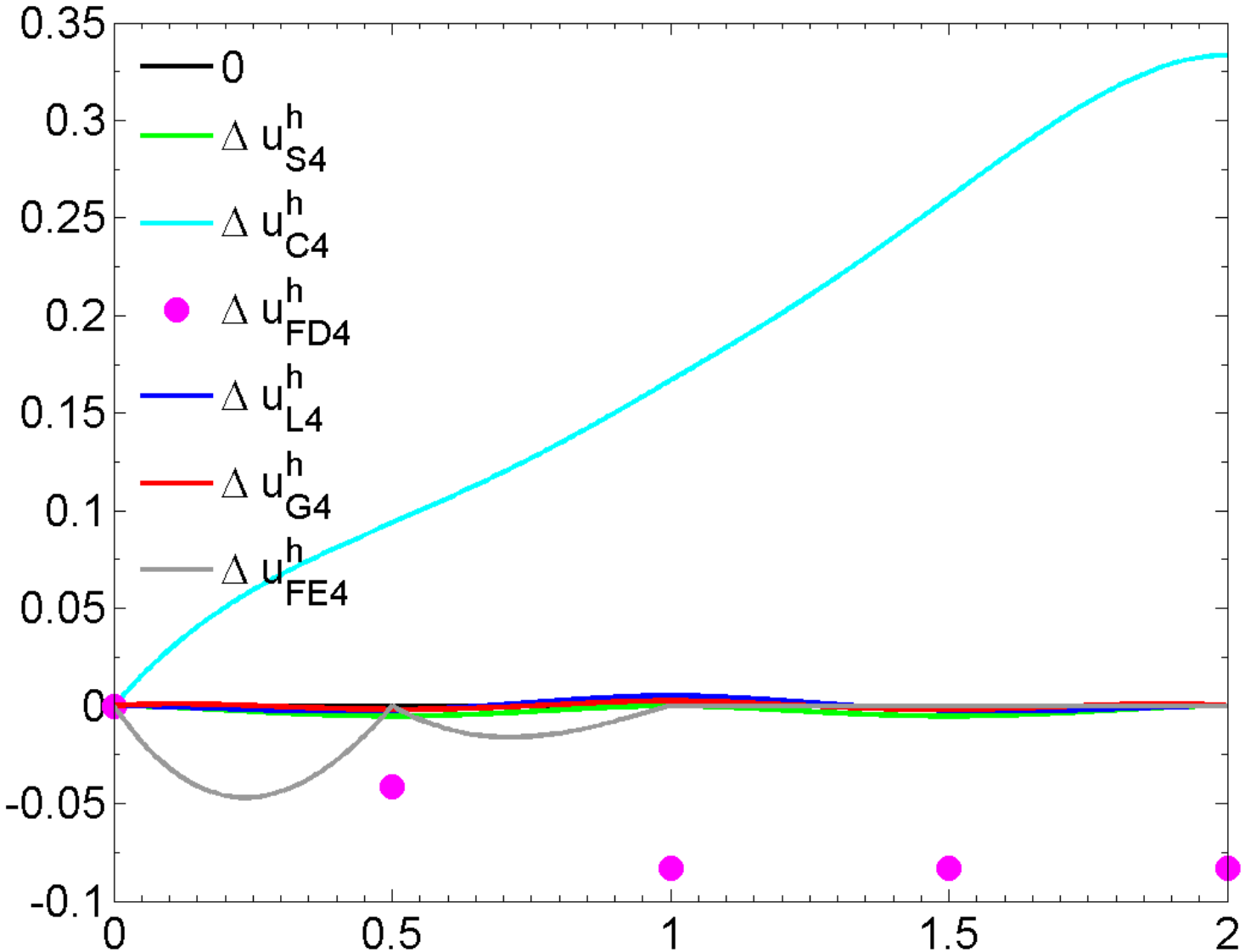
	S	C	FD
$\mathbf{K}$	$\begin{bmatrix} 0 & 1 & \frac{3}{4} & \frac{1}{2} \\ 0 & 1 & \frac{9}{4} & \frac{7}{2} \\ 0 & 1 & \frac{15}{4} & \frac{19}{2} \\ -1 & -3 & -\frac{27}{4} & -\frac{27}{2} \end{bmatrix}$	$\begin{bmatrix} 0 & 2 & 3 & 3 \\ 0 & 2 & 6 & 12 \\ 0 & 2 & 9 & 27 \\ -1 & -4 & -12 & -32 \end{bmatrix}$	$\begin{bmatrix} -8 & 4 & 0 & 0 \\ 4 & -8 & 4 & 0 \\ 0 & 4 & -8 & 4 \\ 0 & 0 & 2 & -2 \end{bmatrix}$
$\mathbf{F}^{\mathrm{T}}$	$\begin{bmatrix} -\frac{3}{4} & -\frac{1}{4} & 0 & -1 \end{bmatrix}$	$\begin{bmatrix} -1 & 0 & 0 & -1 \end{bmatrix}$	$\begin{bmatrix} -5 & 0 & 0 & -1 \end{bmatrix}$
$\mathbf{a}^{\mathrm{T}}$	$\begin{bmatrix} 2 & -\frac{13}{12} & \frac{1}{2} & -\frac{1}{12} \end{bmatrix}$	$\begin{bmatrix} \frac{7}{3} & -\frac{3}{2} & \frac{5}{6} & -\frac{1}{6} \end{bmatrix}$	$u_i = \begin{bmatrix} \frac{7}{4} & \frac{9}{4} & \frac{11}{4} & \frac{13}{4} \end{bmatrix}$
	LS	G	FE
$\mathbf{K}$	$\begin{bmatrix} 1 & 4 & 12 & 32 \\ 4 & 24 & 72 & 192 \\ 12 & 72 & 240 & 672 \\ 32 & 192 & 672 & \frac{9728}{5} \end{bmatrix}$	$\begin{bmatrix} 2 & 4 & 8 & 16 \\ 4 & \frac{32}{3} & 24 & \frac{256}{5} \\ 8 & 24 & \frac{288}{5} & 128 \\ 16 & \frac{256}{5} & 128 & \frac{2048}{7} \end{bmatrix}$	$\begin{bmatrix} 4 & -2 & 0 & 0 \\ -2 & 4 & -2 & 0 \\ 0 & -2 & 4 & -2 \\ 0 & 0 & -2 & 2 \end{bmatrix}$
$\mathbf{F}^{\mathrm{T}}$	$\begin{bmatrix} 1 & 2 & 10 & 30 \end{bmatrix}$	$\begin{bmatrix} \frac{7}{3} & \frac{25}{6} & \frac{81}{10} & \frac{241}{15} \end{bmatrix}$	$\begin{bmatrix} \frac{1}{2} & \frac{1}{12} & 0 & 1 \end{bmatrix}$
$\mathbf{a}^{\mathrm{T}}$	$\begin{bmatrix} 2 & -\frac{17}{16} & \frac{23}{48} & -\frac{5}{64} \end{bmatrix}$	$\begin{bmatrix} \frac{97}{48} & -\frac{9}{8} & \frac{17}{32} & -\frac{35}{384} \end{bmatrix}$	$\begin{bmatrix} \frac{19}{24} & \frac{4}{3} & \frac{11}{6} & \frac{7}{3} \end{bmatrix}$

- System Matrix $\mathbf{K}$ is nonsymmetric for Subdomain, Collocation and Finite Difference methods.
- System Matrix $\mathbf{K}$ is always symmetric for Least Square method.
- For this self adjoint problem $\mathbf{K}$ is symmetric for Galerkin methods ($\mathbf{w} = [x^i]$ and FE hat functions).
- Finite Element trial functions are local leading to sparse structure of $\mathbf{K}$ matrix.
- Spectral trial functions are continuus and span the entire domain. The matrix $\mathbf{K}$ is dense.
- Spectral methods have better convergence properties than FE methods, while their use is most often is limited to simple geometries.

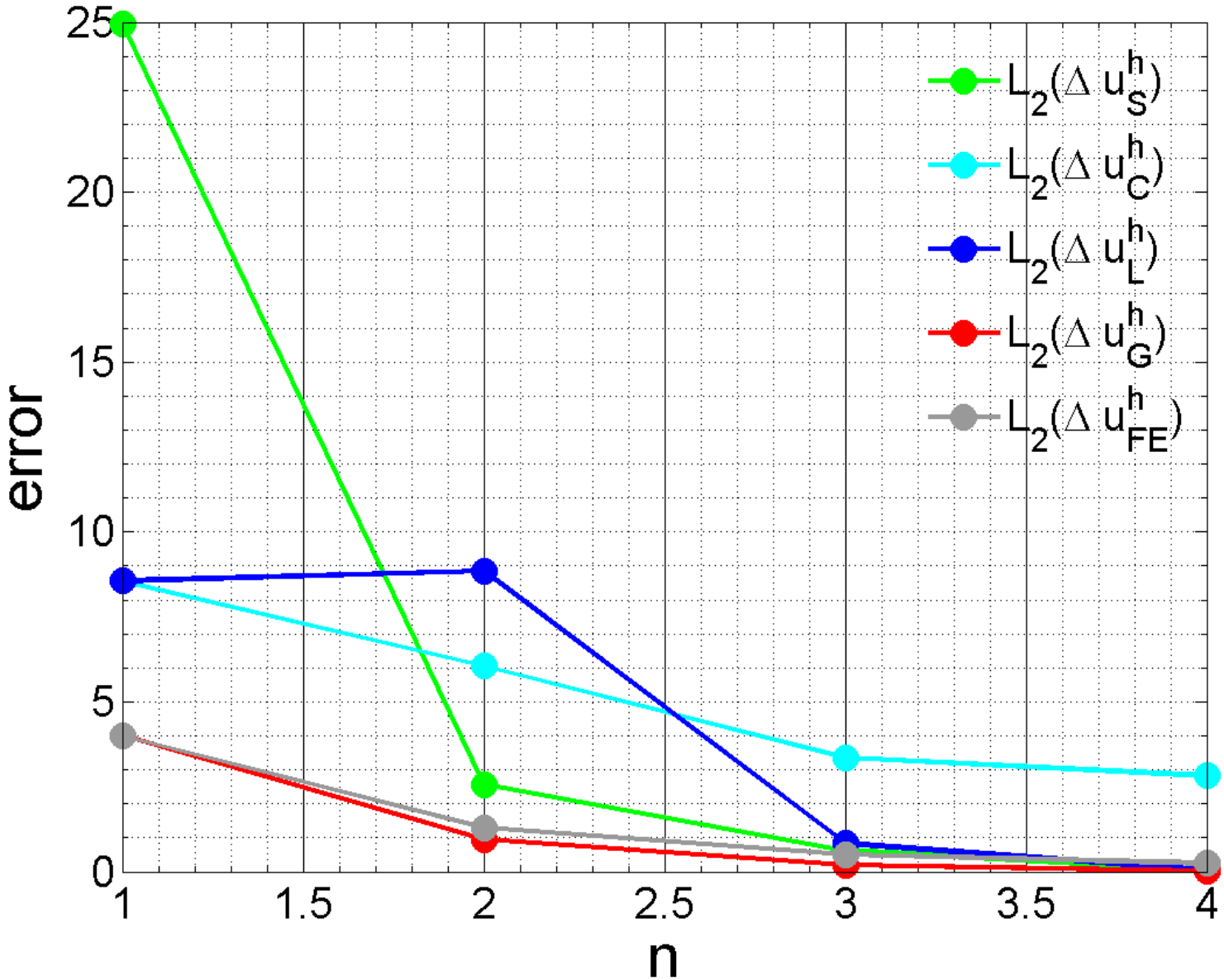
Bar example, $n = 4$, Comparison of solutions



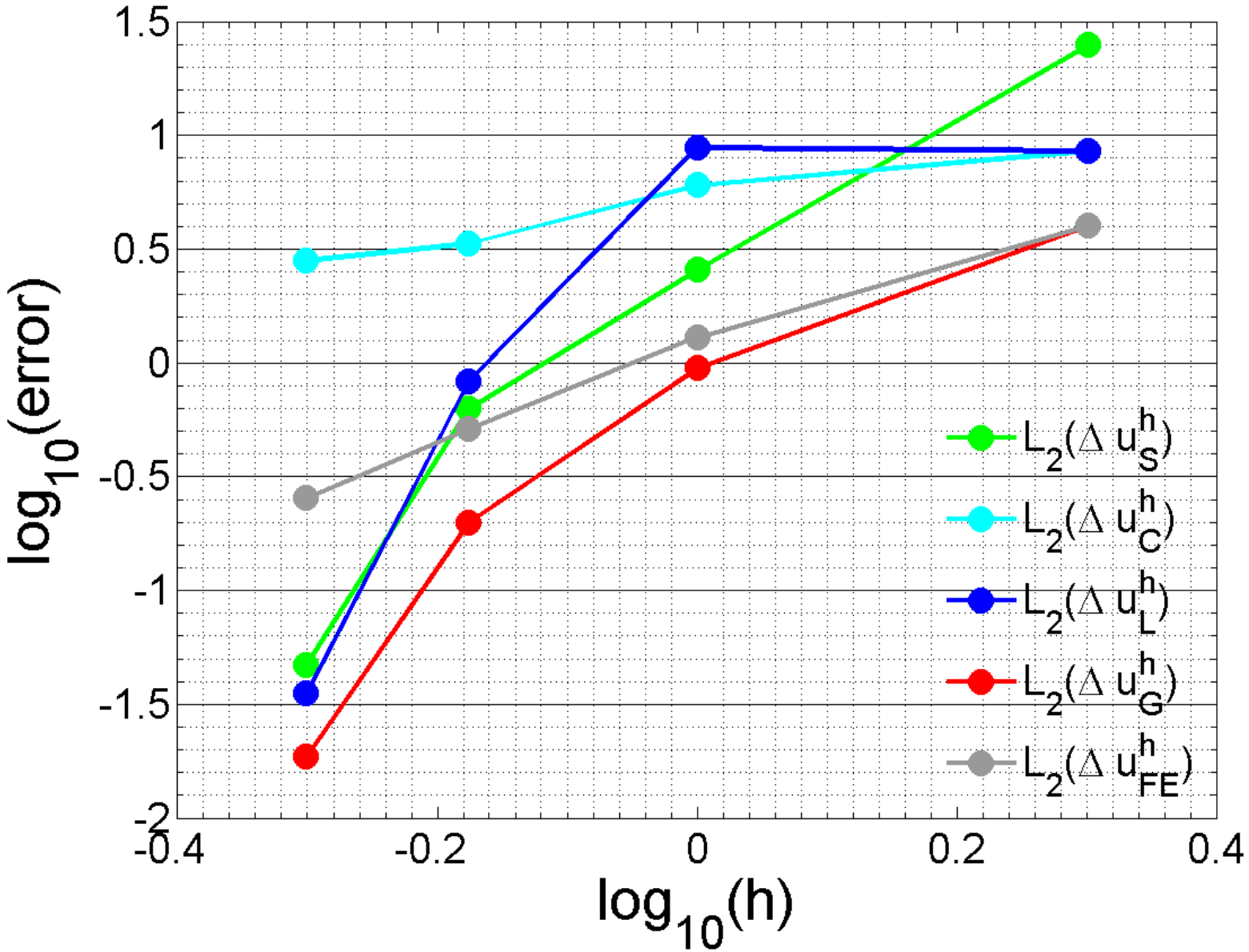
Bar example, $n = 4$, Comparison of solutions



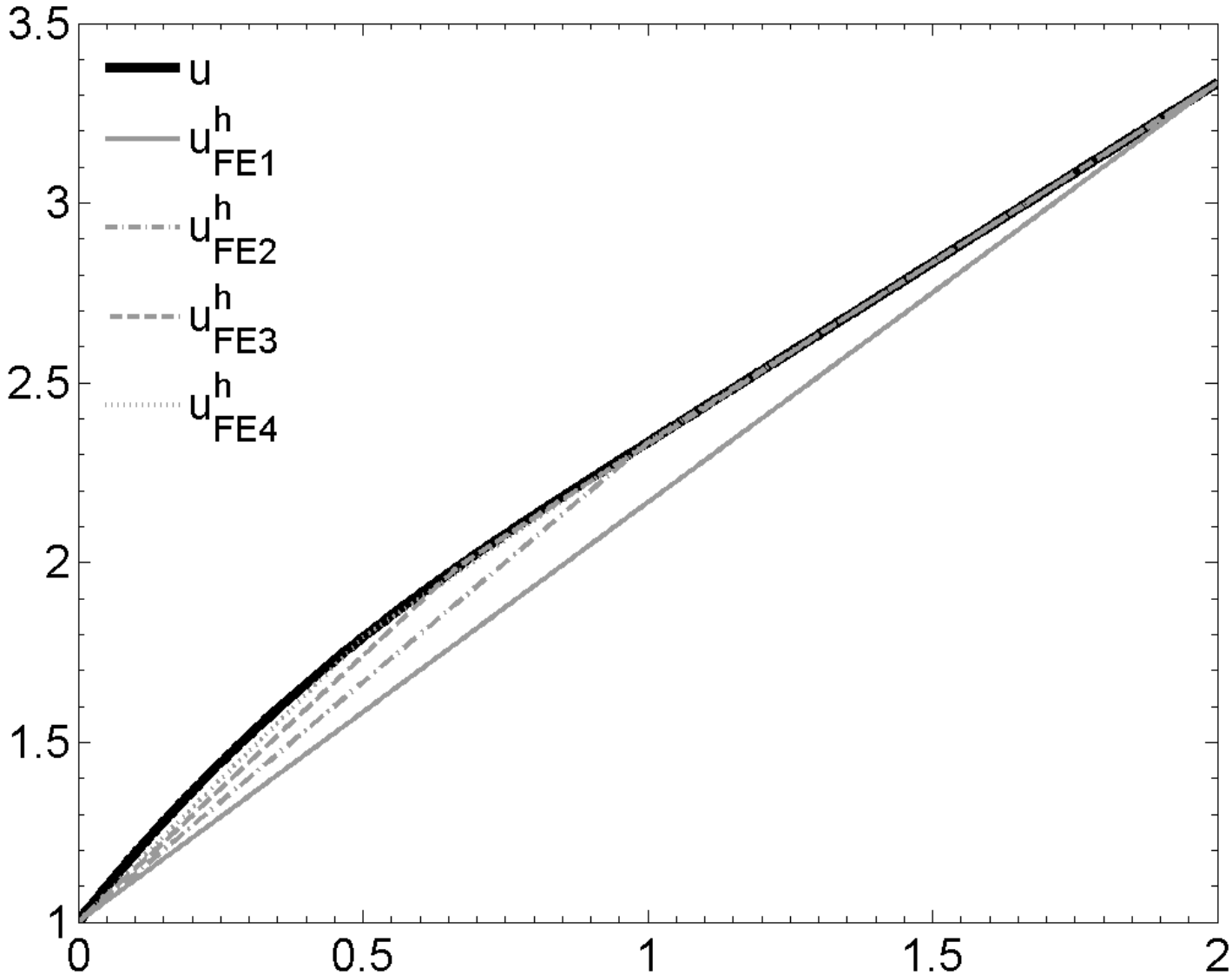
Bar example, Error Convergence



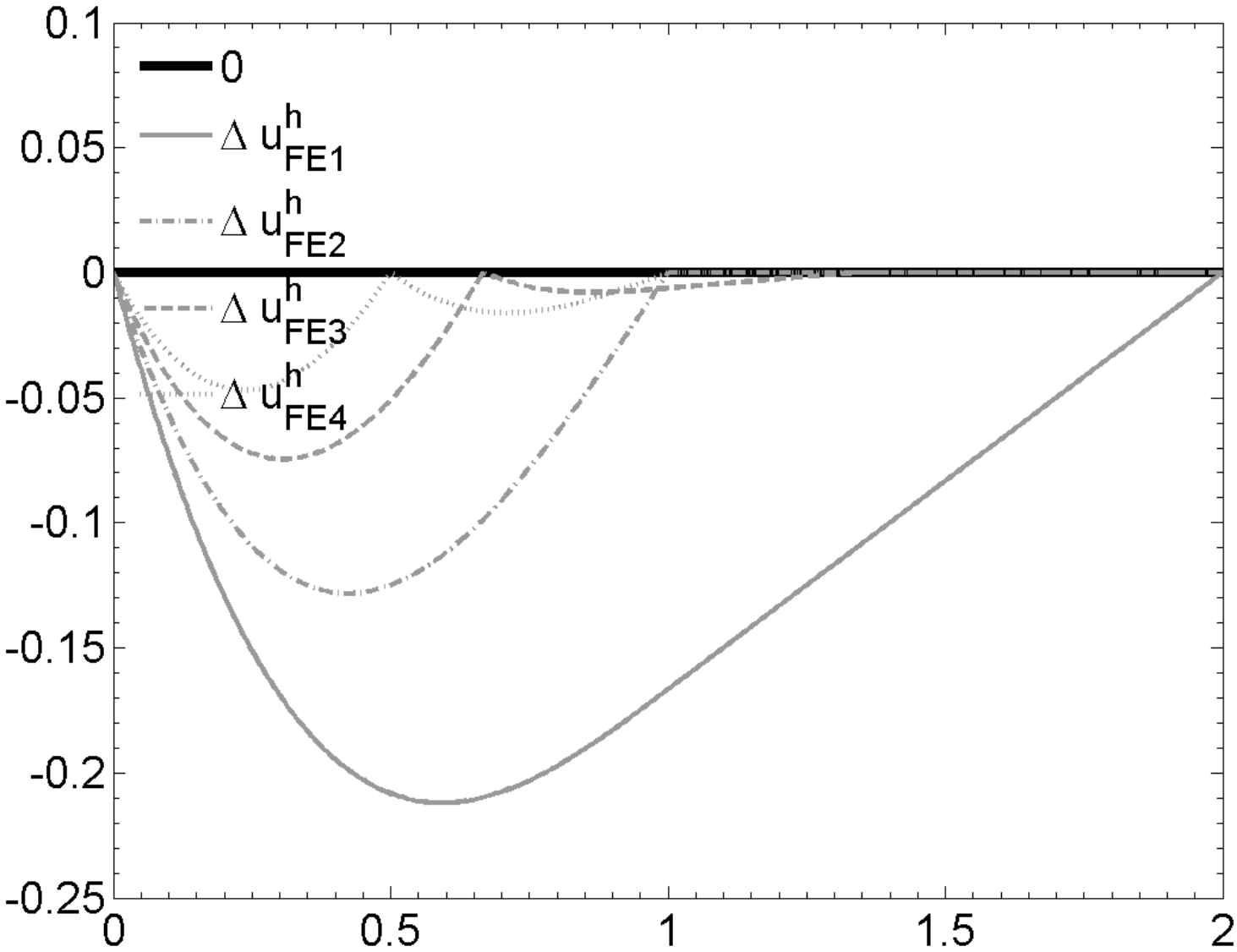
Bar example, Error Convergence



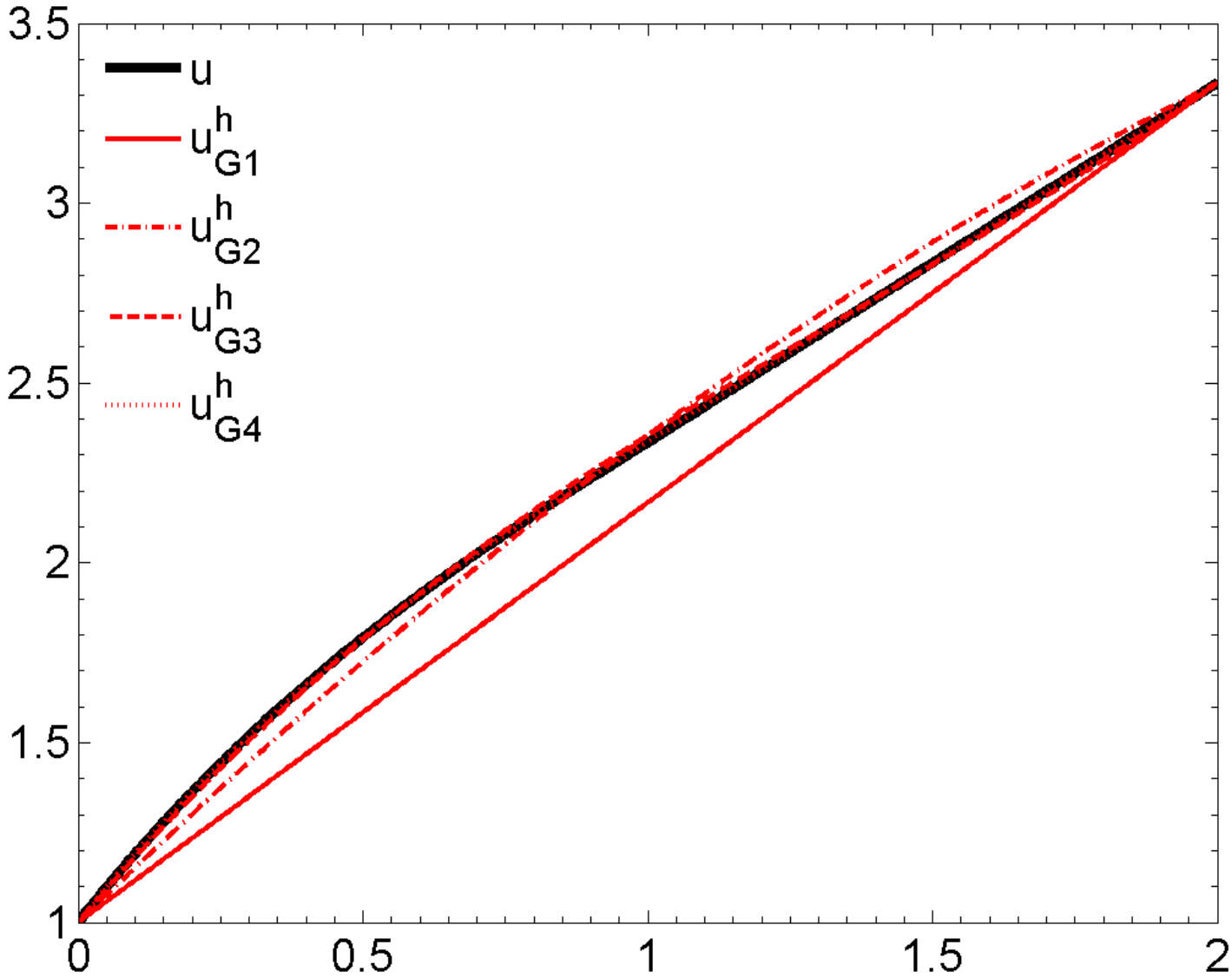
Bar example, Galerkin Finite Element Method



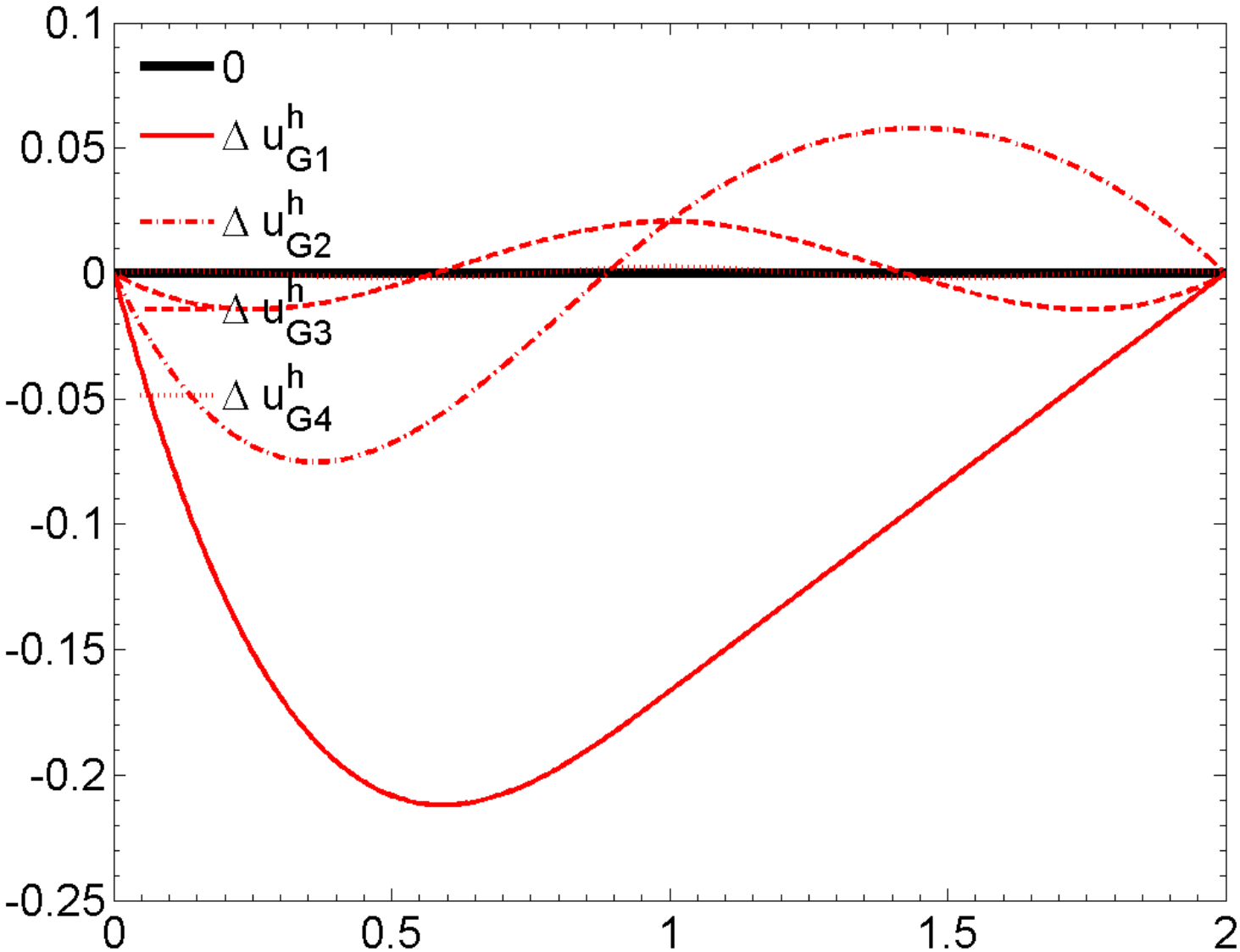
Bar example, Galerkin Finite Element Method



Bar example, Galerkin Spectral Method



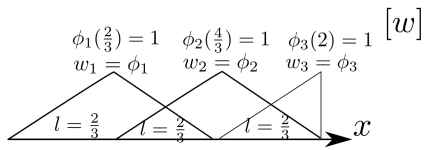
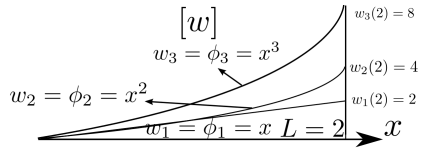
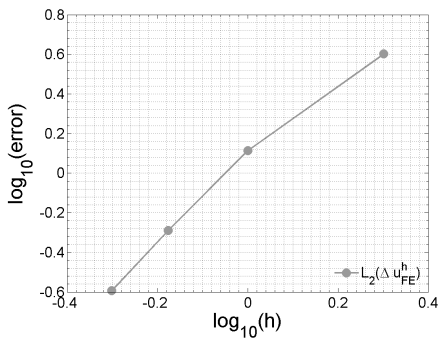
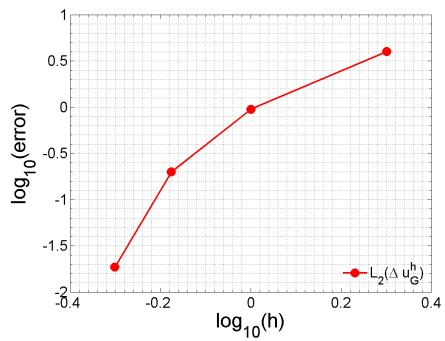
Bar example, Galerkin Spectral Method



Observations: method performance

- While Collocation and Finite Difference methods satisfy the strong form at discrete points (approximate for FD method) this clearly does not imply that the discrete solution matches exact solution at these points (we enforce $\mathcal{R}_i(u^h(x_j)) = 0$ rather than $u^h(x_j) - u(x_j) = 0$).
- For Finite Element method in 1D the approximate solution exactly matches the exact solution at nodal point. This property does not generalize to all finite elements.
- For the space of admissible solutions $\mathcal{V}^h$ we designed the least square method to minimize R^2 (*i.e.*, the residual error). We observe that this does not directly translate to least error between approximate and exact solution. That is, while residual error is minimum for LS method, the error $u^h(x_j) - u(x_j)$ is not minimized. For example, Galerkin method performs much better in this respect (Galerkin methods minimize the solution error in certain norms).

Observations: FE versus spectral methods

Feature	Finite Element	Spectral Methods
Trial Functions Example	Local / Finite Regularity hat functions 	Globally Smooth $\phi = [x\ x^2\ x^3]$ 
Matrix $\mathbf{K}$ Example	Sparse $\begin{bmatrix} 4 & -2 & 0 & 0 \\ -2 & 4 & -2 & 0 \\ 0 & -2 & 4 & -2 \\ 0 & 0 & -2 & 2 \end{bmatrix}$	Full (diagonal for orthogonal ϕ) $\begin{bmatrix} 2 & 4 & 8 & 16 \\ 4 & \frac{32}{3} & 24 & \frac{256}{5} \\ 8 & 24 & \frac{288}{5} & 128 \\ 16 & \frac{256}{5} & 128 & \frac{2048}{7} \end{bmatrix}$
order of accuracy of u^h (p)	fixed (*e.g.*, $p=1$)	$\nearrow$ vs. n (*e.g.*, $p=n$)
Convergence Example	Linear: $e = Ch^\alpha$ $\alpha = 2$ 	higher than linear exponential 
Geometry	Very general geometries	simple (*e.g.*, rectangular) in practice to get diagonal $\mathbf{K}$

Diagonal matrix for spectral methods

- The global nature of trial functions ϕ in spectral method results in full $\mathbf{K}$ matrices that are expensive to solve.
- To circumvent this problem we employ trial functions that make $\mathbf{K}$ diagonal.
- In weak statement $K_{ij} := \mathcal{A}(\phi_i, \phi_j) = \int_{\mathcal{D}} L_m^w(\phi_i) L_m(\phi_j)\, \mathrm{dv}$.
- If the problem is self-adjoint $\mathcal{A}(.,.)$ is an inner product and we can construct an orthogonal trial function basis ϕ_i for example using Gram Schmidt method.
- Given the particular form of $\mathcal{A}$ (from L_m^w and L_m) and domain of integration $\mathcal{D}$ ($[0\ 1]$, $[-1\ 1]$, semi-infinite, infinite, *etc.*) we employ various trigonometric and orthogonal polynomial spaces. Some examples are:
  - $\phi_k(x) = e^{\mathrm{i}kx}$ Fourier spectral method.
  - $\phi_k(x) = T_k(x)$ Chebyshev spectral method.
  - $\phi_k(x) = L_k(x)$ or $P_k(x)$ Legendre spectral method.
  - $\phi_k(x) = \mathcal{L}_k(x)$ Laguerre spectral method.
  - $\phi_k(x) = H_k(x)$ Hermite spectral method.

  where $T_k(x)$, $L_k(x)(P_k(x))$, $\mathcal{L}_k(x)$, and $H_k(x)$ are the Chebyshev, Legendre, Laguerre, and Hermite polynomials of degree k, respectively.
- The orthogonal property of these functions is for simple geometries. That is why spectral methods are more popular for simple geometries where we can take advantage of their exponential convergence property while keeping computational costs low by using orthogonal trial functions.

Summary: matrix structure

- The solution of the matrix equation is often the most expensive part of the computation.
- System Matrix $\mathbf{K}$ is nonsymmetric for Subdomain, Collocation and Finite Difference methods.
- System Matrix $\mathbf{K}$ is always symmetric for Least Square method.
- For this self adjoint problem $\mathbf{K}$ is symmetric for Galerkin methods ($\mathbf{w} = [x^i]$ and FE hat functions).
- Finite Element trial functions are local leading to sparse structure of $\mathbf{K}$ matrix.
- Spectral trial functions are continuus and span the entire domain. The matrix $\mathbf{K}$ is dense.
- Use of orthogonal trial functions in spectral methods leads to computationally attractive diagonal matrices. This approach is generally limited to simple geometries.

Appendix: Function spaces (optional)

Function spaces Weighted residual method

- For the weighted residual method we required the weight functions w be C^0 (*cf.* (180)):

$$\int_0^2 w \left(\frac{\mathrm{d}^2 u^h}{\mathrm{d}x^2} + q(x) \right) \, \mathrm{d}x + w^f (1 - \frac{\mathrm{d}u}{\mathrm{d}x}(2)) = 0$$

- At the same time in collocation method, we used $w_i = \delta(\mathbf{x} - \mathbf{x}_i)$ which is a distribution and is not a C^0 function.
- Still as long as the integrals can be computed these weight "functions" can be used.

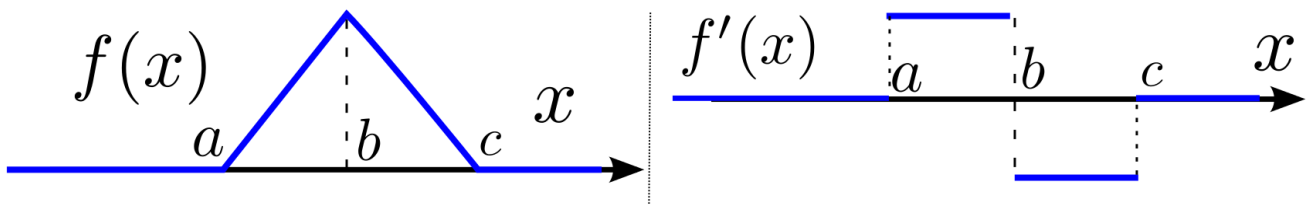
C^k function spaces

- We define the function spaces

$$C^k(\mathcal{D}) = \{f|\ f \text{ and } \frac{\partial^i f}{\partial \mathbf{x}^i} \text{ exist and are continuus } \forall 0 < i \leq k\ \wedge \mathbf{x} \in \mathcal{D}\} \tag{274}$$

$$C^0(\mathcal{D}) = \text{ continuus functions on } \mathcal{D}$$
$$C^1(\mathcal{D}) = \text{ functions with continuus derivative on } \mathcal{D}$$
$$C^\infty(\mathcal{D}) = \text{ infinitely differentiable function on } \mathcal{D}$$

- These conditions are for all points in $\mathcal{D}$. For example the hat function below function f is continuus but $f'(x)$ does not exist at a, b, and c. So, f is only in $C^0(\mathbb{R})$.



We define the corresponding bounded C spaces as,

$$C_b^k(\mathcal{D}) = \{f|f \in C^k(\mathcal{D}), \forall \mathbf{x} :\ |f(\mathbf{x})| < \infty\} \tag{275}$$

- Clearly,

$$C^{k+1}(\mathcal{D}) \subset C^k(\mathcal{D})$$
$$C_b^k(\mathcal{D}) \subset C^k(\mathcal{D})$$

- For closed sets $\mathcal{D}$ ($\mathcal{D} = \bar{\mathcal{D}}$) $C_b^k(\mathcal{D}) = C^k(\mathcal{D})$. For example $\mathcal{D}_1 = (0,1) = \{x|\ 0 < x < 1\}$ is open while $\mathcal{D}_2 = [0,1] = \{x|\ 0 \leq x \leq 1\}$ is closed. Function $f(x) = \frac{1}{x}$ is in $C^\infty(\mathcal{D}_1)$ but not $C_b^\infty(\mathcal{D}_1)$.

Sobolev Function spaces

- Definition: we define the function space $L^2(\mathcal{D})$ as,

$$L^2(\mathcal{D}) := \{f| \int_{\mathcal{D}} f^2(\mathbf{x}) < 0\} \tag{276}$$

  That is $L^2(\mathcal{D})$ is the space of square integrable functions on $\mathcal{D}$.
- More generally we define Sobolev space of functions of degree k as,

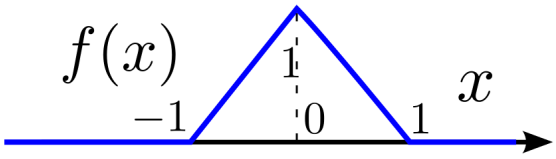
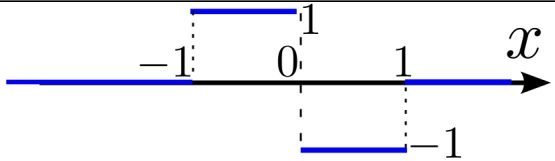
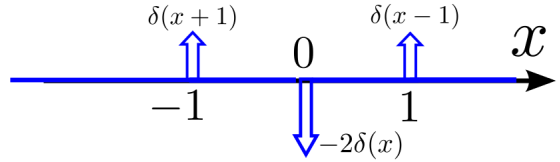
$$H^k(\mathcal{D}) = \{f| \ f \in L^2(\mathcal{D}), \frac{\partial f}{\partial \mathbf{x}} \in L^2(\mathcal{D}), \ldots \frac{\partial^k f}{\partial \mathbf{x}^k} \in L^2(\mathcal{D})\} \tag{277}$$

  where $\frac{\partial f}{\partial \mathbf{x}}$ and $\frac{\partial^k f}{\partial \mathbf{x}^k}$ refer to (arbitrary partial) derivatives of 1st and k^{th} order of f.
- That is $H^k(\mathcal{D})$ is the space of "functions" that possess square-integrable **generalized** derivatives through order k. We have,

$$H^0(\mathcal{D}) = L^2(\mathcal{D}) \tag{278}$$

$$H^{k+1}(\mathcal{D}) \subset H^k(\mathcal{D}) \tag{279}$$

Comparison of C^k and Sobolev spaces

$f(x)$	$f'(x)$	$f''(x)$
		
$C^0(\mathbb{R})$	$C^1(\mathbb{R})$	$C^2(\mathbb{R})$
Yes	No no derivatives at $\{-1, 0, 1\}$	No not a C^0
$H^0(\mathbb{R}) = L^2(\mathbb{R})$	$H^1(\mathbb{R})$	$H^2(\mathbb{R})$
Yes $\int_{-\infty}^{\infty}(f(x))^2\,\mathrm{d}x = \frac{2}{3} < \infty$	Yes $\int_{-\infty}^{\infty}(f(x))^2\,\mathrm{d}x = \frac{2}{3} < \infty$ $\int_{-\infty}^{\infty}(f'(x))^2\,\mathrm{d}x = 2 < \infty$	No $\int_{-\infty}^{\infty}(f(x))^2\,\mathrm{d}x = \frac{2}{3} < \infty$ $\int_{-\infty}^{\infty}(f'(x))^2\,\mathrm{d}x = 2 < \infty$
		$\int_{-\infty}^{\infty}(f''(x))^2\,\mathrm{d}x =$ $\int_{-\infty}^{\infty}(\delta(x+1))^2\,\mathrm{d}x +$ $\int_{-\infty}^{\infty}(2\delta(x))^2\,\mathrm{d}x +$ $\int_{-\infty}^{\infty}(\delta(x-1))^2\,\mathrm{d}x$ Not Defined

- So, the hat function $f(x)$ is $C^0(\mathbb{R})$ (continuus) but not $C^1(\mathbb{R})$.
- $f(x)$ is $H^1(\mathbb{R})$ but not $H^2(\mathbb{R})$.
- Sobolev's theorem:

$$\boxed{H^{k+1}(\mathcal{D}) \subset C_b^k(\mathcal{D})} \tag{280}$$

- As an example $f \in H^1(\mathbb{R}) \quad \Rightarrow f \in C_b^0(\mathbb{R})$

Galerkin Weak Statement Function spaces

- The Finite Element hat functions are piece-wise linear (*i.e.*, C^0 functions.
- To alleviate high derivative order requirement of 2 for u^h in weighted residual statement we used the weak statement for finite element trial functions (*cf.* (187)):

$$\int_0^2 \frac{\mathrm{d}w}{\mathrm{d}x}\frac{\mathrm{d}u^h}{\mathrm{d}x}\,\mathrm{d}x = \int_0^2 w(x).q(x)\,\mathrm{d}x + w(2)$$

  where we still required u^h and w to be C^1 (Hence ϕ_i should have been C^1).
- That is the C^0 trial hat functions still do not satisfy C^1 condition.
- For Galerkin method ($\mathbf{w} = \boldsymbol{\phi}$) we need to evaluate these integrals:

$$\mathcal{A}(\phi_i, \phi_j) = \int_0^2 \frac{\mathrm{d}\phi_i}{\mathrm{d}x}\frac{\mathrm{d}\phi_j}{\mathrm{d}x}\,\mathrm{d}x$$

- From Cauchy Schwarz inequality $\mathcal{A}(a,b)^2 \leq \mathcal{A}(a,a).\mathcal{A}(b,b)$ for inner product $\mathcal{A}(.,.)$:

$$\left(\int_0^2 \frac{\mathrm{d}\phi_i}{\mathrm{d}x}\frac{\mathrm{d}\phi_j}{\mathrm{d}x}\,\mathrm{d}x\right)^2 \leq \left(\int_0^2 \frac{\mathrm{d}\phi_i}{\mathrm{d}x}\frac{\mathrm{d}\phi_i}{\mathrm{d}x}\,\mathrm{d}x\right).\left(\int_0^2 \frac{\mathrm{d}\phi_j}{\mathrm{d}x}\frac{\mathrm{d}\phi_j}{\mathrm{d}x}\,\mathrm{d}x\right) \quad \text{we observe that If:}$$

$$\int_0^2 \frac{\mathrm{d}\phi_i}{\mathrm{d}x}\frac{\mathrm{d}\phi_i}{\mathrm{d}x}\,\mathrm{d}x < \infty \quad \text{that is exists and is finite} \tag{281}$$

  for all ϕ_i all $\int_0^2 \frac{\mathrm{d}w}{\mathrm{d}x}\frac{\mathrm{d}u^h}{\mathrm{d}x}\,\mathrm{d}x$ can be evaluated.
- Similarly (using Cauchy Schwarz inequality), $\int_0^1 w(x).q(x)\,\mathrm{d}x$ can be evaluated if,

$$\forall i \in \{1,\ldots,n\}: \int_0^2 \phi_i\phi_i\,\mathrm{d}x < \infty \qquad \text{and} \qquad \int_0^2 q.q\,\mathrm{d}x < \infty \tag{282}$$

Galerkin Weak Statement Function spaces

- From (281) (and (282)) and the definition of Sobolev spaces we conclude that,

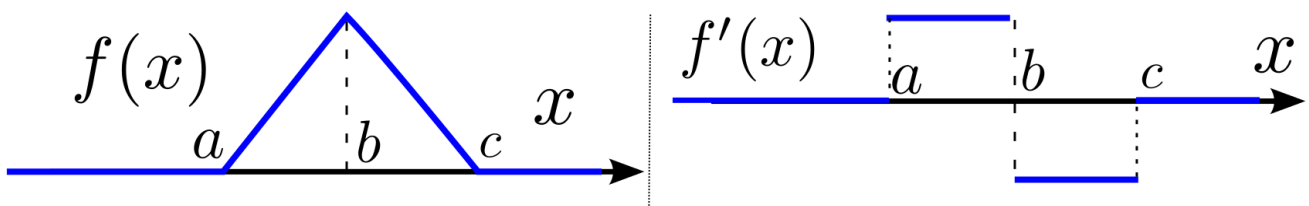
$$\phi_i \in H^1(\mathcal{D}) \quad \Rightarrow \quad \text{weak form can be evaluated} \tag{283}$$

- So, the space of admissible functions and the weak statement can be revised as ($\mathcal{D} = [0\ 2]$):

Find $u \in \mathcal{V} = \{v |\ v \in H^1(\mathcal{D}), v(0) = \bar{u} = 1\}$:$\forall w \in \mathcal{V}_0 = \{v |\ v \in H^1(\mathcal{D}), v(0) = \bar{u} = 0\}$

$$\int_0^2 \frac{\mathrm{d}w}{\mathrm{d}x}\frac{\mathrm{d}u}{\mathrm{d}x}\,\mathrm{d}x = \int_0^1 w(x).q(x)\,\mathrm{d}x + w(2) \tag{284}$$

- The hat shape functions we used for 1D bar (shown below) belong to $H^1(\mathcal{D})$ while not being in restricted $C^1(\mathcal{D})$ function spaces before the revision.



Galerkin Weak Statement Function spaces

1. We first reduce the highest derivative order $M = 2m$ in the strong form (and weighted residual statement) to m in the weak statement.
2. Next, we observe that the functions should only be in $H^m(\mathcal{D})$. We observed that $H^m(\mathcal{D}) \subset C^{m-1}(\mathcal{D})$. In practice, the finite element trial functions that are in $C^{m-1}(\mathcal{D})$ are also $H^m(\mathcal{D})$.

Conventional (continuus) finite element methods:

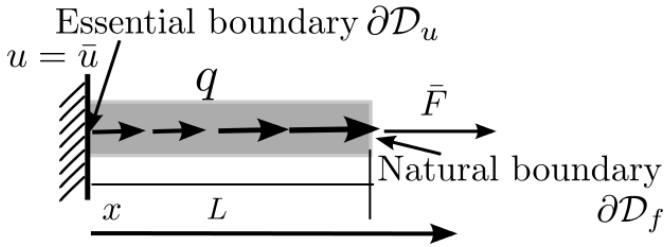
Strong Form order $M = 2m$ $\Rightarrow$

Trial functions are C^{m-1}

Finite Element minimum polynomial order: Solid Bar

- The strong form for the solid bar is,

$$\frac{\mathrm{d}}{\mathrm{d}x}\left(EA\frac{\mathrm{d}u}{\mathrm{d}x}\right)+q(x)=0$$



- The weighted residual statement incorporating the natural boundary conditions for the bar in the figure is (function restrictions on $\partial\mathcal{D}_u$ are not given for brevity),

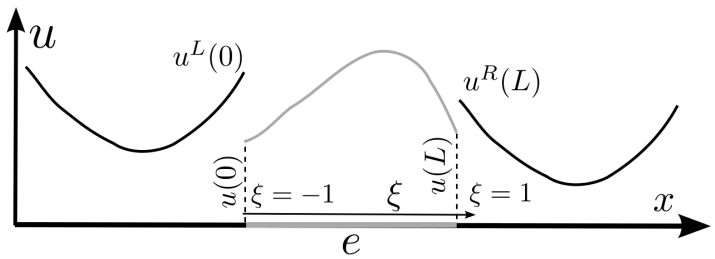
$$\int_0^L w(x)\left(\frac{\mathrm{d}}{\mathrm{d}x}\left(EA\frac{\mathrm{d}u}{\mathrm{d}x}\right)+q(x)\right)\,\mathrm{d}x+\left\{w(x).[\bar{F}-F(u(x))]\right\}|_{x=L}=0$$

- The highest derivative order for u in strong form and WRS is $M=2m=2$.
- The weak statement is obtained from the WRS by integration by parts:

$$\int_0^L \frac{\mathrm{d}w}{\mathrm{d}x}EA\frac{\mathrm{d}u}{\mathrm{d}x}\,\mathrm{d}x=\int_0^L w(x).q(x)\,\mathrm{d}x+\left(w(x).\bar{F}\right)|_{x=L}$$

- Weak statement reduced highest derivative order to $m=1$.
- Finite Element trial functions should be C^0 $(m-1=M/2-1=0)$.

Finite Element minimum polynomial order: Solid Bar



- Finite element trial functions for solid bar should be C^0 (continuous).
- Shape function (part of the trial function in an element) in element e is given by,

$$u(\xi) = \alpha_0 + \alpha_1 \xi + \cdots + \alpha_p \xi^p \tag{285}$$

  $\xi =$ local coordinate system in e $\quad$ $p =$ polynomial order in e.
- Since the trial functions should be C^0 the shape functions at points $\xi = 0$ and $\xi = L$ should match the values of neighboring shape functions:

$$u(0) = u^L(0)$$
$$u(L) = u^R(L)$$

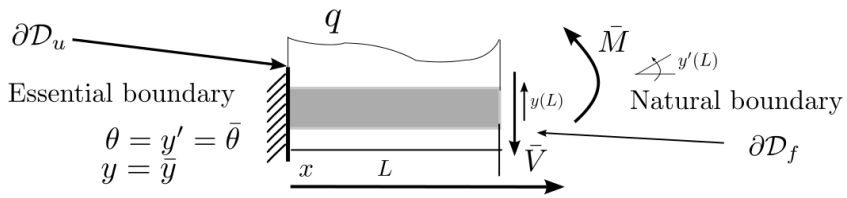
- That is, we have two equations which corresponds to 2 terms in (285). $\Rightarrow$
- Shape function for the bar element is linear.
- We will later observe that the shape functions for the bar element are:

$$N_1(\xi) = \frac{1}{2}(1 - \xi)$$
$$N_2(\xi) = \frac{1}{2}(1 + \xi)$$

Finite Element minimum polynomial order: Solid Beam

- The strong form for the solid beam is,

$$\frac{\mathrm{d}^2}{\mathrm{d}x^2}\left(EI\frac{\mathrm{d}^2 y}{\mathrm{d}x^2}\right)+q(x)=0$$



  - The weighted residual statement incorporating the natural boundary conditions for the beam in the figure is (function restrictions on $\partial\mathcal{D}_u$ are not given for brevity),

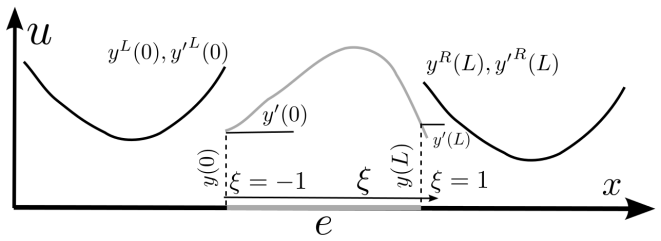
$$\int_0^L w(x)\left(\frac{\mathrm{d}^2}{\mathrm{d}x^2}\left(EI\frac{\mathrm{d}^2 y}{\mathrm{d}x^2}\right)-q(x)\right)\mathrm{d}x+\left\{w.[\bar{V}-V(y)]-\frac{\mathrm{d}w}{\mathrm{d}x}.[\bar{M}-M(y)]\right\}|_L=0$$

  - The highest derivative order for u in strong form and WRS is $M=2m=4$.
  - The weak statement is obtained from the WRS by integration by parts:

$$\int_0^L \frac{\mathrm{d}^2 w}{\mathrm{d}x^2}EA\frac{\mathrm{d}^2 y}{\mathrm{d}x^2}\,\mathrm{d}x=\int_0^L w(x).q(x)\,\mathrm{d}x+\left(-w(x).\bar{V}+\frac{\mathrm{d}w}{\mathrm{d}x}(x).\bar{M}\right)|_{x=L}$$

  - Weak statement reduced highest derivative order to $m=2$.
  - Finite Element trial functions should be C^1 $(m-1=M/2-1=1)$.

Finite Element minimum polynomial order: Solid Beam



- Finite element trial functions for solid bar should be C^1 (continuous derivative).
- Shape function in element e is given by,

$$y(\xi) = \alpha_0 + \alpha_1 \xi + \cdots + \alpha_p \xi^p \tag{286}$$

  $\xi =$ local coordinate system in e $\quad$ $p =$ polynomial order in e.

- Since the trial functions should be C^1 the shape functions at points $\xi = 0$ and $\xi = L$ should match the values of neighboring shape functions:

$$y(0) = y^L(0) \qquad y'(0) = y'^L(0)$$
$$y(L) = y^R(L) \qquad y'(L) = y'^R(L)$$

- That is, we have four equations which corresponds to 4 terms in (286). $\Rightarrow$
- Shape function for the bar element is cubic.
- We will later observe that the shape functions for the beam element are:

$$N_1(\xi) = \frac{1}{4}(2 - 3\xi + \xi^3) \qquad N_2(\xi) = \frac{L}{8}(1 - \xi - \xi^2 + \xi^3)$$
$$N_3(\xi) = \frac{1}{4}(2 + 3\xi - \xi^3) \qquad N_4(\xi) = \frac{L}{8}(-1 - \xi + \xi^2 + \xi^3)$$

1D elements

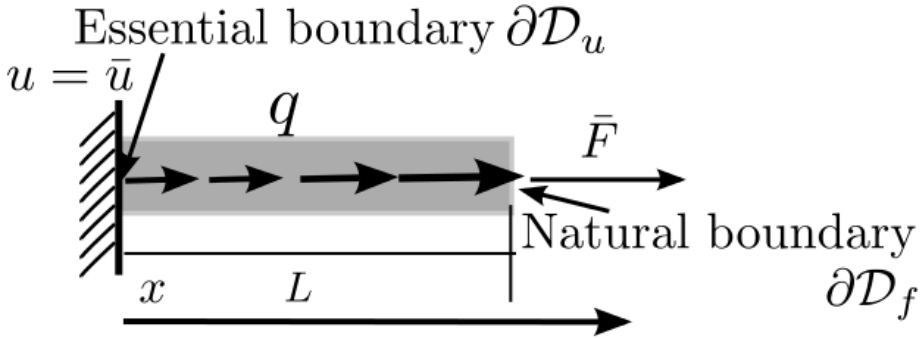
Element types:

1. 1D solid bar element.
2. Truss element.

Concepts:

1. Global (weighted residual) vs local (element level) perspectives.
2. Stiffness matrix.
3. Forces: 1.Source term; 2.Natural BC; 3.Essential BC, 4.Nodal.
4. Nodes, elements, shape function, dof.
5. Nodes with more than one dof (truss).
6. Element local coordinate system ξ (bar).
7. Rotation of element local coordinate system (truss).
8. Full stiffness $\mathbf{K}$ (free + prescribed dofs) vs (free only dofs) $\mathbf{K}_{ff}$.
9. High order differential equations (*e.g.*, C^1 beam elements).
10. Multiphysics coupling (beams: axial, bending, & torsional coupling).

Review: Weak Statement for solid bar



- From previous discussion on the function spaces the continuum revised weak statement for the bar in figure is,

$$\text{Find } u \in \mathcal{V} = \{v |\, v \in H^1(\Omega),\ v(0) = \bar{u}\} \text{ such that}$$
$$\forall w \in \mathcal{V}_0 = \{v |\, v \in H^1(\Omega),\ v(0) = 0\} :$$
$$\int_{\mathcal{D}} \frac{\mathrm{d}w}{\mathrm{d}x} EA \frac{\mathrm{d}u}{\mathrm{d}x}\, \mathrm{d}x = \int_{\mathcal{D}} w.q\, \mathrm{d}x + \left(w.\bar{F}\right)_{x=L} \tag{287}$$

Review: Linear and bilinear forms

- We define the following forms:

$$\mathcal{A}(w,u) := \int_{\mathcal{D}} \frac{\mathrm{d}w}{\mathrm{d}x} EA \frac{\mathrm{d}u}{\mathrm{d}x}\,\mathrm{dv} \qquad \text{bilinear form} \tag{288a}$$

$$(w,q)_r := \int_{\mathcal{D}} w.q\,\mathrm{dv} \qquad \text{linear force from source terms} \tag{288b}$$

$$\left(w,\bar{F}\right)_N := \int_{\partial\mathcal{D}_f} w\bar{\mathbf{F}}.\mathbf{N}\,\mathrm{ds} = \left(w.\bar{F}\right)_{x=L} \qquad \text{linear force from natural BC} \tag{288c}$$

  note that $(\partial\mathcal{D}_f = \{L\})$.
- All these forms are linear in w and u (also linear in q and $\bar{F}$ in (288b) and (288c), respectively:

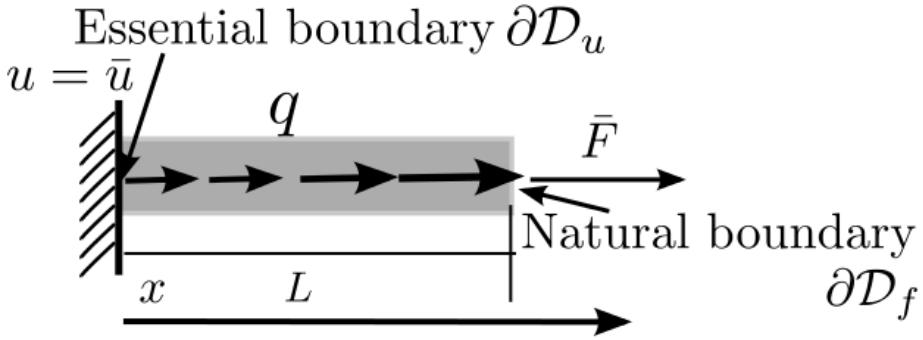
$$\begin{aligned} \mathcal{A}(w_1+w_2,u) &:= \mathcal{A}(w_1,u) + \mathcal{A}(w_2,u) \qquad \mathcal{A}(w,u_1+u_2) := \mathcal{A}(w,u_1) + \mathcal{A}(w,u_2) \\ (w_1+w_2,q)_r &:= (w_1,q)_r + (w_2,q)_r \\ \left(w_1+w_2,\bar{F}\right)_N &:= \left(w_1,\bar{F}\right)_N + \left(w_2,\bar{F}\right)_N \end{aligned} \tag{289}$$

- Since the problem is self-adjoint the bilinear is commutative as well,

$$\mathcal{A}(w,u) := \mathcal{A}(u,w) \tag{290}$$

- This leads to the symmetry of system matrix $\mathbf{K}$.

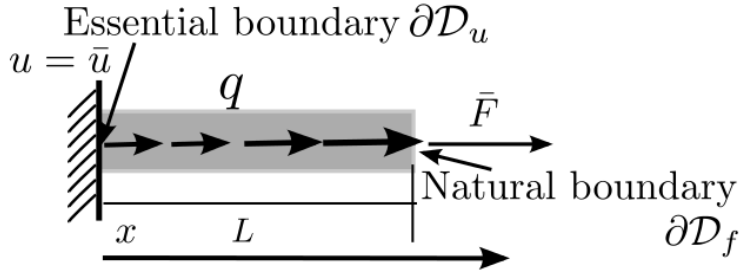
Review: Weak Statement



- Using the definitions for linear and bilinear forms for bar example, we can write the weak statement as,

$$\begin{aligned}
&\text{Find } u \in \mathcal{V} = \{v |\, v \in H^1(\Omega),\ v(0) = \bar{u}\} \text{ such that} \\
&\forall w \in \mathcal{V}_0 = \{v |\, v \in H^1(\Omega),\ v(0) = 0\} : \\
&\mathcal{A}(w, u) = (w, q)_r + \left(w, \bar{F}\right)_N
\end{aligned} \tag{291}$$

Discrete Galerkin formulation for solid bar



- The corresponding Galerkin ($w = \phi$) discrete weak statement is,

$$\text{Find } u^h \in \mathcal{V}^h := \phi_p + \mathcal{V}_0^h \text{ such that } \forall I \in \{1, 2, \ldots, n_{\mathrm{f}}\}$$

$$\int_{\mathcal{D}} \frac{\mathrm{d}\phi_I}{\mathrm{d}x} EA \frac{\mathrm{d}u^h}{\mathrm{d}x}\, \mathrm{d}x = \int_{\mathcal{D}} \phi_I . q\, \mathrm{d}x + \left(\phi_I . \bar{F}\right)_{x=L} \tag{292}$$

where

$$\begin{aligned} n_{\mathrm{f}} &:= \text{Number of equations (unknowns) for the problem which is} \\ &:= \text{Number of free degrees of freedom (dof) for the problem} \end{aligned} \tag{293}$$

- Trial and weight functions are chosen such that,

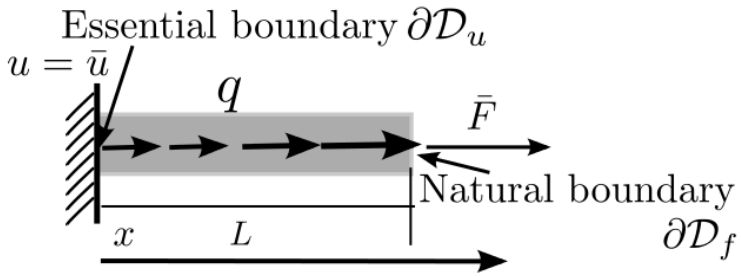
$$w_I = \phi_I \in \mathcal{V}_0^h \qquad \text{Satisfy homogeneous essential BC: } \phi_I(0) = 0 \tag{294a}$$

$$\phi_p \in \mathcal{V}^h \qquad \text{Satisfies essential BC: } \phi_p(0) = \bar{u} \tag{294b}$$

- Discrete $\mathcal{V}_0^h$ is defined by,

$$\mathcal{V}_0^h = \mathrm{span}(\phi_1, \phi_2, \ldots, \phi_{n_{\mathrm{f}}})$$

Discrete Galerkin formulation for solid bar



- Similar to (291) the concise form of the Galerkin statement is,

$$\begin{aligned} &\text{Find } u^h \in \mathcal{V}^h := \phi_p + \mathcal{V}_0^h \text{ such that } \forall I \in \{1, 2, \ldots, n_{\mathrm{f}}\} \\ &\mathcal{A}\left(\phi_I, u^h\right) = \left(\phi_I, q\right)_r + \left(\phi_I, \bar{F}\right)_N \end{aligned} \tag{295}$$

- using the expansion, $\mathbf{u}^h = \sum_{j=1}^{n_{\mathrm{f}}} a_J \boldsymbol{\phi}_J + \boldsymbol{\phi}_p$ (117a), we obtain,

$$\begin{aligned} &\text{Find } \mathbf{a} = \{a_1, a_2, \ldots, a_{n_{\mathrm{f}}}\} \text{ such that } \forall I \in \{1, 2, \ldots, n_{\mathrm{f}}\} \\ &\mathcal{A}\left(\phi_I, a_J \phi_J + \phi_p\right) = \left(\phi_I, q\right)_r + \left(\phi_I, \bar{F}\right)_N \end{aligned} \tag{296}$$

Discrete Galerkin formulation for solid bar

- Using the bilinearity of $\mathcal{A}$ we obtain,

$$\begin{aligned}&\text{Find } \mathbf{a} = \{a_1, a_2, \ldots, a_{n_\mathrm{f}}\} \text{ such that } \forall I \in \{1, 2, \ldots, n_\mathrm{f}\}\\ &\mathcal{A}\left(\phi_I, \phi_J\right) a_J = \left(\phi_I, q\right)_r + \left(\phi_I, \bar{F}\right)_N - \mathcal{A}\left(\phi_I, \phi_p\right)\end{aligned} \tag{297}$$

- It is customary to denote the RHS components as,

$$F_{r\,I} := \left(\phi_I, q\right)_r \qquad \text{Force from Source terms} \tag{298a}$$

$$F_{N\,I} := \left(\phi_I, \bar{F}\right)_N \qquad \text{Force from Natural BCs} \tag{298b}$$

$$F_{D\,I} := \mathcal{A}\left(\phi_I, \phi_p\right) \qquad \text{Force from Essential BCs} \tag{298c}$$

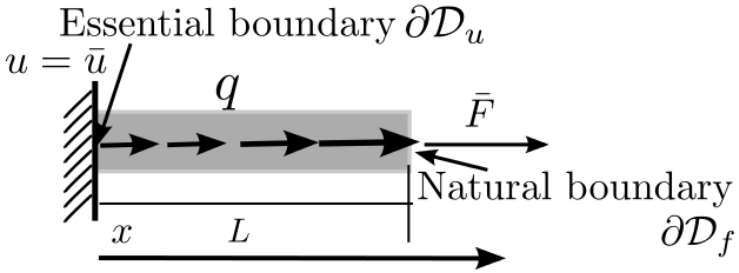
- That is, $\mathbf{a}$ is obtained from the system,

$$K_{I\,J}\, a_J = F_I \text{ for} \tag{299a}$$

$$K_{I\,J} = \mathcal{A}\left(\phi_I, \phi_J\right) \tag{299b}$$

$$\begin{aligned}F_I &= F_{r\,I} + F_{N\,I} - F_{D\,I}\\ &= \left(\phi_I, q\right)_r + \left(\phi_I, \bar{F}\right)_N - \mathcal{A}\left(\phi_I, \phi_p\right)\end{aligned} \tag{299c}$$

Discrete Galerkin formulation for solid bar



- Using the definitions and vector interpretation of the linear and bilinear forms the stiffness and force vectors are,

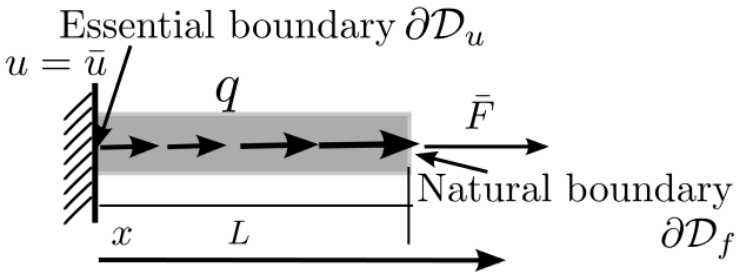
$$\mathbf{K} = \mathcal{A}\left(\boldsymbol{\phi}^{\mathrm{T}}, \boldsymbol{\phi}\right) = \int_{\mathcal{D}} \frac{\mathrm{d}}{\mathrm{d}x}\left[\boldsymbol{\phi}\right]^{\mathrm{T}} EA \frac{\mathrm{d}}{\mathrm{d}x}\left[\boldsymbol{\phi}\right] \,\mathrm{dv}$$

$$= \int_0^L \frac{\mathrm{d}}{\mathrm{d}x} \begin{bmatrix} \phi_1 \\ \phi_2 \\ \vdots \\ \phi_{n_\mathrm{f}} \end{bmatrix} EA \frac{\mathrm{d}}{\mathrm{d}x} \begin{bmatrix} \phi_1 & \phi_2 & \cdots & \phi_{n_\mathrm{f}} \end{bmatrix} \mathrm{d}x \tag{300a}$$

$$\mathbf{F} = \mathbf{F}_r + \mathbf{F}_N - \mathbf{F}_D \tag{300b}$$

- Force $\mathbf{F}$ is the sum of forces induced from source term, $\mathbf{F}_r$, natural boundary conditions, $\mathbf{F}_N$, and essential boundary conditions, $\mathbf{F}_D$.

Discrete Galerkin formulation for solid bar



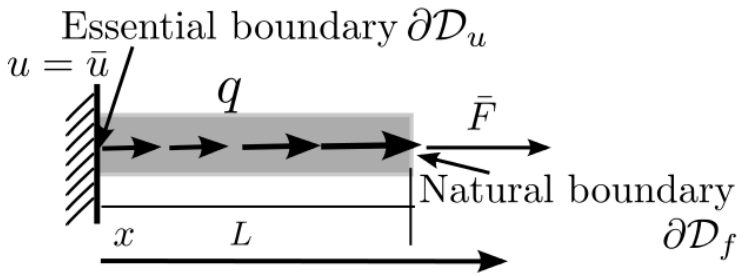
- $\mathbf{F}_r$, $\mathbf{F}_N$, and $\mathbf{F}_D$ are given by,

$$\mathbf{F}_r = \left(\boldsymbol{\phi}^{\mathrm{T}}, q\right)_r = \int_{\mathcal{D}} [\boldsymbol{\phi}]^{\mathrm{T}} q \,\mathrm{dv} = \int_0^L \begin{bmatrix} \phi_1 \\ \phi_2 \\ \vdots \\ \phi_{n_\mathrm{f}} \end{bmatrix} q \,\mathrm{d}x \tag{301a}$$

$$\mathbf{F}_N = \left(\boldsymbol{\phi}^{\mathrm{T}}, \bar{F}\right)_N = \int_{\partial\mathcal{D}_f} [\boldsymbol{\phi}]^{\mathrm{T}} \bar{\mathbf{F}}.\mathbf{N} \,\mathrm{ds} = \left(\begin{bmatrix} \phi_1 \\ \phi_2 \\ \vdots \\ \phi_{n_\mathrm{f}} \end{bmatrix} \bar{F} \right)_{x=L} \tag{301b}$$

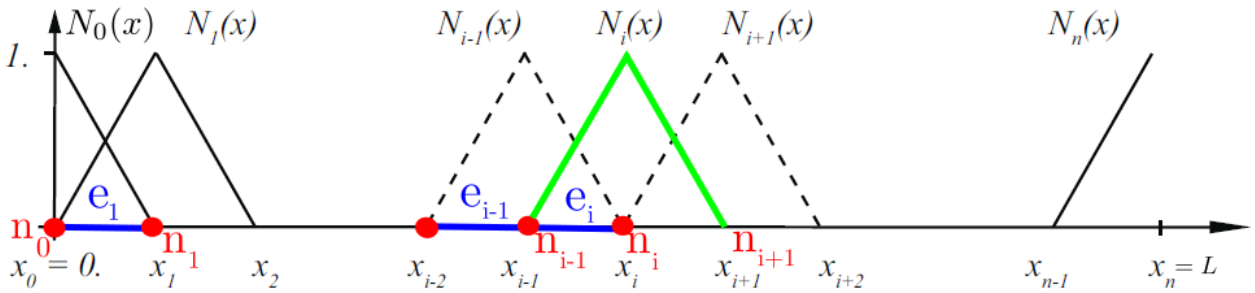
$$\mathbf{F}_D = \mathcal{A}\left(\boldsymbol{\phi}^{\mathrm{T}}, \phi_p\right) = \int_{\mathcal{D}} \frac{\mathrm{d}}{\mathrm{d}x} [\boldsymbol{\phi}]^{\mathrm{T}} EA \frac{\mathrm{d}}{\mathrm{d}x} \phi_p \,\mathrm{dv} = \int_0^L \frac{\mathrm{d}}{\mathrm{d}x} \begin{bmatrix} \phi_1 \\ \phi_2 \\ \vdots \\ \phi_{n_\mathrm{f}} \end{bmatrix} EA \frac{\mathrm{d}}{\mathrm{d}x} \phi_p \,\mathrm{d}x \tag{301c}$$

Discrete Galerkin formulation for solid bar



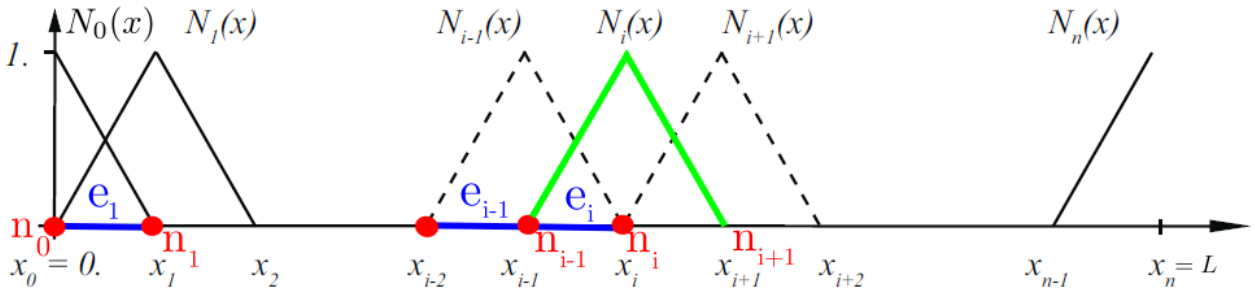
- We use the shorthand bilinear and linear forms for the expressions of the integrals used in the weak statement.
- We separated the contributions to the force vector $\mathbf{F} = \mathbf{F}_r + \mathbf{F}_N - \mathbf{F}_D$ as the contributions from the source term, natural boundary condition, and essential boundary condition.

Finite Element Method: Global perspective



- Computational Domain: The span $[0\ L]$.
- Nodes: The domain is discretized with n_{n} nodes. Nodes n_0, n_{i_1}, n_i and n_{i+1} are shown in the figure.
- Elements: are the line segments obtained by the discretization of the domain by nodes. Elements, e_1, e_{i-1} and e_i are shown in the figure. Number of elements is denoted by n_{e}.
- Shape functions: are especial kind of functions that take that value at one node and zero at other nodes. We denote shape functions by N_i. In Finite Element method **shape functions are used for trial functions**. Shape function $N_i(x)$ is highlighted in the figure.

Properties of Shape functions



- Shape function i **takes that value 1 at node i and takes the value of zero at all other nodes.**:

$$\boxed{N_i(x_j) = \delta_{ij}} \tag{302}$$

- Since $u^h(x) = a_i N_i(x) + \phi_p(x)$, $u^h(x_j) = a_i N_i(x_j) + \phi_p(x_j) = a_i \delta_{ij} + \phi_p(x_j) = a_j + \phi_p x_j$. We will shortly observe that ϕ_p is also constructed using shape functions and:

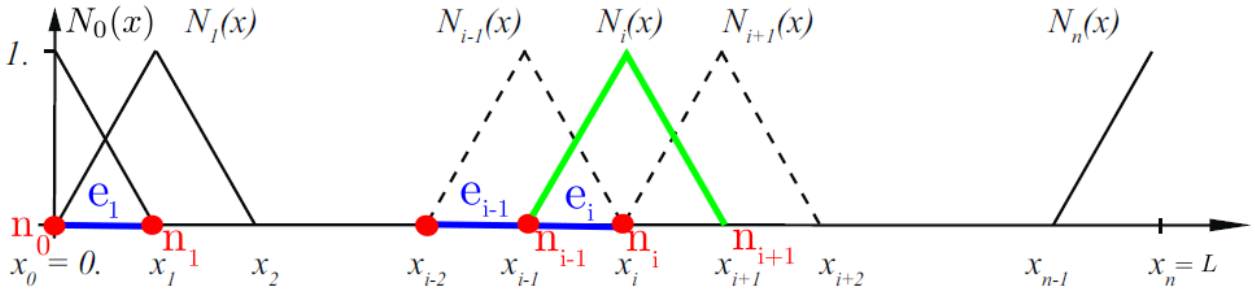
$$\boxed{u^h(x_j) = a_j} \tag{303}$$

- Shape functions are piece-wise linear. Based on our prior discussion about function spaces:

$$N_i \in H^1(\mathcal{D})$$

  that is, although they are not C^1 (1 being highest derivative order in the weak form) they can still be used in the finite element formulation).
- Shape functions have **local support**. This leads to **banded structure** in **stiffness matrix**.

Local support

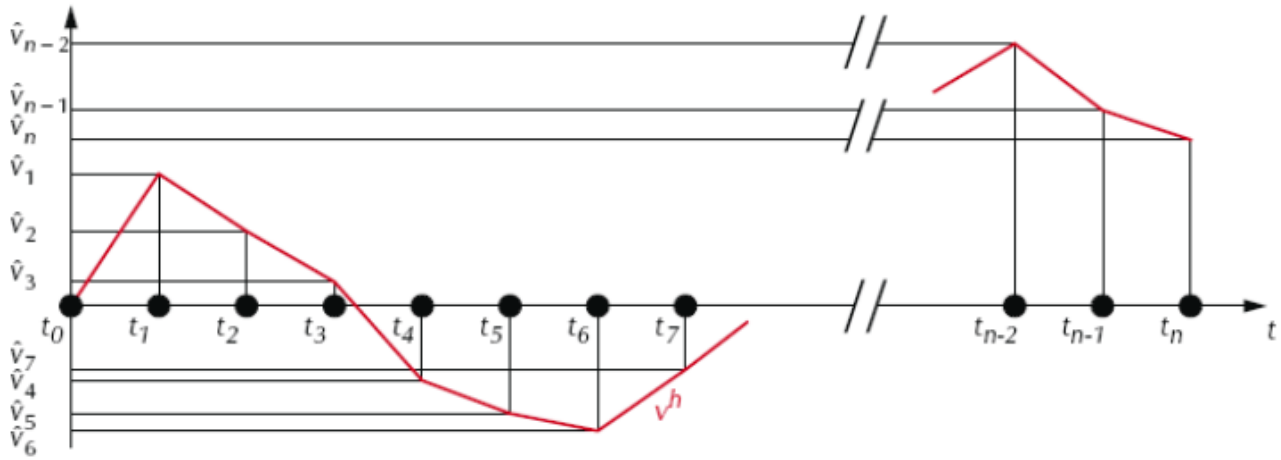


$$N_0(x) = \begin{cases} (x_1 - x)/(x_1 - x_0) & 0. \le x \le x_1 \\ 0. & x_1 \le x \le x_n \end{cases}$$

$$N_i(x) = \begin{cases} 0. & x_0 \le x \le x_{i-1} \\ (x - x_{i-1})/(x_i - x_{i-1}) & x_{i-1} \le x \le x_i \\ (x_{i+1} - x)/(x_{i+1} - x_i) & x_i \le x \le x_{i+1} \\ 0. & x_{i+1} \le x \le x_n \end{cases}$$

$$N_n(x) = \begin{cases} 0. & x_0 \le x \le x_{n-1} \\ (x - x_{n-1})/(x_n - x_{n-1}) & x_{n-1} \le x \le x_n \end{cases}$$

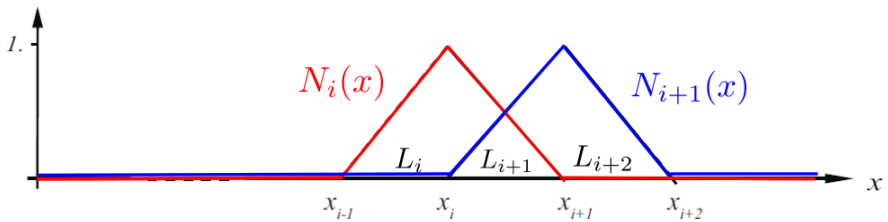
Interpretation of nodal degrees of freedom

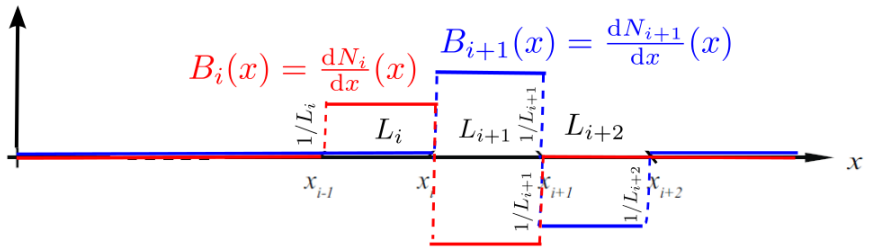


Solution unknowns are nodal solution.

$$u^h(t_j) = v_j \ (u^h(x_j) = a_j)$$

A. Strain-Displacement Matrix





- In bilinear form we have (288a),

$$\mathcal{A}(w,u) = \int_{\mathcal{D}} \frac{\mathrm{d}w}{\mathrm{d}x} EA \frac{\mathrm{d}u}{\mathrm{d}x}\,\mathrm{d}x = \int_{\mathcal{D}} \epsilon(w(x)) \overbrace{EA\epsilon(u(x))}^{\sigma(u(x))}\,\mathrm{d}x$$

- Accordingly we define the displacement to strain matrix,

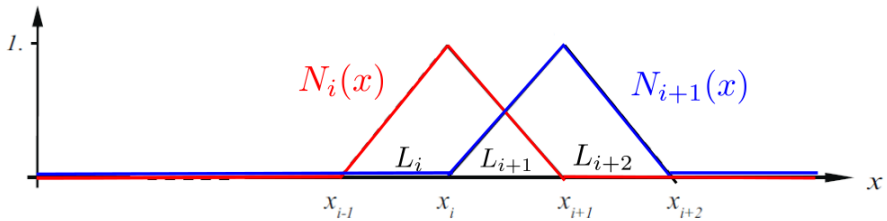
$$\boxed{\mathbf{B}(x) = \frac{\mathrm{d}\mathbf{N}(x)}{\mathrm{d}x} = \left[\frac{\mathrm{d}N_1(x)}{\mathrm{d}x}, \frac{\mathrm{d}N_2(x)}{\mathrm{d}x}, \ldots, \frac{\mathrm{d}N_{n_\mathrm{f}}(x)}{\mathrm{d}x}\right]} \tag{304}$$

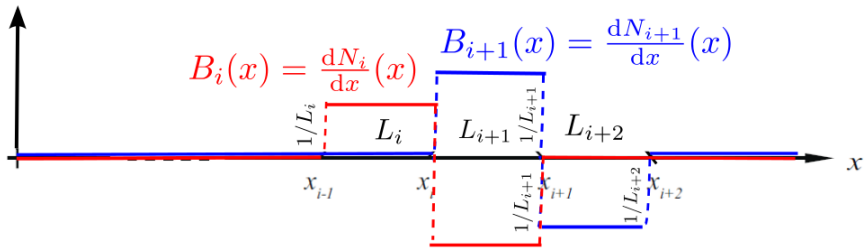
- and in component form

$$\boxed{B_I(x) = \frac{\mathrm{d}N_I(x)}{\mathrm{d}x}} \tag{305}$$

- In the figure $B_i(x)$ and $B_{i+1}(x)$ are shown.
- In general, $\mathbf{B}$ is obtained from L_m differential operator in the weak statement (excluding material properties such as AE, IE, *etc.*). For example for the beam problem $\mathcal{A}(w,y) = \int_0^L \frac{\mathrm{d}^2 w}{\mathrm{d}x^2} EI \frac{\mathrm{d}^2 y}{\mathrm{d}x^2}\,\mathrm{d}x$ and corresponding $\mathbf{B}$ is $\mathbf{B} = \frac{\mathrm{d}^2}{\mathrm{d}x^2}\mathbf{N}$.

A. Strain-Displacement Matrix





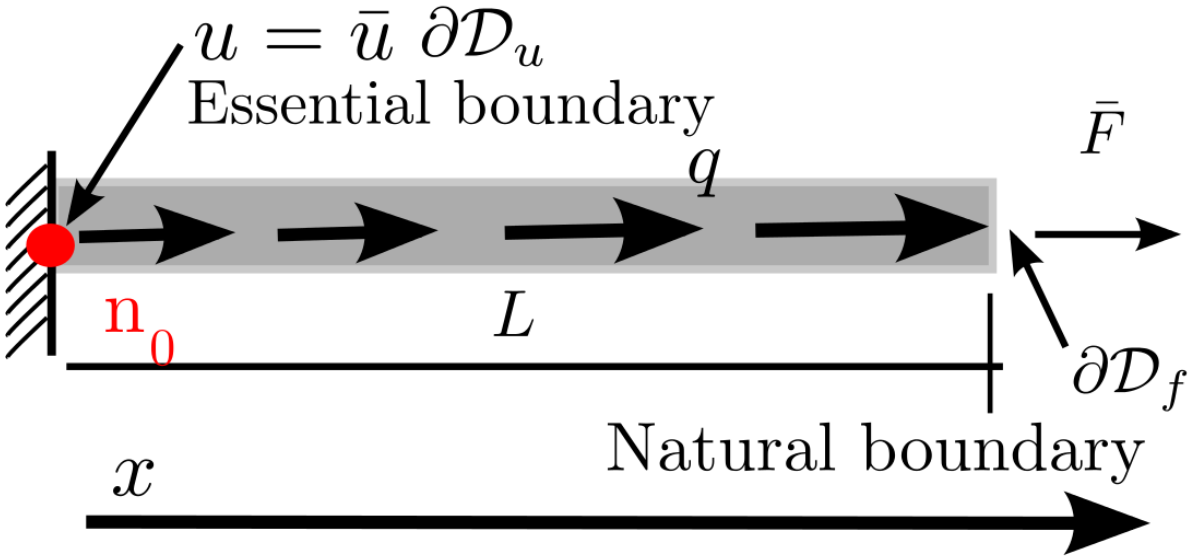
- From (300a) and (304) we have,

$$\mathbf{K} = \mathcal{A}\left(\mathbf{N}^{\mathrm{T}}, \mathbf{N}\right) = \int_{\mathcal{D}} \frac{\mathrm{d}}{\mathrm{d}x}\mathbf{N}^{\mathrm{T}} EA \frac{\mathrm{d}}{\mathrm{d}x}\mathbf{N}\,\mathrm{dv} \quad \Rightarrow$$

$$\mathbf{K} = \mathcal{A}\left(\mathbf{N}^{\mathrm{T}}, \mathbf{N}\right) = \int_{\mathcal{D}} \mathbf{B}^{\mathrm{T}} EA\mathbf{B}\,\mathrm{dv} = \int_0^L \begin{bmatrix} B_1 \\ B_2 \\ \vdots \\ B_{n_{\mathrm{f}}} \end{bmatrix} EA \begin{bmatrix} B_1 & B_2 & \cdots & B_{n_{\mathrm{f}}} \end{bmatrix} \mathrm{d}x \tag{306}$$

Note that $\mathbf{K}$ is an $n_{\mathrm{f}} \times n_{\mathrm{f}}$ matrix.

B. Essential Boundary Conditions

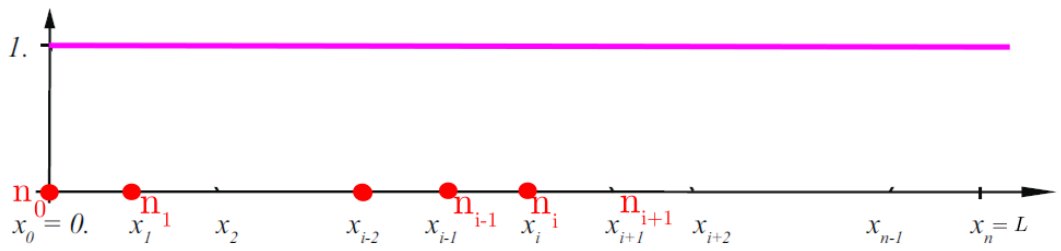


- We mentioned that the approximate solution u^h must satisfy the essential boundary condition.
- To satisfy essential boundary condition in the expansion $u^h = a_i\phi_i + \phi_p$ we required:

$$\phi_i(x) = 0 \qquad \forall x \in \partial\mathcal{D}_u$$
$$\phi_p(x) = \bar{u}(x) \qquad \forall x \in \partial\mathcal{D}_u$$

- In the sample problem $\partial\mathcal{D}_f = \{0\}$ is comprised of one point.

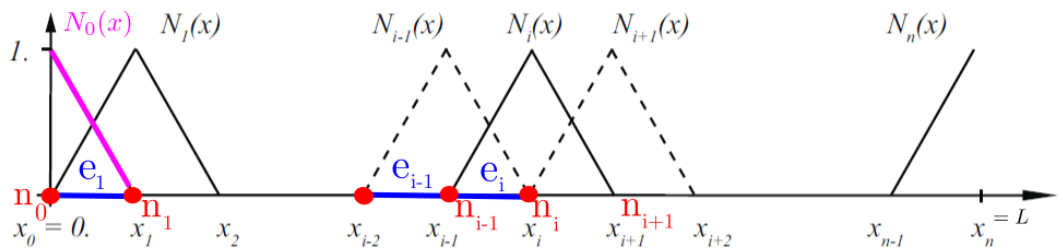
B. Essential Boundary Conditions



- Previously to satisfy the essential conditions for ϕ_p we used,

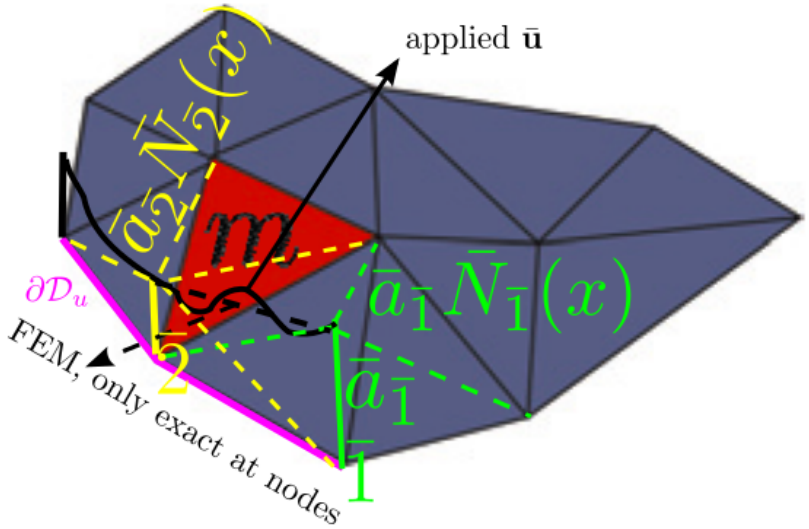
$$\phi_p = \bar{u}1 \quad \Rightarrow \quad \phi_p(0) = \bar{u}$$

- However, Finite Element Shape functions provide a natural way to construct ϕ_p by using (302): $N_i(x_j) = \delta_{ij}$



- We let, $\phi_p = \bar{u}N_0 \quad \Rightarrow \phi_p(0) = \phi_p(x_0) = \bar{u}N_0(x_0) = \bar{u}\delta_{11} = \bar{u}$ (307)
- It is clear that for all the trial functions (*i.e.*, shape functions corresponding to unknowns - $I \in \{N_1, \ldots, N_{n_f}\}$, $N_I(0) = N_I(x_0) = \delta_{I0} = 0$. That is, trial functions satisfy homogeneous essential boundary condition.

B. Essential Boundary Conditions



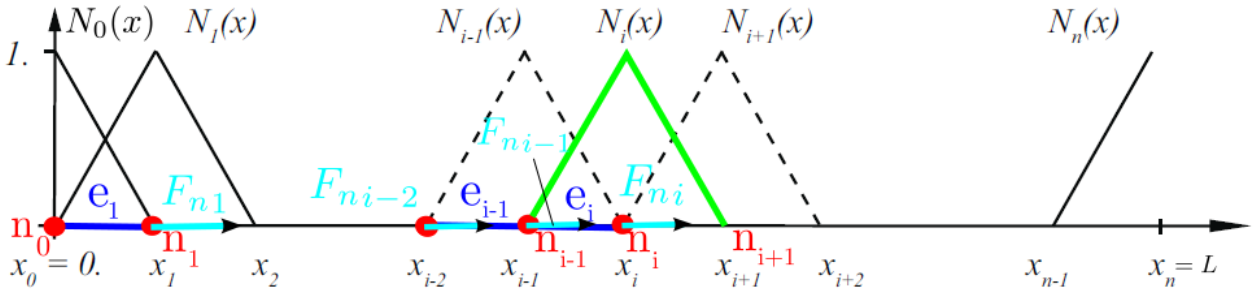
$n_{\bar{\mathrm{p}}}$	:= number of (prescribed) dof on $\partial\mathcal{D}_u$	
$(\bar{.})$	= decoration for prescribed dofs	
$\{\bar{1},\ldots,\bar{n_{\mathrm{p}}}\}$	= global dofs on $\partial\mathcal{D}_u$	
$\bar{\mathbf{a}} = \left[\bar{a}_{\bar{1}},\ldots,\bar{a}_{\bar{n_{\mathrm{p}}}}\right]^{\mathrm{T}}$	= vector of prescribed values for these dofs	(308)
$\bar{\mathbf{N}} = \left[\bar{N}_{\bar{1}},\ldots,\bar{N}_{\bar{n_{\mathrm{p}}}}\right]$	= (row) vector of shape functions for these dofs	
$\bar{\mathbf{B}} = \left[\bar{B}_{\bar{1}},\ldots,\bar{B}_{\bar{n_{\mathrm{p}}}}\right]$	= (row) vector of "displacement to strain" for these dofs	

- ϕ_p is formed by,

$$\phi_p = \bar{\mathbf{N}}\bar{\mathbf{a}} = \sum_{\bar{i}=1}^{n_{\mathrm{p}}} \bar{a}_{\bar{i}}\bar{N}_{\bar{i}} \;\Rightarrow\; \frac{\mathrm{d}\phi_p}{\mathrm{d}x} = \bar{\mathbf{B}}\bar{\mathbf{a}} = \sum_{\bar{i}=1}^{n_{\mathrm{p}}} \bar{a}_{\bar{i}}\bar{B}_{\bar{i}} \tag{309}$$

- While this construction guarantees satisfaction of essential boundary conditions at nodes $\bar{1},\ldots,\bar{n_{\mathrm{p}}}$, as shown in the figure the intermediate values may not match $\bar{u}$.

C. Concentrated forces



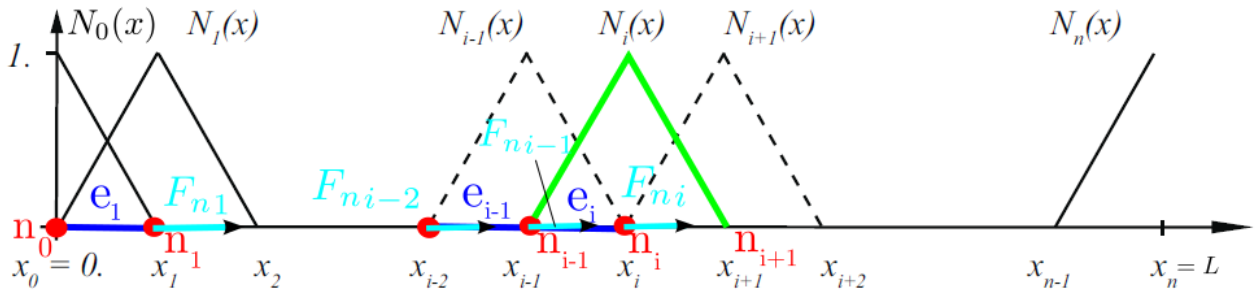
- From (292) and the fact that in FEM $\phi_I = N_I$ the weak statement for I^{th} weight function (corresponding to I^{th} rows of stiffness matrix and force vector) was:

$$\int_{\mathcal{D}} \frac{\mathrm{d}N_I}{\mathrm{d}x} EA \frac{\mathrm{d}u^h}{\mathrm{d}x}\, \mathrm{d}x = \int_{\mathcal{D}} N_I.q\, \mathrm{d}x + \left(N_I.\bar{F}\right)_{x=L}$$

- The LHS is the Virtual Internal Work by N_I.
- The RHS is the Virtual External Work by N_I.
- If we add concentrated forces $F_{n\,J}$ at nodes n_J we
  need to add the virtual external work of concentrated forces to the weak statement:

$$\int_{\mathcal{D}} \frac{\mathrm{d}N_I}{\mathrm{d}x} EA \frac{\mathrm{d}u^h}{\mathrm{d}x}\, \mathrm{d}x = \int_{\mathcal{D}} N_I.q\, \mathrm{d}x + \left(N_I.\bar{F}\right)_{x=L} + N_I(x_J) F_{n\,J}$$

C. Concentrated forces



- From (302) we note that,

$$N_I(x_J)F_{nJ} = \delta_{IJ}F_{nJ} = F_{nI}$$

- The weak statement is,

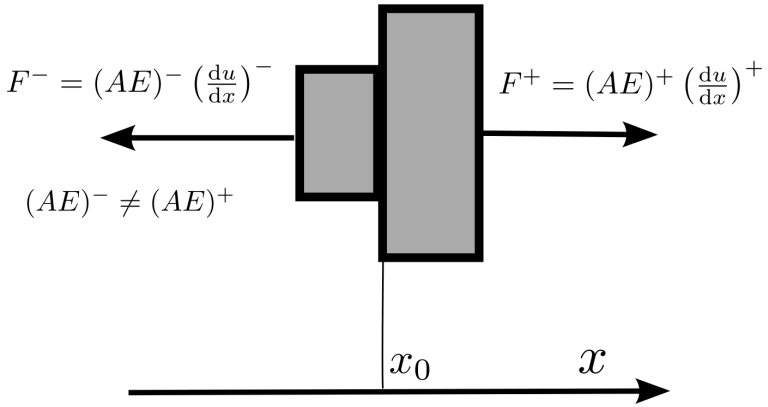
$$\int_{\mathcal{D}} \frac{\mathrm{d}N_I}{\mathrm{d}x} EA \frac{\mathrm{d}u^h}{\mathrm{d}x}\,\mathrm{d}x = \int_{\mathcal{D}} N_I.q\,\mathrm{d}x + \left(N_I.\bar{F}\right)_{x=L} + F_{nI}$$

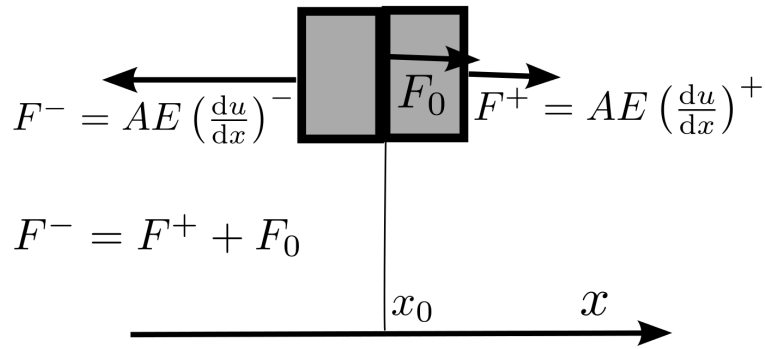
- Accordingly the force vector

$$\mathbf{F}_n = \begin{bmatrix} F_{n1} \\ F_{n2} \\ \vdots \\ F_{nn_\mathrm{f}} \end{bmatrix} \tag{310}$$

should be added to $\mathbf{F}$.

(Optional): Sources of discontinuity in analytical solution





- Jump in AE:

$$\left.\begin{aligned} F^- &= (AE)^- \left(\frac{\mathrm{d}u}{\mathrm{d}x}\right)^- \\ F^+ &= (AE)^+ \left(\frac{\mathrm{d}u}{\mathrm{d}x}\right)^+ \\ (AE)^- &\neq (AE)^+ \end{aligned}\right\} \Rightarrow$$

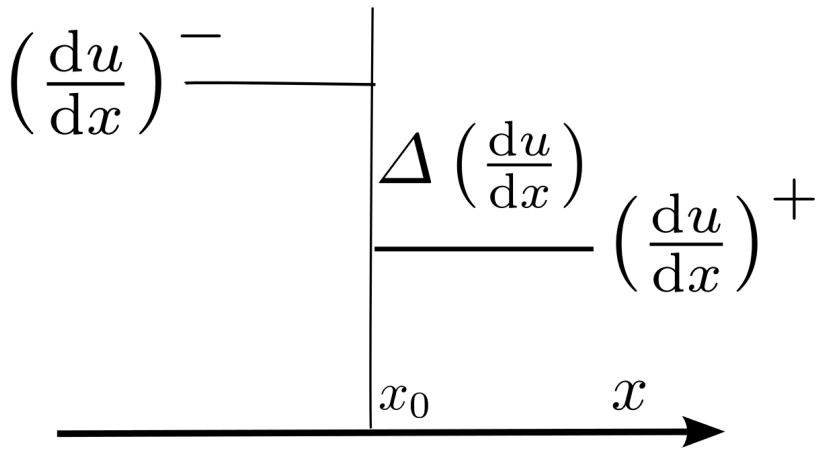
$$\boxed{\left(\frac{\mathrm{d}u}{\mathrm{d}x}\right)^- \neq \left(\frac{\mathrm{d}u}{\mathrm{d}x}\right)^+}$$

- Jump in F:

$$\left.\begin{aligned} F^- &= (AE) \left(\frac{\mathrm{d}u}{\mathrm{d}x}\right)^- \\ F^+ &= (AE) \left(\frac{\mathrm{d}u}{\mathrm{d}x}\right)^+ \\ F^- &= F^+ + F_0 \end{aligned}\right\} \Rightarrow$$

$$\boxed{\left(\frac{\mathrm{d}u}{\mathrm{d}x}\right)^- \neq \left(\frac{\mathrm{d}u}{\mathrm{d}x}\right)^+}$$

(Optional): Sources of discontinuity in analytical solution

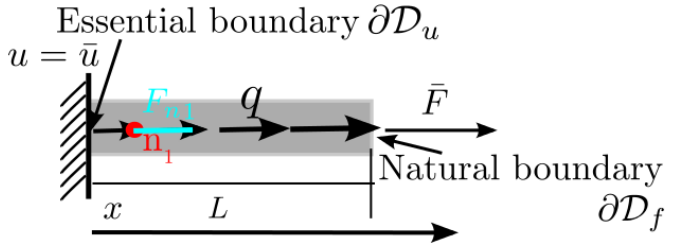


- We observe that abrupt changes to AE or a concentrated load cause a jump in $\frac{\mathrm{d}u}{\mathrm{d}x}$.
- That is $\frac{\mathrm{d}u}{\mathrm{d}x}$ does not exist at x_0 in classical sense.
- The analytical solution u is not C^1.
- However, u is still H^1 (*e.g.*, $\int_{\mathcal{D}}(u^2 + u_{,x}^2)\,\mathrm{d}x < \infty$).
- That is analytical solution u belongs to the same space we construct finite element solutions using piece-wise linear shape functions.
- This is another reason why the more relaxed H^1 condition is more convenient than C^1 for the solution space.

Summary: Force vectors

- Force vector is given by:

$$\boxed{\mathbf{F} = \mathbf{F}_r + \mathbf{F}_N + \mathbf{F}_n - \mathbf{F}_D} \qquad (311)$$



- $\mathbf{F}_r$, $\mathbf{F}_N$, $\mathbf{F}_n$ and $\mathbf{F}_D$ are given by (*cf.* (301) and (310))

$$\mathbf{F}_r = \left(\mathbf{N}^{\mathrm{T}}, q\right)_r = \int_{\mathcal{D}} \mathbf{N}^{\mathrm{T}} q \,\mathrm{dv} = \int_0^L \begin{bmatrix} N_1 \\ \vdots \\ N_{n_\mathrm{f}} \end{bmatrix} q \,\mathrm{d}x \qquad (312a)$$

$$\mathbf{F}_N = \left(\mathbf{N}^{\mathrm{T}}, \bar{F}\right)_N = \int_{\partial\mathcal{D}_f} \mathbf{N}^{\mathrm{T}} \bar{\mathbf{F}}.\mathbf{N} \,\mathrm{ds} = \left(\begin{bmatrix} N_1 \\ \vdots \\ N_{n_\mathrm{f}} \end{bmatrix} \bar{F} \right)_{x=L} \qquad (312b)$$

$$\mathbf{F}_D = \mathcal{A}\left(\mathbf{N}^{\mathrm{T}}, \phi_p\right) = \int_{\mathcal{D}} \frac{\mathrm{d}}{\mathrm{d}x}\mathbf{N}^{\mathrm{T}} EA \frac{\mathrm{d}}{\mathrm{d}x}\phi_p \,\mathrm{dv} \qquad (312c)$$

$$= \left\{ \int_{\mathcal{D}} \mathbf{B}^{\mathrm{T}} EA \bar{\mathbf{B}} \,\mathrm{dv} \right\} \bar{\mathbf{a}} = \left\{ \int_0^L \begin{bmatrix} B_1 \\ \vdots \\ B_{n_\mathrm{f}} \end{bmatrix} EA \begin{bmatrix} \bar{B}_{\bar{1}} & \cdots & \bar{B}_{\bar{n}_\mathrm{p}} \end{bmatrix} \mathrm{d}x \right\} \begin{bmatrix} \bar{a}_{\bar{1}} \\ \vdots \\ \bar{a}_{\bar{n}_\mathrm{p}} \end{bmatrix} = \mathbf{K}_{fp}\bar{\mathbf{a}}$$

$$\mathbf{F}_n = \begin{bmatrix} F_{n1} \\ \vdots \\ F_{n n_\mathrm{f}} \end{bmatrix} \qquad (312d)$$

Force Essential Boundary Condition

- We have used (309) in (312c) to write,

$$\boxed{\mathbf{F}_D = \mathcal{A}\left(\mathbf{N}^{\mathrm{T}}, \phi_p\right) = \mathbf{K}_{fp}\bar{\mathbf{a}}} \tag{313}$$

- The prescribed to free stiffness matrix $\mathbf{K}_{fp}$ is an $n_{\mathrm{f}} \times n_{\mathrm{p}}$ matrix given by,

$$\mathbf{K}_{fp} = \int_{\mathcal{D}} \mathbf{B}^{\mathrm{T}} EA\bar{\mathbf{B}}\, \mathrm{dv} = \int_0^L \begin{bmatrix} B_1 \\ \vdots \\ B_{n_{\mathrm{f}}} \end{bmatrix} EA \begin{bmatrix} \bar{B}_{\bar{1}} & \cdots & \bar{B}_{\bar{n_{\mathrm{p}}}} \end{bmatrix} \mathrm{d}x \tag{314}$$

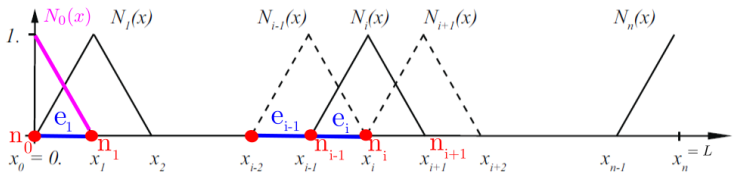
- From (306) we had,

$$\mathbf{K} = \mathcal{A}\left(\boldsymbol{\phi}^{\mathrm{T}}, \boldsymbol{\phi}\right) = \int_{\mathcal{D}} \mathbf{B}^{\mathrm{T}} EA\mathbf{B}\, \mathrm{dv} = \int_0^L \begin{bmatrix} B_1 \\ B_2 \\ \vdots \\ B_{n_{\mathrm{f}}} \end{bmatrix} EA \begin{bmatrix} B_1 & B_2 & \cdots & B_{n_{\mathrm{f}}} \end{bmatrix} \mathrm{d}x$$

  where $\mathbf{K}$ was an $n_{\mathrm{f}} \times n_{\mathrm{p}}$ matrix.
- "Prescribed" dofs $\bar{i}$ do not go into $\mathbf{K}$ because their value $\bar{a}_{\bar{i}}$ are already known.
- This is opposite to dofs $I = 1, \ldots, n_{\mathrm{f}}$ which correspond to "free" dofs.

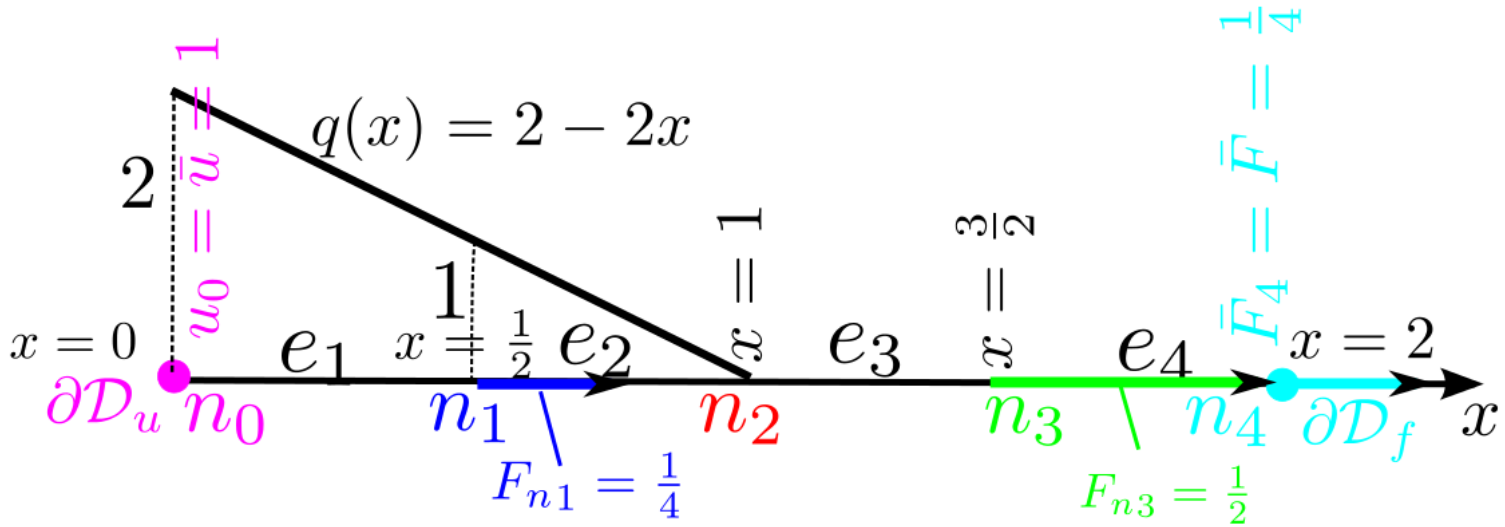
Force Essential Boundary Condition

For example for the problem in the figure $n_{\mathrm{p}} = 1$, $\bar{1} = 0$, $\bar{a}_{\bar{1}} = \bar{u} \Rightarrow$



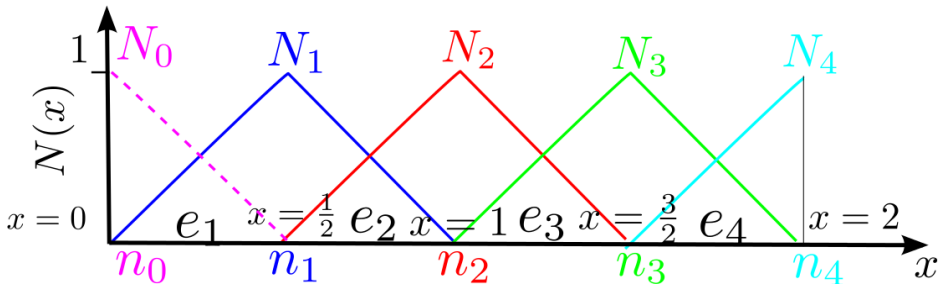
$$\mathbf{F}_D = \bar{a}_{\bar{1}} \int_0^L \begin{bmatrix} B_1 \\ \vdots \\ B_{n_{\mathrm{f}}} \end{bmatrix} EA\bar{B}_{\bar{1}}(x)\,\mathrm{d}x = \bar{u} \int_0^L \begin{bmatrix} B_1 \\ \vdots \\ B_{n_{\mathrm{f}}} \end{bmatrix} EAB_0(x)\,\mathrm{d}x \quad (315)$$

Bar Example: Overview

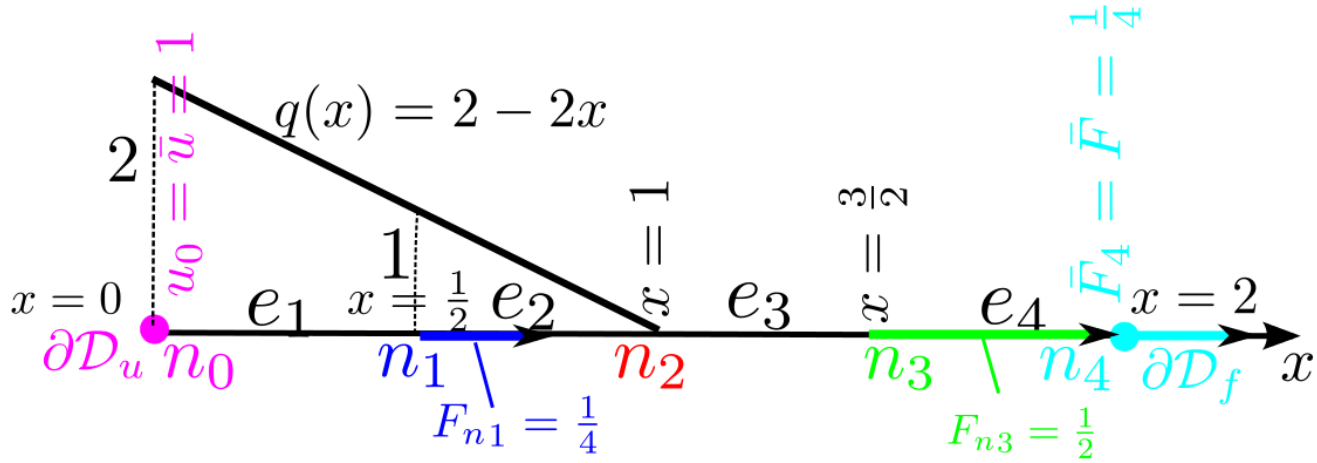


- We discretize the domain shown $\mathcal{D} = [0\ 2]$ to **four** elements e_1, e_2, e_3, e_4.
- The problem has **five** nodes n_0, n_1, n_2, n_3, n_4 at $x = 0, \frac{1}{2}, 1, \frac{3}{2}$ and 2 respectively.
- Nodes $\{n_1, n_2, n_3, n_4\}$ are **free** $\Rightarrow n_\mathrm{f} = 4$.
- Node n_0 is **prescribed (on $\partial\mathcal{D}_u$)** with the value $\bar{a}_{\bar{1}} = \bar{u} = 1 \Rightarrow n_\mathrm{p} = 1$.
- The material and section properties are chosen: $E = 1,\ A = 1$.

Finite element shape functions:



Bar Example: Sources of force



1. Source Term: is given by

$$q(x) = \begin{cases} 2 - 2x & x < 1 \\ 0 & 1 \leq x \leq 2 \end{cases}$$

2. Natural boundary condition: The force $\bar{F} = \frac{1}{4}$ is applied at $x = 2(n_4)$. Note $\partial\mathcal{D}_f = \{2\}$
3. Essential boundary condition: We enforce the condition $u^h(0) = \bar{u} = 1$. Note $\partial\mathcal{D}_u = \{0\} = \{n_0\}$.
4. Nodal Forces: are given by,

$$F_{n1} = \frac{1}{4}$$

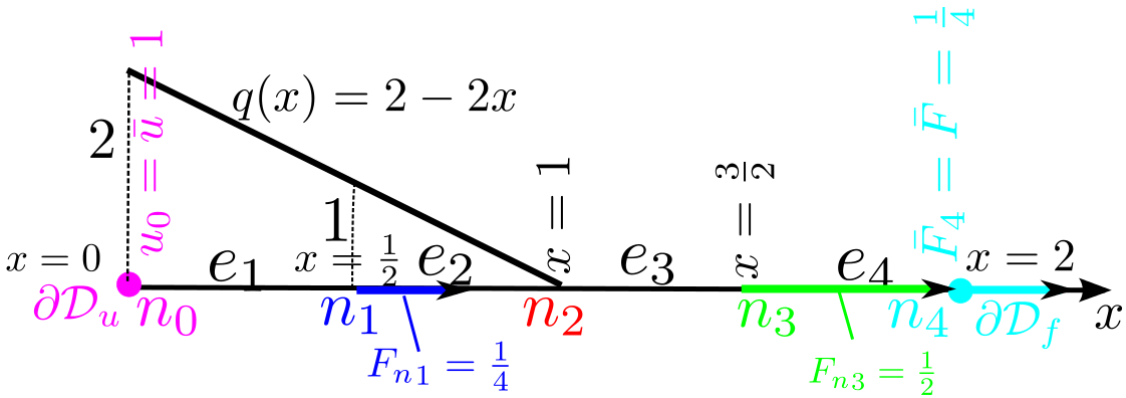
$$F_{n3} = \frac{1}{2}$$

Bar Example: Step 1: Stiffness matrix

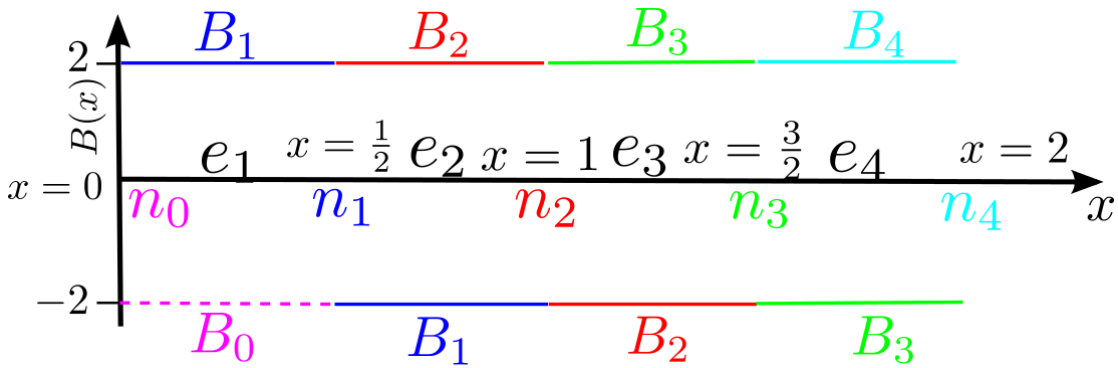
- From (306) we need to evaluate $\mathbf{B}$:

$$\mathbf{K} = \int_0^L \begin{bmatrix} B_1 \\ B_2 \\ \vdots \\ B_{n_\mathrm{f}} \end{bmatrix} EA \begin{bmatrix} B_1 & B_2 & \cdots & B_{n_\mathrm{f}} \end{bmatrix} \mathrm{d}x$$

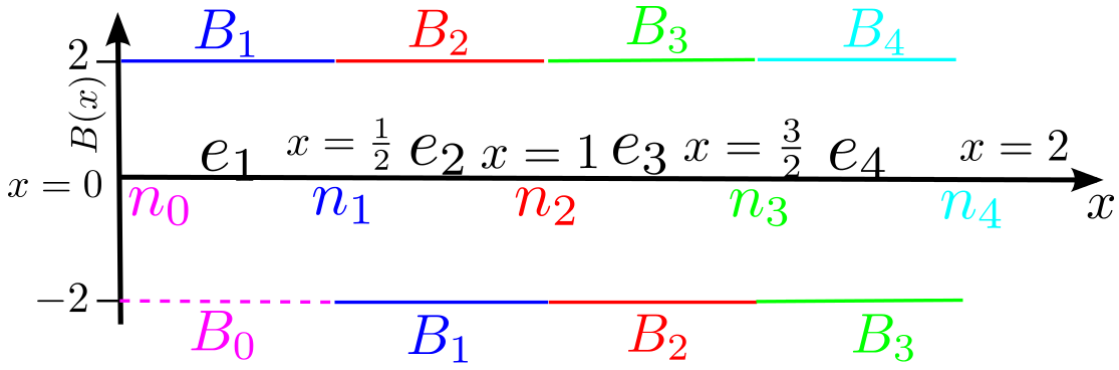
- Shape functions ϕ_I are,



- Displacement to strain $B_I = \frac{\mathrm{d}\phi_I(x)}{\mathrm{d}x}$ are then given by,



Bar Example: Step 1: Stiffness matrix



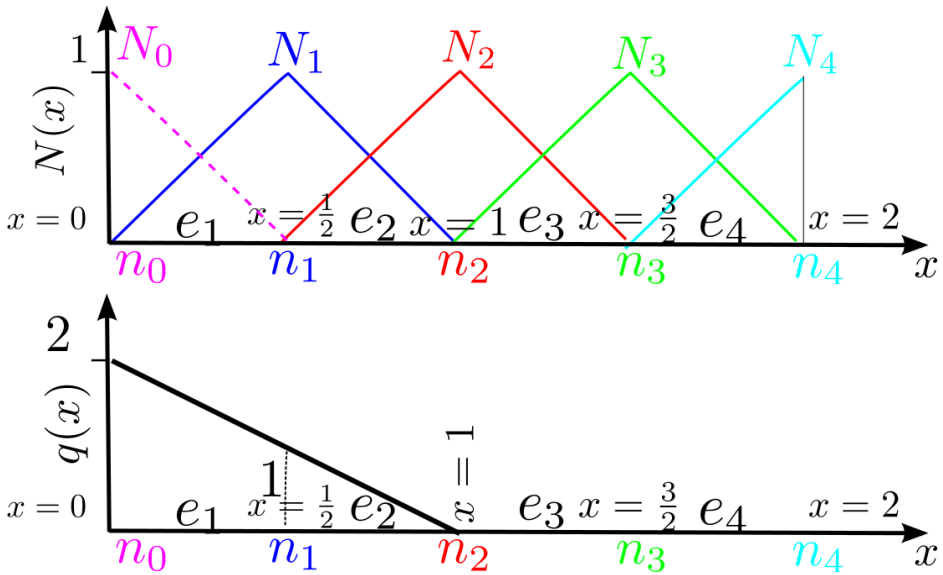
$$\mathbf{K} = \int_0^2 \begin{bmatrix} B_1 \\ B_2 \\ B_3 \\ B_4 \end{bmatrix} EA \begin{bmatrix} B_1 & B_2 & B_3 & B_4 \end{bmatrix} \mathrm{d}x = \begin{bmatrix} \int_0^2 B_1 B_1 \,\mathrm{d}x & \int_0^2 B_1 B_2 \,\mathrm{d}x & \int_0^2 B_1 B_3 \,\mathrm{d}x & \int_0^2 B_1 B_4 \,\mathrm{d}x \\ & \int_0^2 B_2 B_2 \,\mathrm{d}x & \int_0^2 B_2 B_3 \,\mathrm{d}x & \int_0^2 B_2 B_4 \,\mathrm{d}x \\ & & \int_0^2 B_3 B_3 \,\mathrm{d}x & \int_0^2 B_3 B_4 \,\mathrm{d}x \\ \text{sym.} & & & \int_0^2 B_4 B_4 \,\mathrm{d}x \end{bmatrix}$$

$$= \begin{bmatrix} \int_{e_1} B_1 B_1 \,\mathrm{d}x + \int_{e_2} B_1 B_1 \,\mathrm{d}x & \int_{e_2} B_1 B_2 \,\mathrm{d}x & 0 & 0 \\ & \int_{e_2} B_2 B_2 \,\mathrm{d}x + \int_{e_3} B_2 B_2 \,\mathrm{d}x & \int_{e_3} B_2 B_3 \,\mathrm{d}x & 0 \\ & & \int_{e_3} B_3 B_3 \,\mathrm{d}x + \int_{e_4} B_3 B_3 \,\mathrm{d}x & \int_{e_4} B_3 B_4 \,\mathrm{d}x \\ \text{sym.} & & & \int_{e_4} B_4 B_4 \,\mathrm{d}x \end{bmatrix}$$

$$= \begin{bmatrix} \frac{1}{2}.(2).(2) + \frac{1}{2}.(-2).(-2) & \frac{1}{2}.(-2).(2) & 0 & 0 \\ & \frac{1}{2}.(2).(2) + \frac{1}{2}.(-2).(-2) & \frac{1}{2}.(-2).(2) & 0 \\ & & \frac{1}{2}.(2).(2) + \frac{1}{2}.(-2).(-2) & \frac{1}{2}.(-2).(2) \\ \text{sym.} & & & \frac{1}{2}.(2).(2) \end{bmatrix} \Rightarrow$$

$$\mathbf{K} = \begin{bmatrix} 4 & -2 & 0 & 0 \\ & 4 & -2 & 0 \\ & & 4 & -2 \\ \text{sym.} & & & 2 \end{bmatrix} \tag{316}$$

Bar Example: Step 2.1: Source term force



From (312a),

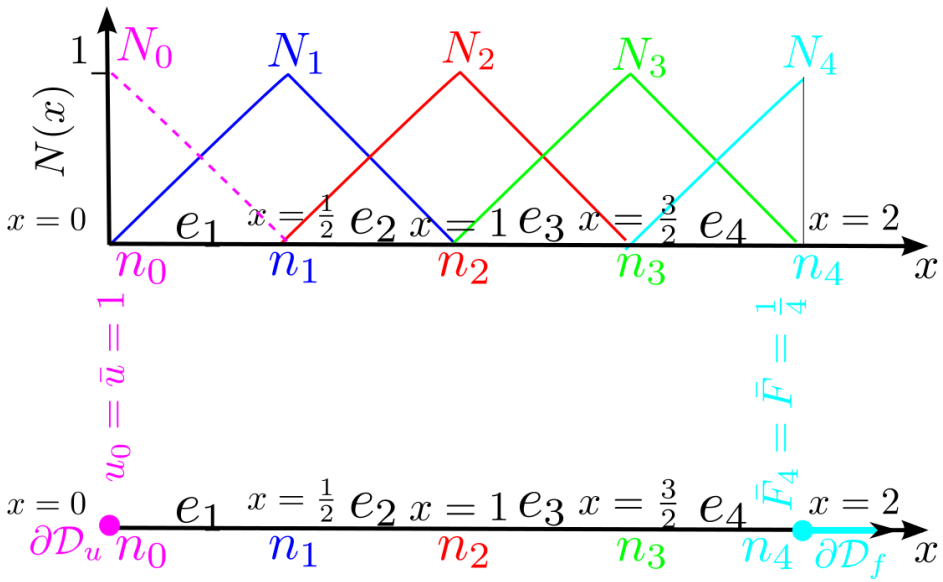
$$\mathbf{F}_r = \int_0^L \begin{bmatrix} N_1 \\ \vdots \\ N_{n_f} \end{bmatrix} q \, \mathrm{d}x = \begin{bmatrix} \int_0^2 N_1(x) q(x) \, \mathrm{d}x \\ \int_0^2 N_2(x) q(x) \, \mathrm{d}x \\ \int_0^2 N_3(x) q(x) \, \mathrm{d}x \\ \int_0^2 N_4(x) q(x) \, \mathrm{d}x \end{bmatrix} = \begin{bmatrix} \int_{e_1} N_1(x) q(x) \, \mathrm{d}x + \int_{e_2} N_1(x) q(x) \, \mathrm{d}x \\ \int_{e_2} N_2(x) q(x) \, \mathrm{d}x \\ 0 \\ 0 \end{bmatrix}$$

$$= \begin{bmatrix} \frac{1/2}{6} ((2).(0).(2) + (2).(1).(1) + (0).(1) + (1).(2)) + \frac{1/2}{6} ((2).(1).(1) + (2).(0).(0) + (1).(0) + (0).(1)) \\ \frac{1/2}{6} ((2).(0).(1) + (2).(1).(0) + (0).(0) + (1).(1)) \\ 0 \\ 0 \end{bmatrix}$$

$$\Rightarrow$$

$$\mathbf{F_r} = \begin{bmatrix} \frac{1}{2} \\ \frac{1}{12} \\ 0 \\ 0 \end{bmatrix} \tag{317}$$

Bar Example: Step 2.2: Natural BC force

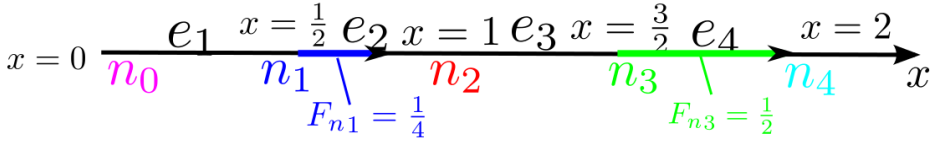


From (312b),

$$\mathbf{F}_N = \left(\mathbf{N}^{\mathrm{T}}, \bar{F}\right)_N = \int_{\partial\mathcal{D}_f} [\mathbf{N}]^{\mathrm{T}} \bar{\mathbf{F}}.\mathbf{N}\, \mathrm{ds} = \left(\begin{bmatrix} N_1 \\ \vdots \\ N_{n_{\mathrm{f}}} \end{bmatrix} \bar{F} \right)_{x=L} = \left(\begin{bmatrix} N_1 \\ N_2 \\ N_3 \\ N_4 \end{bmatrix} . \left(\frac{1}{4}\right) \right)_{x=2} = \left(\frac{1}{4}\right) \begin{bmatrix} N_1(n_4) \\ N_2(n_4) \\ N_3(n_4) \\ N_4(n_4) \end{bmatrix} = \begin{bmatrix} 0 \\ 0 \\ 0 \\ \frac{1}{4} \end{bmatrix}$$

$$\mathbf{F}_N = \begin{bmatrix} 0 \\ 0 \\ 0 \\ \frac{1}{4} \end{bmatrix} \tag{318}$$

Bar Example: Step 2.3: Nodal force



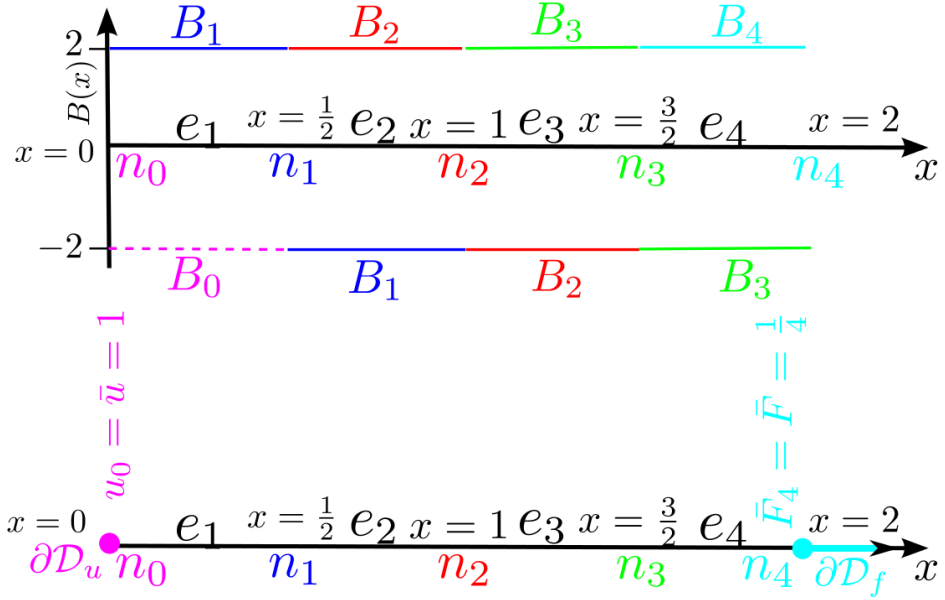
From (312d),

$$\mathbf{F}_n = \begin{bmatrix} F_{n\,1} \\ \vdots \\ F_{n\,n_{\mathrm{f}}} \end{bmatrix} = \begin{bmatrix} F_{n\,1} \\ F_{n\,2} \\ F_{n\,3} \\ F_{n\,4} \end{bmatrix} = \begin{bmatrix} \frac{1}{4} \\ 0 \\ \frac{1}{2} \\ 0 \end{bmatrix} \quad \Rightarrow$$

$$\mathbf{F}_n = \begin{bmatrix} \frac{1}{4} \\ 0 \\ \frac{1}{2} \\ 0 \end{bmatrix} \tag{319}$$

1. Note that the flux on the natural boundary $\partial\mathcal{D}_f = \{n_4\}$ in 1D case is a nodal force as well and could be directly added to $\mathbf{F}_n$ here.
2. In higher dimensions (2D and 3D), $\partial\mathcal{D}_f$ is a curve or a surface, respectively, and its contribution is different from that of a nodal force.

Bar Example: Step 2.4: Essential BC force



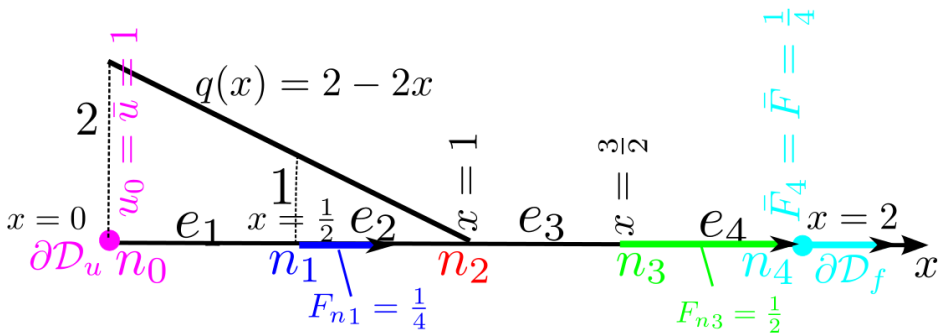
From (314) and (313), $n_{\rm p}=1$, $\bar{1}=0$, and $\bar{a}_{\bar{1}}=\bar{u}=1$ we get,

$$\mathbf{F}_D=\sum_{i=1}^{n_{\rm p}}\left\{\int_0^L\begin{bmatrix}B_1\\ \vdots\\ B_{n_{\rm f}}\end{bmatrix}EA\begin{bmatrix}\bar{B}_{\bar{1}} & \cdots\\ \bar{B}_{\bar{n_{\rm p}}} & \end{bmatrix}\mathrm{d}x\right\}\begin{bmatrix}\bar{a}_{\bar{1}}\\ \vdots\\ \bar{a}_{\bar{n_{\rm p}}}\end{bmatrix}=\left\{\int_0^L\begin{bmatrix}B_1(x)\\ B_2(x)\\ B_3(x)\\ B_4(x)\end{bmatrix}\bar{B}_{\bar{1}}(x)\,\mathrm{d}x\right\}\bar{u}$$

$$=\begin{bmatrix}\int_0^2 B_1(x).B_0(x)\,\mathrm{d}x\\ \int_0^2 B_2(x).B_0(x)\,\mathrm{d}x\\ \int_0^2 B_3(x).B_0(x)\,\mathrm{d}x\\ \int_0^2 B_4(x).B_0(x)\,\mathrm{d}x\end{bmatrix}\bar{u}=\begin{bmatrix}\int_{e_1} B_1(x).B_0(x)\,\mathrm{d}x\\ 0\\ 0\\ 0\end{bmatrix}\bar{u}=\begin{bmatrix}\frac{1}{2}.(2).(-2)\\ 0\\ 0\\ 0\end{bmatrix}.(1)\quad\Rightarrow$$

$$\mathbf{F}_D=\begin{bmatrix}-2\\ 0\\ 0\\ 0\end{bmatrix}\tag{320}$$

Bar Example: FEM Solution



- From (311) we have

$$\mathbf{F} = \mathbf{F}_r + \mathbf{F}_N + \mathbf{F}_n - \mathbf{F}_D$$

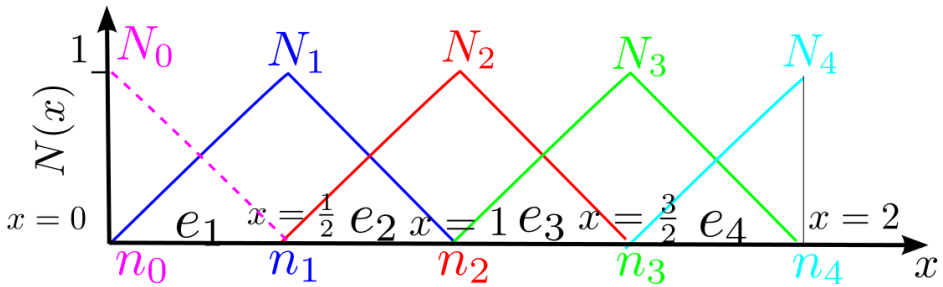
- Obtaining the individual values from (317), (318), (319), and (320) we obtain,

$$\mathbf{F} = \mathbf{F}_r + \mathbf{F}_N + \mathbf{F}_n - \mathbf{F}_D = \begin{bmatrix} \frac{1}{2} \\ \frac{1}{12} \\ 0 \\ 0 \end{bmatrix} + \begin{bmatrix} 0 \\ 0 \\ 0 \\ \frac{1}{4} \end{bmatrix} + \begin{bmatrix} \frac{1}{4} \\ 0 \\ \frac{1}{2} \\ 0 \end{bmatrix} - \begin{bmatrix} -2 \\ 0 \\ 0 \\ 0 \end{bmatrix} = \begin{bmatrix} \frac{11}{4} \\ \frac{1}{12} \\ \frac{1}{2} \\ \frac{1}{4} \end{bmatrix}$$

- Recalling the value for the stiffness matrix (316) and $\mathbf{Ka} = \mathbf{F}$ we obtain,

$$\mathbf{K} = \begin{bmatrix} 4 & -2 & 0 & 0 \\ & 4 & -2 & 0 \\ \text{sym.} & & 4 & -2 \\ & & & 2 \end{bmatrix}, \qquad \mathbf{F} = \begin{bmatrix} \frac{11}{4} \\ \frac{1}{12} \\ \frac{1}{2} \\ \frac{1}{4} \end{bmatrix} \quad \Rightarrow \quad \boxed{\mathbf{a} = \begin{bmatrix} \frac{43}{24} \\ \frac{53}{24} \\ \frac{31}{12} \\ \frac{65}{24} \end{bmatrix}} \qquad (321)$$

Bar Example: FEM Solution



- The FEM solution is,

$$u^h = a_i \phi_i + \phi_p = a_i N_i + \phi_p$$

- ϕ_p is given by (309),

$$\phi_p = \sum_{i=1}^{n_p} \bar{u}_{\bar{i}} N_{\bar{i}} = \bar{u}_0 N_0 = \bar{u} N_0 = 1 N_0 \quad \Rightarrow$$

$$\boxed{u^h = \bar{u} N_0 + \{a_1 N_1 + a_2 N_2 + a_3 N_3 + a_4 N_4\} = 1.N_0 + \frac{43}{24}.N_1 + \frac{53}{24}.N_2 + \frac{31}{12}.N_3 + \frac{65}{24}.N_4} \quad (322)$$

- What is $u^h(\frac{1}{2})$?

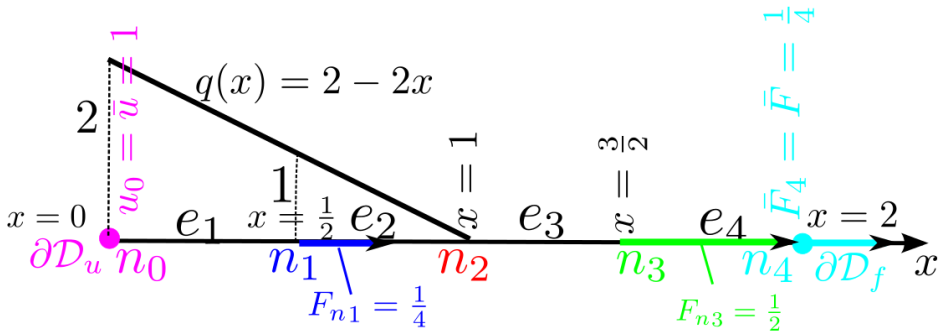
$$u^h(\frac{1}{2}) = u^h(n_1) = \bar{u} N_0(n_1) + a_1.N_1(n_1) + a_2.N_2(n_1) + a_3.N_3(n_1) + a_4.N_4(n_1)$$

$$= \bar{u}.\delta_{01} + a_1.\delta_{11} + a_2.\delta_{21} + a_3.\delta_{31} + a_4.\delta_{41} = +a_1 = \frac{43}{24}$$

- In general,

$$\boxed{u^h(n_I) = a_I} \quad (323)$$

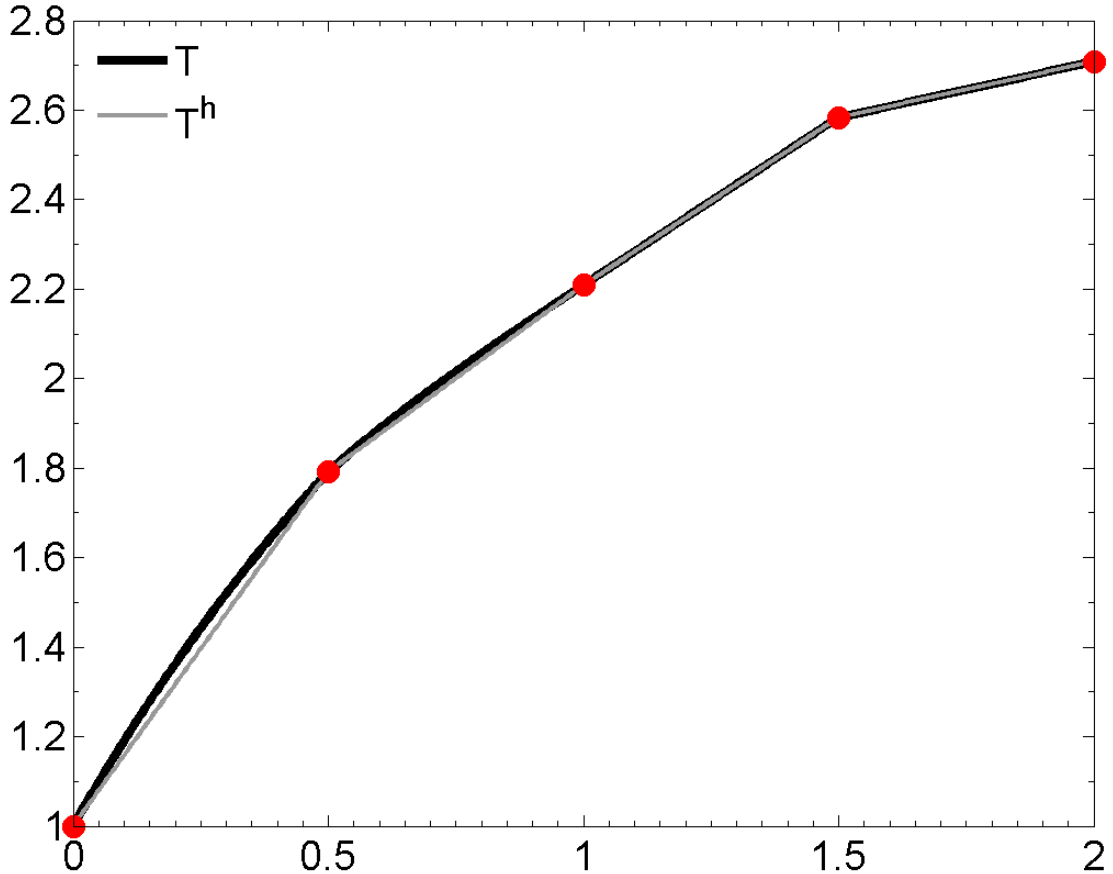
Bar Example: Exact solution



- The Exact solution u and its derivative $\frac{\mathrm{d}u}{\mathrm{d}x}$ are

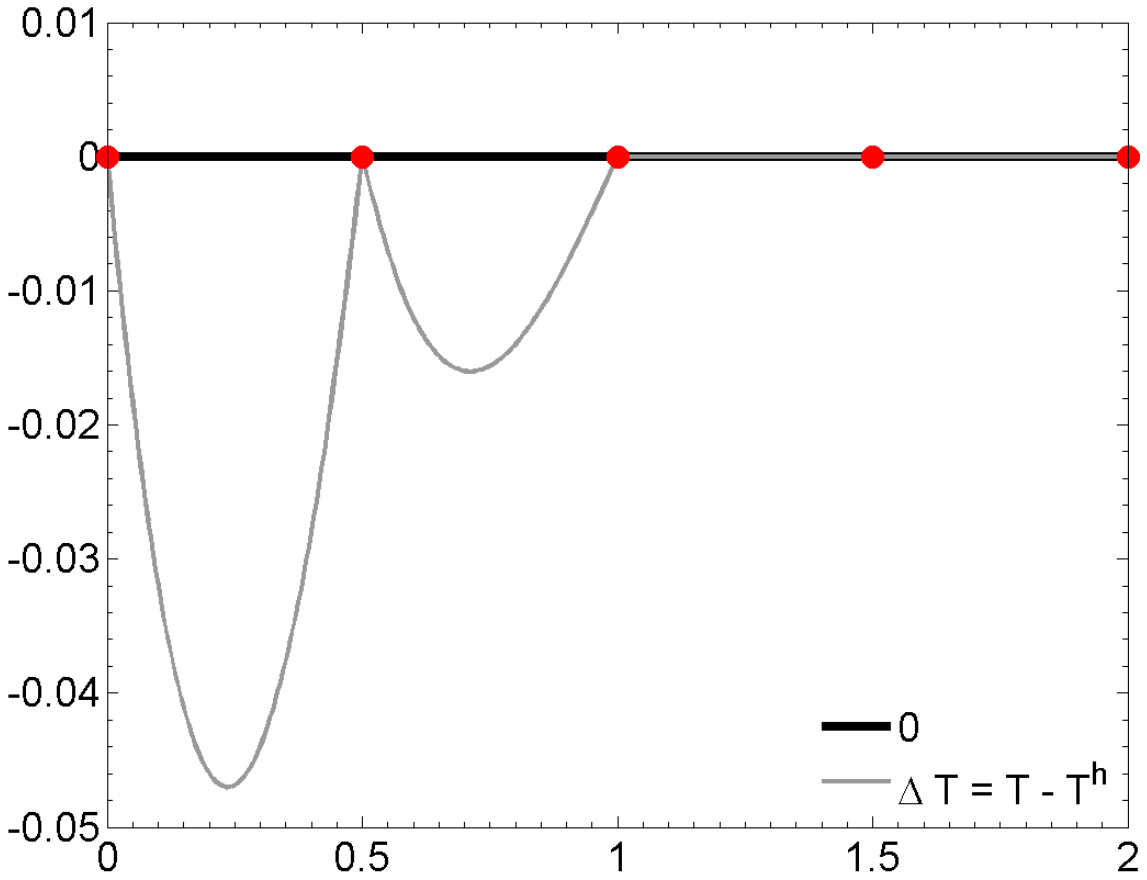
$$u = \begin{cases} \frac{x^3}{3} - x^2 + 2x + 1 & 0 \le x < \frac{1}{2} \\ \frac{x^3}{3} - x^2 + \frac{7}{4}x + \frac{9}{8} & \frac{1}{2} \le x < 1 \\ \frac{3}{4}x + \frac{35}{24} & 1 \le x < \frac{3}{2} \\ \frac{1}{4}x + \frac{53}{24} & \frac{3}{2} \le x \le 2 \end{cases} \qquad \frac{\mathrm{d}u}{\mathrm{d}x} = \begin{cases} x^2 - 2x + 2 & 0 \le x < \frac{1}{2} \\ x^2 - 2x + \frac{7}{4} & \frac{1}{2} \le x < 1 \\ \frac{3}{4} & 1 \le x < \frac{3}{2} \\ \frac{1}{4} & \frac{3}{2} \le x \le 2 \end{cases} \tag{324}$$

Bar Example: solution values



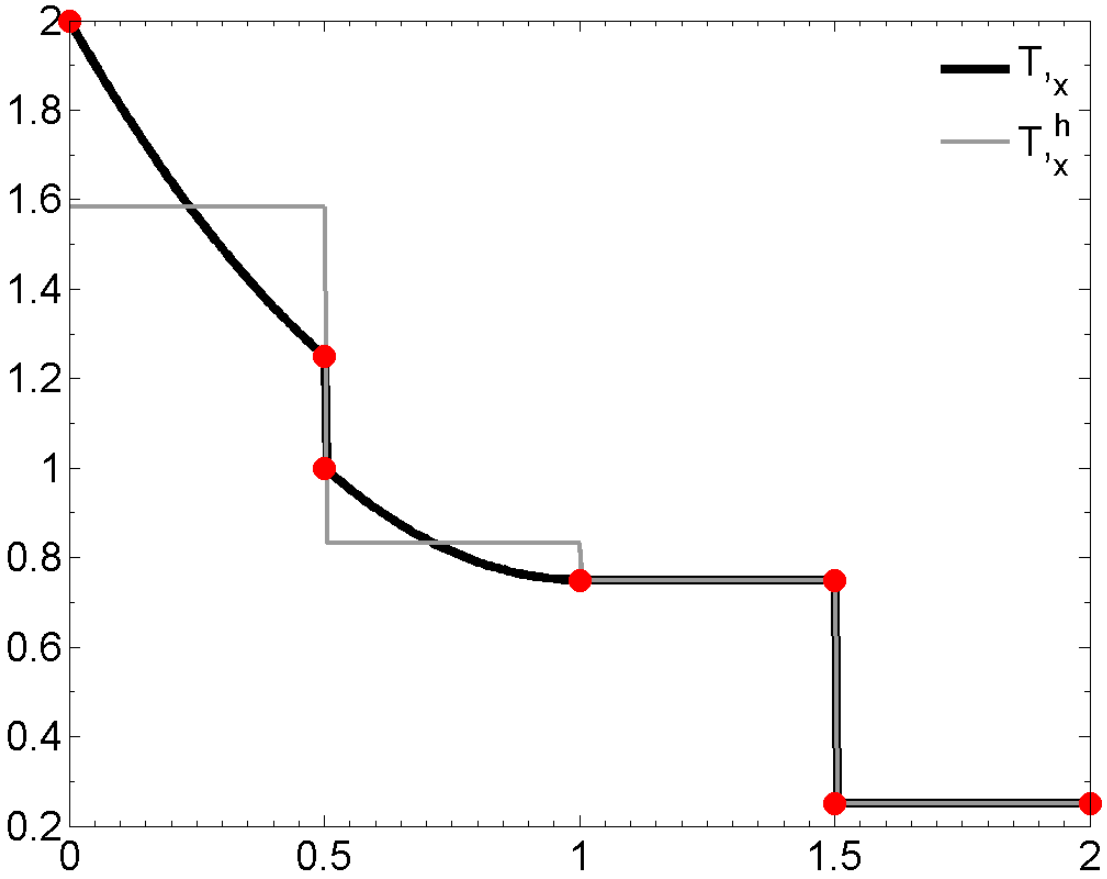
- u^h and u match at all nodes n_0, n_1, n_2, n_3, and n_4. This holds for 1D solid elements with uniform AE and **does not hold** in general.

Bar Example: error in solution values



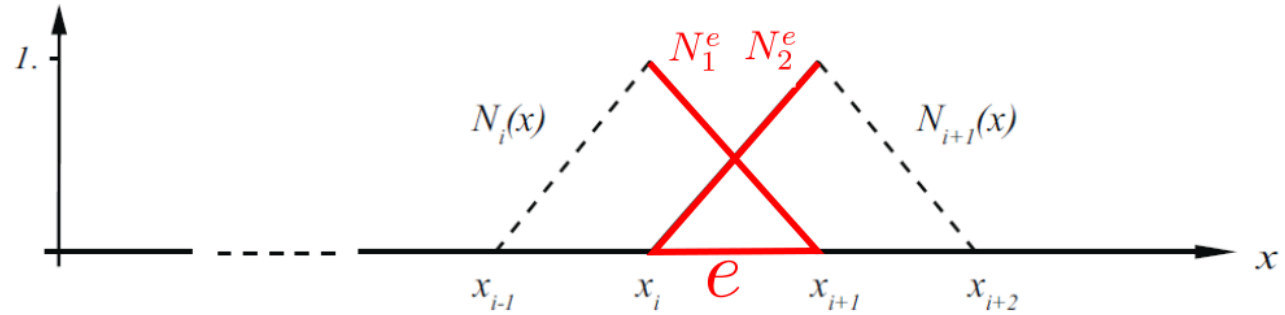
- As mentioned before, the solution error at all nodes n_0, n_1, n_2, n_3, and n_4 is zero. This does not hold in general for FEM method.

Bar Example: solution derivatives ($\propto$ axial force)



- The errors in solution derivative is larger than those in the solution itself. In general, the accuracy of FE solution decreases for solution derivatives (*e.g.*, strains, stresses, *etc.*).
- Approximate solution u^h exhibits jumps in $\frac{\mathrm{d}u^h}{\mathrm{d}x}$ at all interior nodes. This is because the solution is piece-wise constant in $H^1([0\ 2])$.
- Even the exact solution exhibits jumps in $\frac{\mathrm{d}u}{\mathrm{d}x}$ at n_1 and n_3 from the concentrated forces.
- The $H^1([0\ 2])$, rather than $C^1([0\ 2])$, is the right solution space for u and u^h as none of them belong to the latter space.

Global shape functions to element shape functions

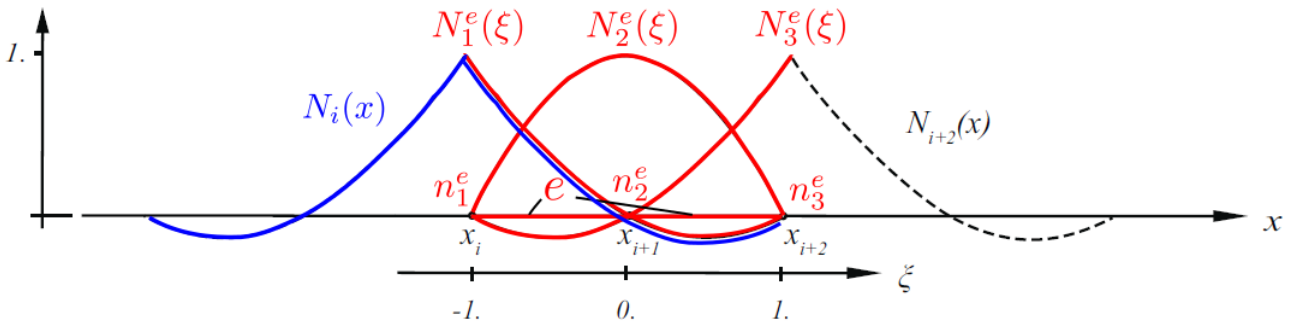


- Finite Elements are the domain subdivisions that are used for the construction of the shape functions
- Restriction of (global) shape functions to elements form the elements' shape functions (local).
- To distinguish element level and global level quantities, any element level value is decorated by $(.)^e$.
- Local node numbers in the element start from 1 to number of nodes in element n_{n}^e and are denoted by $n_1^e, \ldots, n_{n_{\mathrm{n}}^e}^e$.
- Similarly local dof start from 1 to the number of dof in element n_{dof}^e.
- For example in the figure both n_{n}^e and n_{dof}^e are both 2 and the range for local node number and dof is from 1 to 2.
- Element shape functions satisfy the condition,

$$N_i^e(n_j^e) = \delta_{ij} \tag{325}$$

- More generally (*e.g.*, beam elements), shape function i has a value 1 at dof i while has a value zero at all other element dofs.

Element shape functions to global shape functions

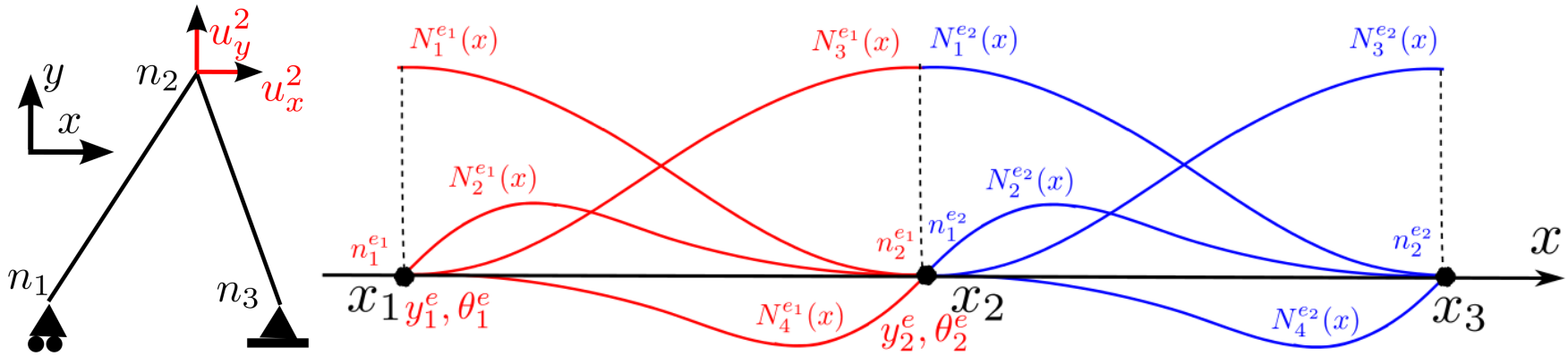


- While the global view of finite element has some advantages in mathematical analysis, we often form the shape functions at the local level and if needed form global shape functions.
- It was this local perspective that first was employed in engineering finite element analysis.
- For example, in the figure the 1D bar element has three nodes with one being internal node and has interpolation order $p = 2$.
- We observe that,

$$N_i^e(n_j^e) = \delta_{ij} \quad \Rightarrow N_I(n_J) = \delta_{IJ}$$

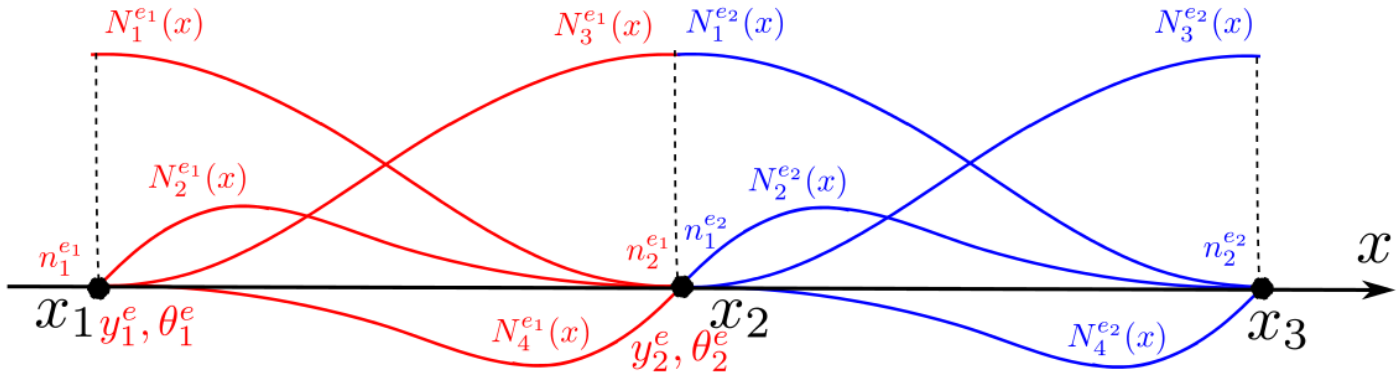
  which was the condition we first stipulated for finite elements in global view.
- As an example, we observe that the global shape function $N_i(x)$ is formed from local element shape functions.
- Notice that while local element order is $p = 2$ the global shape functions are still C^0 (piece-wise quadratic in this case).
- Elements can have internal nodes. This generally occurs for higher than linear elements ($p > 1$).

dof versus number of nodes



- While in the solid bar example there was a one to one correspondence between degrees of freedom (dof) = unknowns and nodes in general, a node may correspond to more than one unknown.
- We denote number of dof per node by n_{dpn}.
- Examples of cases where $n_{\text{dpn}} > 1$:
  1. Solving tensor fields as unknown: For example in the truss shown on the left we attempt to solve the vector field displacement as our unknown. In 2D, there are two displacement components u_x and u_y per node. $\Rightarrow n_{\text{dpn}} = 2$.
  2. Higher than C^0 continuity of the solution is required: For the beam example as we observed before the differential equation was $\frac{\mathrm{d}^4 y}{\mathrm{d}x^4} + q = 0 \quad \Rightarrow \quad M = 4$. Thus, the finite element solutions should be at least $C^{M/2-1} = C^1$. That is at each node we have two unknowns of function value y and derivative $\theta = \frac{\mathrm{d}y}{\mathrm{d}x}$.

dof versus number of nodes



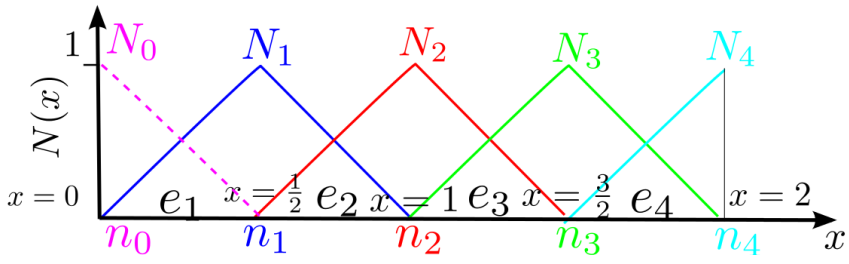
- In the beam shown the degrees of freedom are $\{y(x_1), \frac{\mathrm{d}y}{\mathrm{d}x}(x_1), y(x_2), \frac{\mathrm{d}y}{\mathrm{d}x}(x_2)\} = \{y(x_1), \theta(x_1), y(x_2), \theta(x_2)\}$. We notice that N_i^{e1} yields a value of 1 for dof i while having a zero value at other dofs.
- That is unknown coefficient i (a_i^{e1}) from finite element method would correspond to the solution value for dof i. For example,

$$\begin{aligned}\frac{\mathrm{d}y}{\mathrm{d}x}(x_2) = \theta(x_2) &= \frac{\mathrm{d}N_1^{e1}}{\mathrm{d}x}(x_2)a_1^{e1} + \frac{\mathrm{d}N_2^{e1}}{\mathrm{d}x}(x_2)a_2^{e1} + \frac{\mathrm{d}N_3^{e1}}{\mathrm{d}x}(x_2)a_3^{e1} + \frac{\mathrm{d}N_4^{e1}}{\mathrm{d}x}(x_2)a_4^{e1} \\ &= (0).a_1^{e1} + (0).a_2^{e1} + (0).a_3^{e1} + (1).a_4^{e1} = a_4^{e1}\end{aligned}$$

- It is clear that the condition $N_i^e(n_j^e) = \delta_{ij}$ does not hold in this more general case.
- Notice that the function is globally C^1.

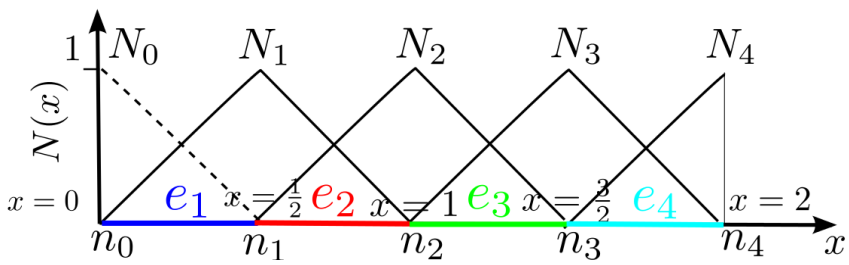
Finite Element Method: Global versus Local approach

1. **Global approach**: This approach is **shape function-centered** and we directly compute each component of the stiffness matrix and force vector by integration of shape functions (or their derivatives) over the entire domain $\mathcal{D}$. The integrals are carried out and summed over all the elements in $\mathcal{D}$.



$$\mathbf{K} = \int_0^2 \begin{bmatrix} B_1 \\ B_2 \\ B_3 \\ B_4 \end{bmatrix} EA \begin{bmatrix} B_1 & B_2 & B_3 & B_4 \end{bmatrix} \mathrm{d}x$$

2. **Local approach** is **element centered**: As we eventually the form of the shape functions change element to element, it is more convenient to first divide the integration domain, calculate element level matrices and vectors, and add them together:

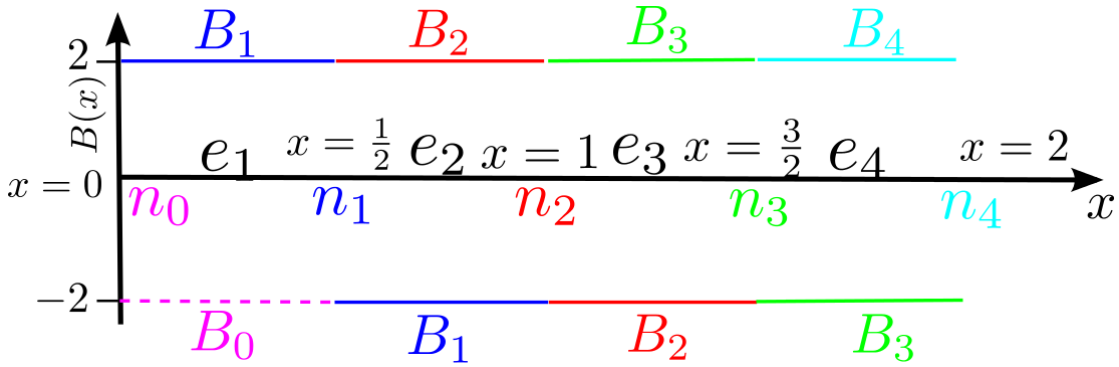


For example in the figure,

$$\mathbf{K} = \int_0^2 \mathbf{B}^T EA\mathbf{B}\, \mathrm{d}x = \mathbf{K}^{e1} + \mathbf{K}^{e2} + \mathbf{K}^{e3} + \mathbf{K}^{e4} \quad \text{where} \tag{326}$$

$$\mathbf{K}^{e1} = \int_{e_1} \mathbf{B}^{\mathrm{T}} EA\mathbf{B}\, \mathrm{d}x \quad \mathbf{K}^{e2} = \int_{e_2} \mathbf{B}^{\mathrm{T}} EA\mathbf{B}\, \mathrm{d}x \quad \mathbf{K}^{e3} = \int_{e_3} \mathbf{B}^{\mathrm{T}} EA\mathbf{B}\, \mathrm{d}x \quad \mathbf{K}^{e4} = \int_{e_4} \mathbf{B}^{\mathrm{T}} EA\mathbf{B}\, \mathrm{d}x$$

Global approach (shape function-centered)



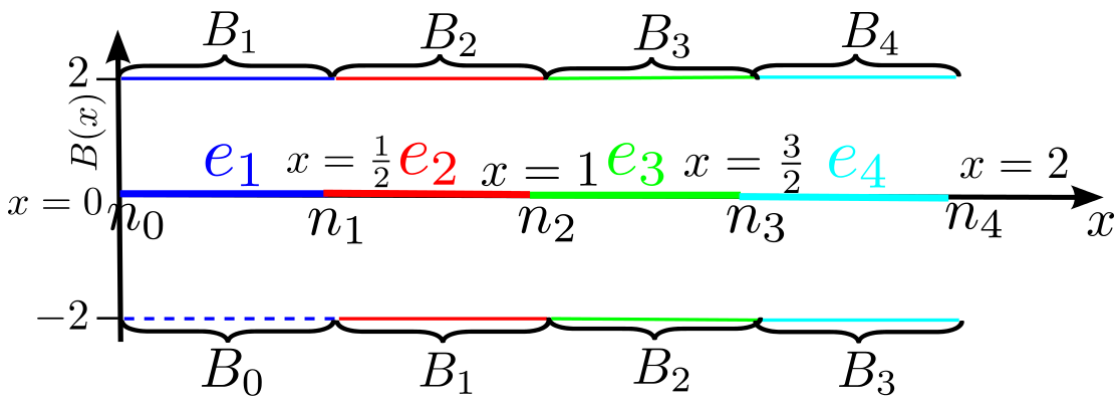
$$\mathbf{K} = \int_0^2 \begin{bmatrix} B_1 \\ B_2 \\ B_3 \\ B_4 \end{bmatrix} EA \begin{bmatrix} B_1 & B_2 & B_3 & B_4 \end{bmatrix} \mathrm{d}x = \begin{bmatrix} \int_0^2 B_1 B_1 \,\mathrm{d}x & \int_0^2 B_1 B_2 \,\mathrm{d}x & \int_0^2 B_1 B_3 \,\mathrm{d}x & \int_0^2 B_1 B_4 \,\mathrm{d}x \\ & \int_0^2 B_2 B_2 \,\mathrm{d}x & \int_0^2 B_2 B_3 \,\mathrm{d}x & \int_0^2 B_2 B_4 \,\mathrm{d}x \\ & & \int_0^2 B_3 B_3 \,\mathrm{d}x & \int_0^2 B_3 B_4 \,\mathrm{d}x \\ \text{sym.} & & & \int_0^2 B_4 B_4 \,\mathrm{d}x \end{bmatrix}$$

$$= \begin{bmatrix} \int_{e_1} B_1 B_1 \,\mathrm{d}x + \int_{e_2} B_1 B_1 \,\mathrm{d}x & \int_{e_2} B_1 B_2 \,\mathrm{d}x & 0 & 0 \\ & \int_{e_2} B_2 B_2 \,\mathrm{d}x + \int_{e_3} B_2 B_2 \,\mathrm{d}x & \int_{e_3} B_2 B_3 \,\mathrm{d}x & 0 \\ & & \int_{e_3} B_3 B_3 \,\mathrm{d}x + \int_{e_4} B_3 B_3 \,\mathrm{d}x & \int_{e_4} B_3 B_4 \,\mathrm{d}x \\ \text{sym.} & & & \int_{e_4} B_4 B_4 \,\mathrm{d}x \end{bmatrix}$$

$$= \begin{bmatrix} \frac{1}{2}.(2).(2) + \frac{1}{2}.(-2).(-2) & \frac{1}{2}.(-2).(2) & 0 & 0 \\ & \frac{1}{2}.(2).(2) + \frac{1}{2}.(-2).(-2) & \frac{1}{2}.(-2).(2) & 0 \\ & & \frac{1}{2}.(2).(2) + \frac{1}{2}.(-2).(-2) & \frac{1}{2}.(-2).(2) \\ \text{sym.} & & & \frac{1}{2}.(2).(2) \end{bmatrix} \Rightarrow$$

$$\mathbf{K} = \begin{bmatrix} 4 & -2 & 0 & 0 \\ & 4 & -2 & 0 \\ & & 4 & -2 \\ \text{sym.} & & & 2 \end{bmatrix}$$

Local approach (element-centered)



$$\mathbf{K} = \int_0^2 \mathbf{B}^T E A \mathbf{B}\, \mathrm{d}x = \overbrace{\int_{e_1} \mathbf{B}^{\mathrm{T}} E A \mathbf{B}\, \mathrm{d}x}^{\mathbf{K}^{e_1}} + \overbrace{\int_{e_2} \mathbf{B}^{\mathrm{T}} E A \mathbf{B}\, \mathrm{d}x}^{\mathbf{K}^{e_2}} + \overbrace{\int_{e_3} \mathbf{B}^{\mathrm{T}} E A \mathbf{B}\, \mathrm{d}x}^{\mathbf{K}^{e_3}} + \overbrace{\int_{e_4} \mathbf{B}^{\mathrm{T}} E A \mathbf{B}\, \mathrm{d}x}^{\mathbf{K}^{e_4}} = \tag{327}$$

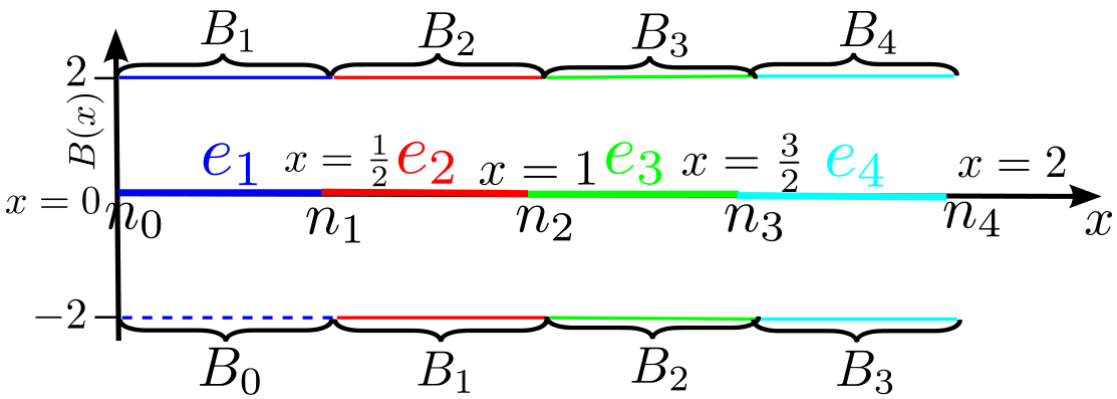
$$= \begin{bmatrix} \int_{e_1} B_1 B_1\, \mathrm{d}x & 0 & 0 & 0 \\ & 0 & 0 & 0 \\ \text{sym.} & & 0 & 0 \\ & & & 0 \end{bmatrix} + \begin{bmatrix} \int_{e_2} B_1 B_1\, \mathrm{d}x & \int_{e_2} B_1 B_2\, \mathrm{d}x & 0 & 0 \\ & \int_{e_2} B_2 B_2\, \mathrm{d}x & 0 & 0 \\ \text{sym.} & & 0 & 0 \\ & & & 0 \end{bmatrix}$$

$$+ \begin{bmatrix} 0 & 0 & 0 & 0 \\ 0 & \int_{e_3} B_2 B_2\, \mathrm{d}x & \int_{e_3} B_2 B_3\, \mathrm{d}x & 0 \\ \text{sym.} & & \int_{e_3} B_3 B_3\, \mathrm{d}x & 0 \\ & & & 0 \end{bmatrix} + \begin{bmatrix} 0 & 0 & 0 & 0 \\ & 0 & 0 & 0 \\ \text{sym.} & & \int_{e_4} B_3 B_3\, \mathrm{d}x & \int_{e_4} B_3 B_4\, \mathrm{d}x \\ & & & \int_{e_4} B_4 B_4\, \mathrm{d}x \end{bmatrix}$$

$$= \begin{bmatrix} \frac{1}{2}.(2).(2) & 0 & 0 & 0 \\ & 0 & 0 & 0 \\ \text{sym.} & & 0 & 0 \\ & & & 0 \end{bmatrix} + \begin{bmatrix} \frac{1}{2}.(-2).(-2) & \frac{1}{2}.(-2).(2) & 0 & 0 \\ & \frac{1}{2}.(2).(2) & 0 & 0 \\ \text{sym.} & & 0 & 0 \\ & & & 0 \end{bmatrix}$$

$$+ \begin{bmatrix} 0 & 0 & 0 & 0 \\ 0 & \frac{1}{2}.(-2).(-2) & \frac{1}{2}.(-2).(2) & 0 \\ \text{sym.} & & \frac{1}{2}.(2).(2) & 0 \\ & & & 0 \end{bmatrix} + \begin{bmatrix} 0 & 0 & 0 & 0 \\ & 0 & 0 & 0 \\ \text{sym.} & & \frac{1}{2}.(-2).(-2) & \frac{1}{2}.(-2).(2) \\ & & & \frac{1}{2}.(2).(2) \end{bmatrix} \quad \Rightarrow$$

Local approach (element-centered)



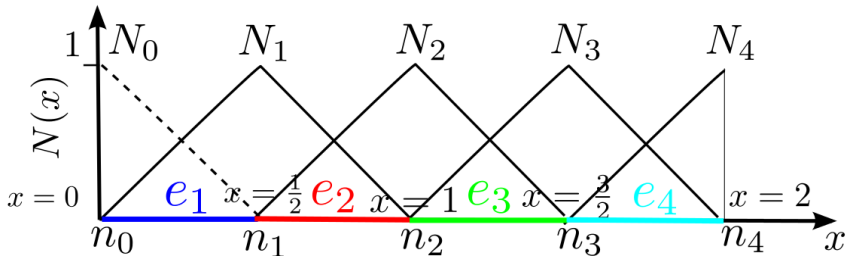
$$\mathbf{K}^{e_1} = \begin{bmatrix} 2 & 0 & 0 & 0 \\ & 0 & 0 & 0 \\ \text{sym.} & & 0 & 0 \\ & & & 0 \end{bmatrix} \qquad \mathbf{K}^{e_2} = \begin{bmatrix} 2 & -2 & 0 & 0 \\ & 2 & 0 & 0 \\ \text{sym.} & & 0 & 0 \\ & & & 0 \end{bmatrix}$$

$$\mathbf{K}^{e_3} = \begin{bmatrix} 0 & 0 & 0 & 0 \\ 0 & 2 & -2 & 0 \\ \text{sym.} & & 2 & 0 \\ & & & 0 \end{bmatrix} \qquad \mathbf{K}^{e_4} = \begin{bmatrix} 0 & 0 & 0 & 0 \\ & 0 & 0 & 0 \\ \text{sym.} & & 2 & -2 \\ & & & 2 \end{bmatrix}$$

$$\mathbf{K} = \mathbf{K}^{e_1} + \mathbf{K}^{e_2} + \mathbf{K}^{e_3} + \mathbf{K}^{e_4} \quad \Rightarrow$$

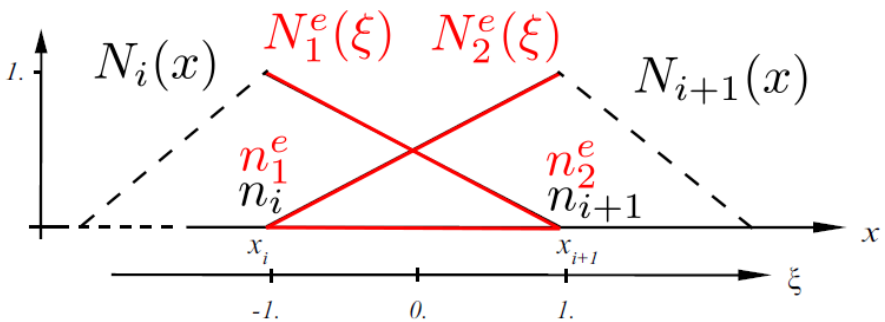
$$\mathbf{K} = \begin{bmatrix} 4 & -2 & 0 & 0 \\ & 4 & -2 & 0 \\ \text{sym.} & & 4 & -2 \\ & & & 2 \end{bmatrix}$$

Summary: Formation of $\mathbf{K}$ and $\mathbf{F}$



The formation of $\mathbf{K}$ and $\mathbf{F}$ includes two steps:

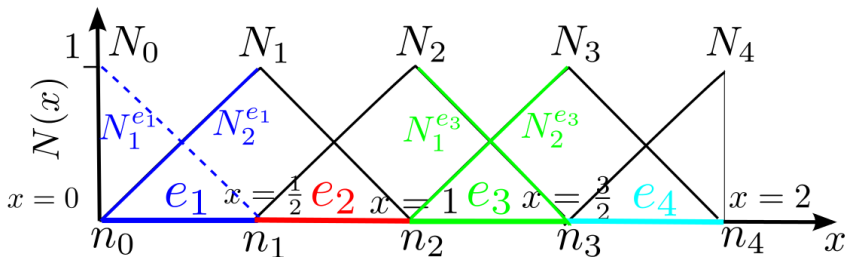
1. Formation of element (local) stiffness matrix and force vectors: Element level matrix $\mathbf{k}^e$ is $n^e_{\text{dof}} \times n^e_{\text{dof}}$ and forces are $n^e_{\text{dof}} \times 1$. As we will shortly see the element stiffness matrix for the bar element shown in the figure (constant A and E) is:



$$\mathbf{k}^e = \frac{AE}{L}\begin{bmatrix} 1 & -1 \\ -1 & 1 \end{bmatrix}, \quad L = x_{i+1} - x_i \tag{328}$$

2. Summation to Global system: Components of local stiffness and force vectors are directly added to Global system.

Example: Formation of $\mathbf{K}$ and $\mathbf{F}$



1. Local stiffness matrix: Since $A = 1, E = 1, L = \frac{1}{2}$ for all elements in the example shown, local stiffness matrix is:

$$\mathbf{k}^{e_i} = \begin{bmatrix} 2 & -2 \\ -2 & 2 \end{bmatrix}, i = 1, 2, 3, 4.$$

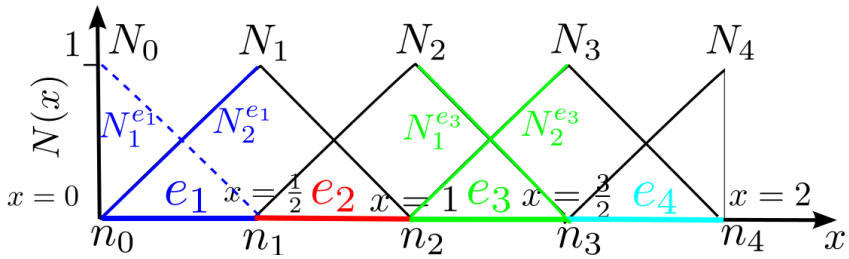
2. Assembly to global system: Around the local stiffness matrix for e_3 we have the corresponding dof of the local dof in the global system:

$$\begin{array}{c} \\ 2 \\ 3 \end{array} \begin{array}{c} \begin{array}{cc} 2 & 3 \end{array} \\ \begin{bmatrix} 2 & -2 \\ -2 & 2 \end{bmatrix} \end{array}$$

This means that for example $k_{11}^{e_3}$ will be added to K_{22}, $k_{12}^{e_3}$ to K_{23} and so forth:

$$\mathbf{K}^{e_3} = \begin{bmatrix} 0 & 0 & 0 & 0 \\ 0 & 2 & -2 & 0 \\ \text{sym.} & & 2 & 0 \\ & & & 0 \end{bmatrix}$$

Example: Formation of $\mathbf{K}$ and $\mathbf{F}$



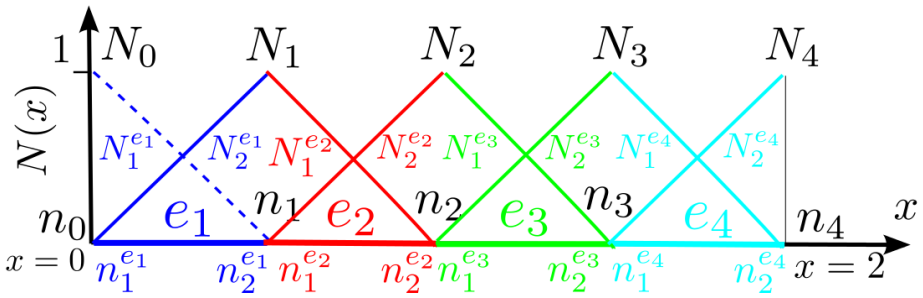
Similarly for element e_1 the local stiffness matrix $\mathbf{k}^{e_1}$ and corresponding **free** global dof are:

	0	1
0	2	-2
1	-2	2

Note that 0 in the first row and column means that dof 1 in the element does not correspond to any free global dofs (n_0 is a prescribed node on $\partial\mathcal{D}_u$). In This case, only $k_{22}^{e1} = 2$ is added to K_{11} from e_1. The corresponding global level matrix from element e_1 is:

$$\mathbf{K}^{e_1} = \begin{bmatrix} 2 & 0 & 0 & 0 \\ & 0 & 0 & 0 \\ \text{sym.} & & 0 & 0 \\ & & & 0 \end{bmatrix}$$

Local to Global dof transfer



We demonstrated the transformation concept using our solid bar example.

- First we denote that for this problem the sets for **free** and **prescribed** dofs are:

$$S_{\mathrm{f}} = \{u(n_1), u(n_2), u(n_3), u(n_4)\} \quad \Rightarrow \quad n_{\mathrm{f}} = 4$$
$$S_{\mathrm{p}} = \{u(n_0)\} \quad \Rightarrow \quad n_{\mathrm{p}} = 1$$

- The corresponding shape function matrices are:

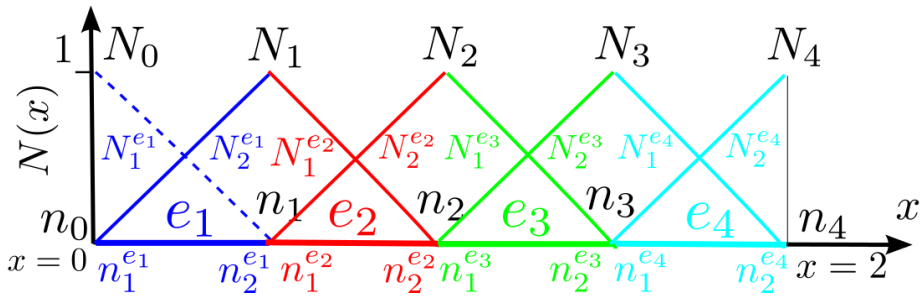
$$\mathbf{N} = [N_1, N_2, N_3, N_4]$$
$$\bar{\mathbf{N}} = [\bar{N}_{\bar{0}}]$$

- At the elements we define element shape function matrices as,

$$\mathbf{N}^e = [N_1^e, N_2^e] \tag{329}$$

- Our goal is to **relate element level shape function $\mathbf{N}^e$ to global $\mathbf{N}$ and $\bar{\mathbf{N}}$**

Local to Global dof transfer



- The **local to global free dof map**, $\mathbf{M}^e$ $(1 \times n^e_{\text{dof}})$ is defined such that,

$$\begin{cases} N_{M^e(i)} = N_i^e \text{ inside } e_i & \text{dof } i \text{ is free} \\ M^e(i) = 0 & \text{dof } i \text{ is fixed} \end{cases} \tag{330}$$

- Similarly we define **local to global fixed dof map** $\bar{\mathbf{M}}^e$ $(1 \times n^e_{\text{dof}})$ as,

$$\begin{cases} M^e(i) = 0 & \text{dof } i \text{ is free} \\ \bar{N}_{\bar{M}^e(i)} = N_i^e \text{ inside } e_i & \text{dof } i \text{ is fixed} \end{cases} \tag{331}$$

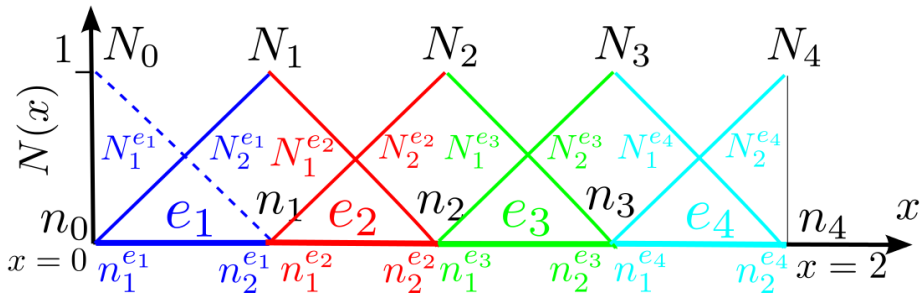
- For example for our sample problem, these maps are,

$$\mathbf{M}^{e_1} = [0\ 1]^{\mathrm{T}} \qquad \bar{\mathbf{M}}^{e_1} = [1\ 0]^{\mathrm{T}}$$
$$\mathbf{M}^{e_2} = [1\ 2]^{\mathrm{T}} \qquad \bar{\mathbf{M}}^{e_2} = [0\ 0]^{\mathrm{T}}$$
$$\mathbf{M}^{e_3} = [2\ 3]^{\mathrm{T}} \qquad \bar{\mathbf{M}}^{e_3} = [0\ 0]^{\mathrm{T}}$$
$$\mathbf{M}^{e_4} = [3\ 4]^{\mathrm{T}} \qquad \bar{\mathbf{M}}^{e_4} = [0\ 0]^{\mathrm{T}}$$

- We also combine both maps into $\mathbf{M}^e_t$ that includes both free and prescribed dofs:

$$\mathbf{M}^{e_1}{}_t = [\bar{1}\ 1]^{\mathrm{T}} \qquad \mathbf{M}^{e_2}{}_t = [1\ 2]^{\mathrm{T}} \qquad \mathbf{M}^{e_3}{}_t = [2\ 3]^{\mathrm{T}} \qquad \mathbf{M}^{e_4}{}_t = [3\ 4]^{\mathrm{T}}$$

Local to Global dof. transfer: Boolean Matrices

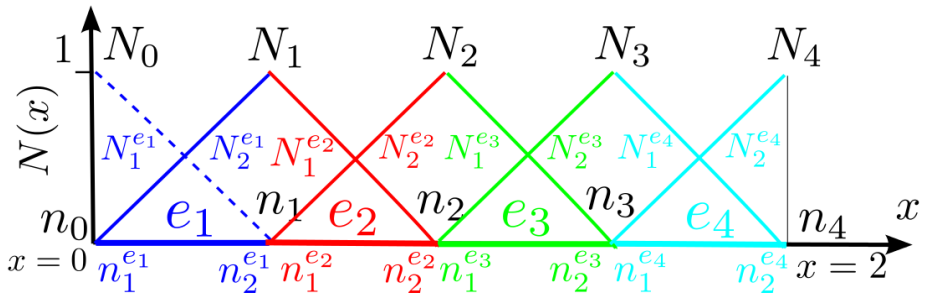


- While $\mathbf{M}^e$ and $\bar{\mathbf{M}}^e$ (or their variants) are convenient for computer coding, the **Boolean Matrices** $\mathbf{C}^e$ and $\bar{\mathbf{C}}^e$ better illustrate the formation of local stiffness and force vectors and their transfer to global system:
- $\mathbf{C}^e$ is defined such that

$$\mathbf{N} = \mathbf{N}^e \mathbf{C}^e \text{ inside element } e \quad \Rightarrow \tag{332}$$

$$[N_1, \ldots, N_{n_\mathrm{f}}] = [N_1^e, \ldots, N_{n^e_\mathrm{dof}}] \begin{bmatrix} C^e_{11} & C^e_{12} & \cdots & C^e_{1n_\mathrm{f}} \\ C^e_{21} & C^e_{22} & \cdots & C^e_{2n_\mathrm{f}} \\ \vdots & \cdots & \ddots & \vdots \\ C^e_{n^e_\mathrm{dof}1} & C^e_{n^e_\mathrm{dof}2} & \cdots & C^e_{n^e_\mathrm{dof}n_\mathrm{f}} \end{bmatrix} \text{ inside } e$$

Local to Global dof. transfer: Boolean Matrices



- $\mathbf{C}^e$ is an $n^e_{\text{dof}} \times n_{\text{f}}$ matrix, filled with zeros and and a maximum of one 1 in each row.
- For example for our sample example and elements e_1 and e_3 we have,

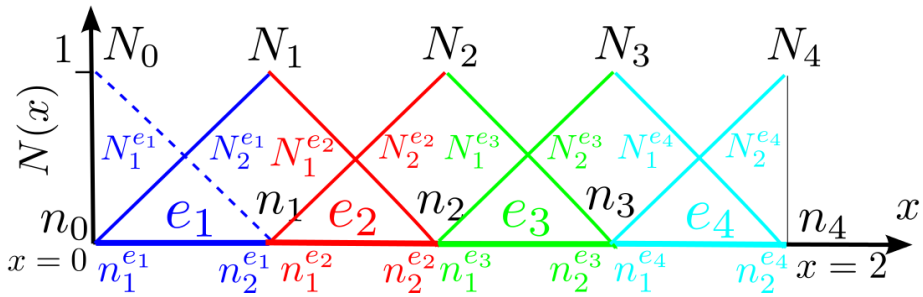
$$\mathbf{C}^{e_1} = \begin{bmatrix} 0 & 0 & 0 & 0 \\ 1 & 0 & 0 & 0 \end{bmatrix} \quad \mathbf{C}^{e_3} = \begin{bmatrix} 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \end{bmatrix}$$

- The relation between $\mathbf{C}^e$ and $\mathbf{M}^e$ is,

$$\text{row } i \text{ of } \mathbf{C}^e = \begin{cases} \mathbf{0} & M^e(i) = 0 \text{ (prescribed dof.)} \\ \mathbf{e}_{M^e(i)} & \mathbf{M}^{e_i} \neq 0 \text{ (free dof.)} \end{cases} \tag{333}$$

  where $\mathbf{e}_k = [0, \ldots, \overbrace{1}^{k^{\text{th}} \text{ position}}, \ldots, 0]$ is the kth unit vector.
- That is, $\mathbf{C}^e$ is a zero matrix except taking the value of one at $C^e_{i,M^e(i)}$ for $M^e(i) > 0$ (free dof.).

Local to Global dof. transfer: Boolean Matrices

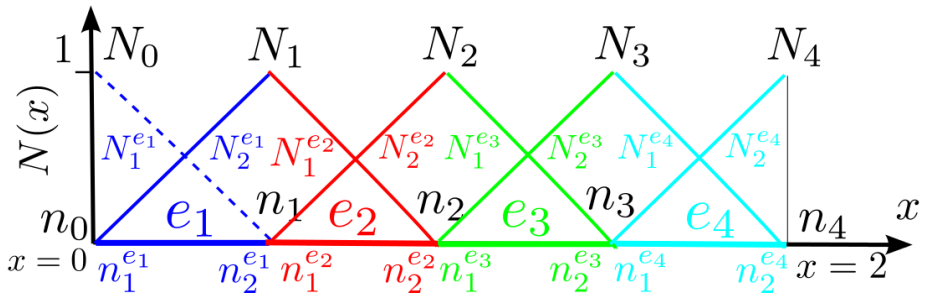


- Instead of free dofs we obtain the transfer between global prescribed dof. and element dofs within an element.
- $\bar{\mathbf{C}}^e$ is defined such that

$$\bar{\mathbf{N}} = \mathbf{N}^e \bar{\mathbf{C}}^e \text{ inside element } e \quad \Rightarrow \tag{334}$$

$$[\bar{N}_{\bar{1}}, \ldots, \bar{N}_{\bar{n_{\mathrm{p}}}}] = [N_1^e, \ldots, N_{n_{\mathrm{dof}}^e}^e] \begin{bmatrix} \bar{C}_{11}^e & \bar{C}_{12}^e & \cdots & \bar{C}_{1n_{\mathrm{p}}}^e \\ \bar{C}_{21}^e & \bar{C}_{22}^e & \cdots & \bar{C}_{2n_{\mathrm{p}}}^e \\ \vdots & \cdots & \ddots & \vdots \\ \bar{C}_{n_{\mathrm{dof}}^e 1}^e & \bar{C}_{n_{\mathrm{dof}}^e 2}^e & \cdots & \bar{C}_{n_{\mathrm{dof}}^e n_{\mathrm{p}}}^e \end{bmatrix} \text{ inside } e$$

Local to Global dof. transfer: Boolean Matrices



- $\bar{\mathbf{C}}^e$ is an $n^e_{\text{dof}} \times n_{\text{p}}$ matrix, filled with zeros and and a maximum of one 1 in each row.
- For example for our sample example and elements e_1 and e_3 we have,

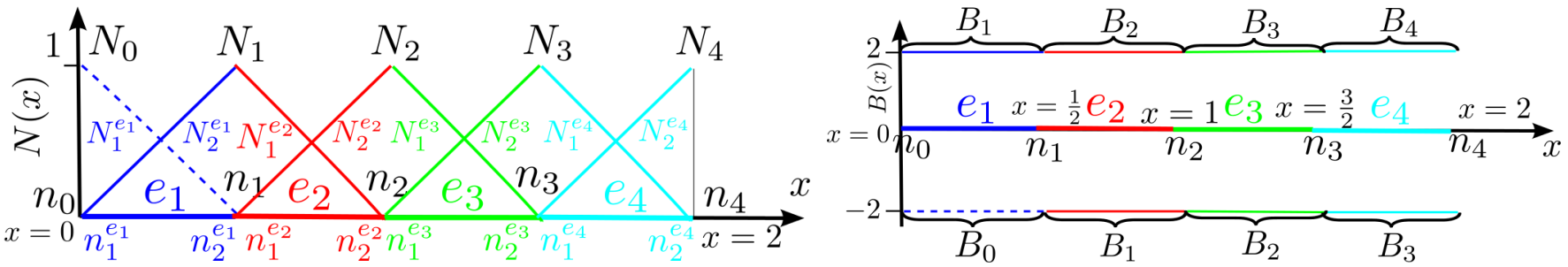
$$\bar{\mathbf{C}}^{e_1} = \begin{bmatrix} 1 \\ 0 \end{bmatrix} \quad \bar{\mathbf{C}}^{e_3} = \begin{bmatrix} 0 \\ 0 \end{bmatrix}$$

- The relation between $\bar{\mathbf{C}}^e$ and $\bar{\mathbf{M}}^e$ is,

$$\text{row } i \text{ of } \bar{\mathbf{C}}^e = \begin{cases} \mathbf{e}_{\bar{M}^e(i)} & \bar{\mathbf{M}}^{e_i} \neq 0 \text{ (prescribed dof.)} \\ \mathbf{0} & \bar{M}^e(i) = 0 \text{ (free dof.)} \end{cases} \tag{335}$$

  as before $\mathbf{e}_k = [0, \ldots, \overbrace{1}^{k^{\text{th}} \text{ position}}, \ldots, 0]$ is the kth unit vector.
- That is, $\bar{\mathbf{C}}^e$ is a zero matrix except taking the value of one at $\bar{C}^e_{i,\bar{M}^e(i)}$ for $\bar{M}^e(i) > 0$ (prescribed dof.).

Assembly of element stiffness and force to global system



- For our bar example we observed (*cf.* (326))

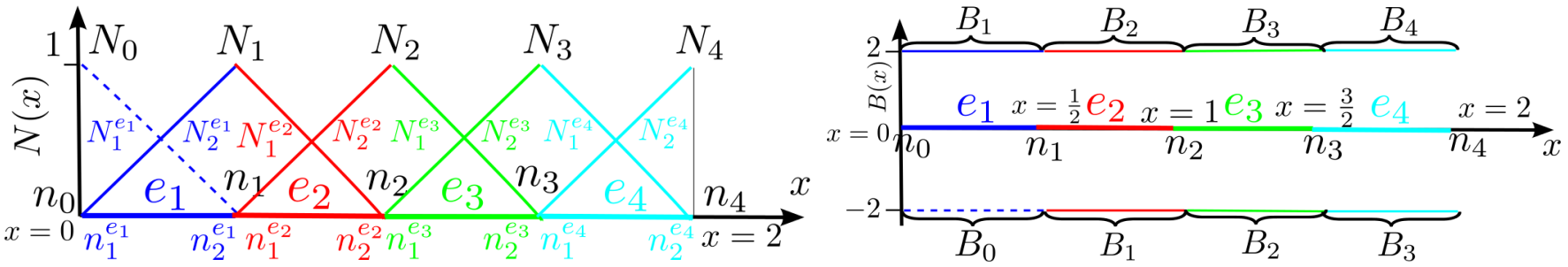
$$\mathbf{K} = \int_0^2 \mathbf{B}^{\mathrm{T}} EA\mathbf{B}\,\mathrm{d}x = \sum_{I=1}^{4} \int_{e_I} \mathbf{B}^{\mathrm{T}} EA\mathbf{B}\,\mathrm{d}x$$

  - We **break the integration over the domain to integration over elements**
  - We define element level stiffness matrices,

$$\mathbf{K}^{e_I} = \int_{e_I} \mathbf{B}^{\mathrm{T}} EA\mathbf{B}\,\mathrm{d}x \quad \Rightarrow \quad \mathbf{K} = \sum_{I=1}^{4} \mathbf{K}^{e_I}$$

- While, by $\mathbf{K} = \sum_{I=1}^{4} \mathbf{K}^{e_I}$ we have localized the computation of stiffness matrix to elements, this process is still not complete because in the computation of $\mathbf{K}^{e_I}$ we still are working with $\mathbf{B}$ which is a global quantity.

Assembly of element stiffness and force to global system



- Our goal is to come up with completely local element-level calculations. The benefits are,
  1. Element-level calculations are completely independent from global level. Element level calculations can be done as a "black box".
  2. The "assembly" of local to global matrices and vectors become a general routine independent of element type.
- Total separation of local and global quantities is feasible by using **boolean matrices**:

$$\mathbf{N} = \mathbf{N}^e\mathbf{C}^e \text{ inside element } e \quad \Rightarrow \mathbf{B} = \frac{\mathrm{d}\mathbf{N}}{\mathrm{d}x} = \frac{\mathrm{d}\mathbf{N}^e\mathbf{C}^e}{\mathrm{d}x} = \mathbf{B}^e\mathbf{C}^e, \quad \text{where } \mathbf{B}^e = [B_1^e \; B_2^e]$$

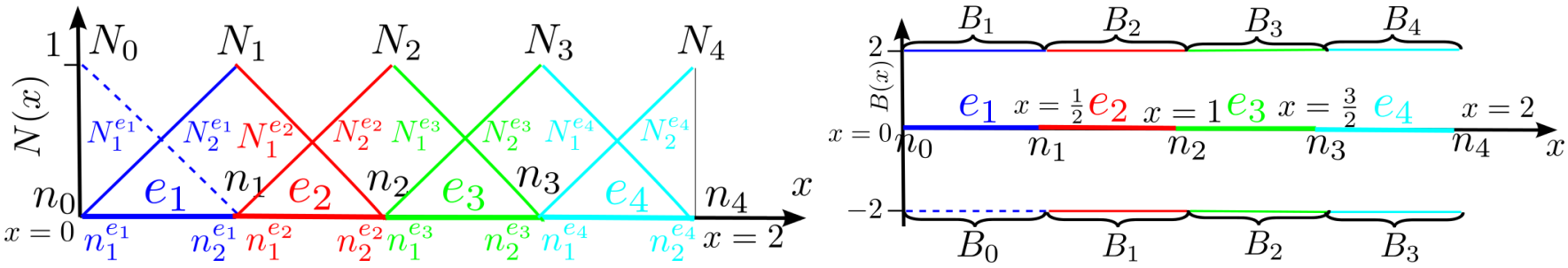
for the bar element (2 dof. per element, $n_{\text{dof}}^e = 2$)

- The local stiffness matrices can be further simplified:

$$\mathbf{K}^e = \int_e \mathbf{B}^{\mathrm{T}} EA\mathbf{B}\,\mathrm{d}x = \int_e (\mathbf{B}^e\mathbf{C}^e)^{\mathrm{T}} EA(\mathbf{B}^e\mathbf{C}^e)\,\mathrm{d}x = \mathbf{C}^{e\,\mathrm{T}} \left\{ \int_e \mathbf{B}^{e\,\mathrm{T}} EA\mathbf{B}^e\,\mathrm{d}x \right\} \mathbf{C}^e$$

the substitution $\mathbf{B} = \mathbf{B}^e\mathbf{C}^e$ is only valid within element e, but the integration is also inside element e.

Assembly of element stiffness and force to global system

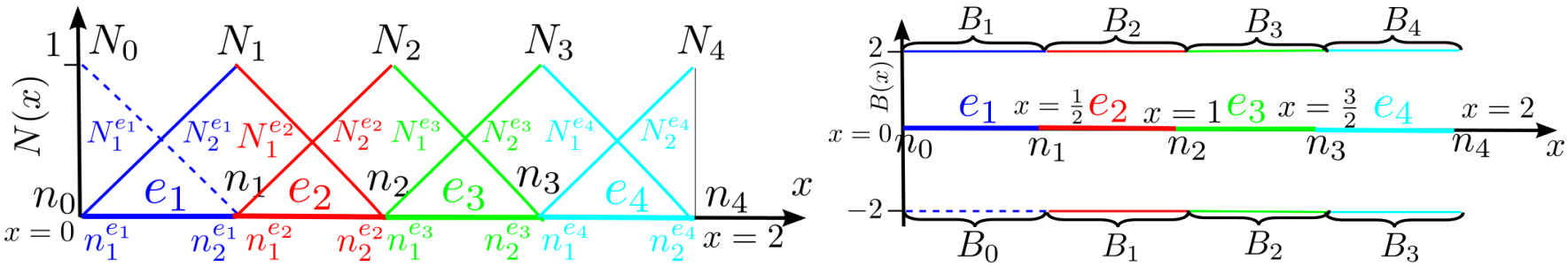


- We define local element matrices $\mathbf{k}^e$:

$$\mathbf{k}^e = \int_e \mathbf{B}^{e\mathrm{T}} E A \mathbf{B}^e \,\mathrm{d}x \quad \Rightarrow \boxed{\mathbf{K}^e = \mathbf{C}^{e\mathrm{T}} \mathbf{k}^e \mathbf{C}^e}$$

- Stiffness matrices $\mathbf{k}^e$ are completely local:
  1. The integration domain is the element e.
  2. The matrices $\mathbf{B}^e$ are local and completely independent of global dof. numbering.
  3. $\mathbf{k}^e$ is a $n^e_{\mathrm{dof}} \times n^e_{\mathrm{dof}}$ matrix as opposed to $n_{\mathrm{dof}} \times n_{\mathrm{dof}}$ for $\mathbf{K}^e$.
- $\mathbf{C}^e$ in the left and right side of $\mathbf{k}^e = \mathbf{C}^{e\mathrm{T}} \mathbf{k}^e \mathbf{C}^e$ only juxtapose components of local matrices into correct rows and columns of $\mathbf{K}^e$, respectively.

Assembly of element stiffness and force to global system



- In summary for our sample bar problem:

$$\mathbf{K} = \sum_{I=1}^{4} \mathbf{K}^{e_I} = \sum_{I=1}^{4} \mathbf{C}^{e_I\,\mathrm{T}} \mathbf{k}^{e_I} \mathbf{C}^{e_I}$$

- We do not pre- and pos-multiply by $\mathbf{C}^e$ matrices; rather, we use mapping $\mathbf{M}^e$ vectors to achieve the same result.
- In practice, we do not form $\mathbf{K}^e$ and directly "assemble" $\mathbf{k}^e$ into $\mathbf{K}$ using the maps between local and global dofs.
- We will subsequently do the same process for **source term force**, **natural BC force**, and **essential BC force**. In each case the element level force vector is pre-multiplied by $\mathbf{C}^{e\,\mathrm{T}}$. That, is we put the components of element level force vectors in the "right" components of the global force vector.

Weak statement for finite element formulation

- Let the weak statement for a self-adjoint problem be of the form (why the problem is self-adjoint? Change $\mathbf{u}$ and $\mathbf{w}$ on the LHS):

$$\int_{\mathcal{D}} L_m(\mathbf{w})\mathbf{D}L_m(\mathbf{u})\,\mathrm{dv} = \int_{\mathcal{D}} \mathbf{w}\mathbf{r}\,\mathrm{dv} + \int_{\partial\mathcal{D}_f} \mathbf{w}\bar{\mathbf{F}}.\mathbf{N}\,\mathrm{ds} \tag{336}$$

  for example for solid bar we have:

$$\int_0^L \frac{\mathrm{d}w}{\mathrm{d}x} EA \frac{\mathrm{d}u}{\mathrm{d}x}\,\mathrm{d}x = \int_0^L wq\,\mathrm{d}x + (w.\bar{F})_{\partial\mathcal{D}_f}$$

  where $\mathcal{D} = [0\ L]$, $\mathbf{D} = EA$, $L_m = \frac{\mathrm{d}}{\mathrm{d}x}$ and $\partial\mathcal{D}_f$ is either $\{0\}$ or $\{L\}$ (since at least one of these points should be essential BC to prevent rigid body motion for statics, not both points can be in $\partial\mathcal{D}_f$ simultaneously).
- Recalling our general definitions from (288):

$$\mathcal{A}(\mathbf{w},\mathbf{u}) := \int_{\mathcal{D}} L_m(\mathbf{w})\mathbf{D}L_m(\mathbf{u})\,\mathrm{dv} \qquad \text{bilinear form} \tag{337a}$$

$$(\mathbf{w},\mathbf{r})_r := \int_{\mathcal{D}} \mathbf{w}.\mathbf{r}\,\mathrm{dv} \qquad \text{linear force from source terms} \tag{337b}$$

$$(\mathbf{w},\bar{\mathbf{F}})_N := \int_{\partial\mathcal{D}_f} \mathbf{w}\bar{\mathbf{F}}.\mathbf{N}\,\mathrm{ds} \qquad \text{linear force from natural BC} \tag{337c}$$

Weak statement for finite element formulation

- The weak statement can be rewritten as,

$$\mathcal{A}\left(\mathbf{w}, \mathbf{u}\right) = \left(\mathbf{w}, \mathbf{r}\right)_r + \left(\mathbf{w}, \bar{\mathbf{F}}\right)_N \tag{338}$$

- Let us discretize this problem to n_{dof} number of dofs, from which n_{f} are free (unknown) and n_{p} are prescribed (known). The corresponding shape vectors are:

$$\mathbf{N} = [\mathbf{N}_1, \ldots, \mathbf{N}_{n_{\mathrm{f}}}] \tag{339a}$$
$$\bar{\mathbf{N}} = [\bar{\mathbf{N}}_{\bar{1}}, \ldots, \bar{\mathbf{N}}_{\bar{n_{\mathrm{p}}}}] \tag{339b}$$

- Similar to solid bar where we defined the so-called kinematic (displacement to strain) matrix $\mathbf{B} = L_m(\mathbf{N}) = \frac{\mathrm{d}\mathbf{N}}{\mathrm{d}x}$, we define,

$$\mathbf{B} = L_m(\mathbf{N}) = [L_m(\mathbf{N}_1), \ldots, L_m(\mathbf{N}_{n_{\mathrm{f}}})] = [\mathbf{B}_1, \ldots, \mathbf{B}_{n_{\mathrm{f}}}] \tag{340a}$$
$$\bar{\mathbf{B}} = L_m(\bar{\mathbf{N}}) = [L_m(\bar{\mathbf{N}}_{\bar{1}}), \ldots, L_m(\bar{\mathbf{N}}_{\bar{n_{\mathrm{p}}}})] = [\bar{\mathbf{B}}_{\bar{1}}, \ldots, \bar{\mathbf{B}}_{\bar{n_{\mathrm{p}}}}] \tag{340b}$$

Weak statement for finite element formulation

- Similar to our development for solid bar, stiffness matrix can be expressed as,

$$\mathbf{K} = \int_D \mathbf{B}^{\mathrm{T}} \mathbf{D} \mathbf{B} \, \mathrm{dv} \tag{341}$$

- Recalling (312) forces are given by:

$$\mathbf{F}_r = \left(\mathbf{N}^{\mathrm{T}}, r \right)_r = \int_{\mathcal{D}} \mathbf{N}^{\mathrm{T}} r \, \mathrm{dv} \quad \text{source term force} \tag{342a}$$

$$\mathbf{F}_N = \left(\mathbf{N}^{\mathrm{T}}, \bar{\mathbf{F}} \right)_N = \int_{\partial \mathcal{D}_f} \mathbf{N}^{\mathrm{T}} \bar{\mathbf{F}} . \mathbf{N} \, \mathrm{ds} \quad \text{essential BC force} \tag{342b}$$

$$\mathbf{F}_D = \mathcal{A} \left(\mathbf{N}^{\mathrm{T}}, \phi_p \right) = \int_{\mathcal{D}} L_m(\mathbf{N})^{\mathrm{T}} \mathbf{D} L_m(\phi_p) \, \mathrm{dv} = \left\{ \int_{\mathcal{D}} \mathbf{B}^{\mathrm{T}} \mathbf{D} \bar{\mathbf{B}} \, \mathrm{dv} \right\} \bar{\mathbf{a}}$$

$$= \mathbf{K}_{fp} \bar{\mathbf{a}} \quad \text{essential BC force} \tag{342c}$$

$$\mathbf{F}_n = \begin{bmatrix} F_{n1} \\ \vdots \\ F_{n n_{\mathrm{f}}} \end{bmatrix} \quad \text{nodal forces} \tag{342d}$$

and the total force is given by (*cf.* (311)):

$$\mathbf{F} = \mathbf{F}_r + \mathbf{F}_N + \mathbf{F}_n - \mathbf{F}_D \tag{343}$$

- Next, we related element-level stiffness and force to global ones.

Local to Global: Stiffness matrix

We derive relations between local and global stiffness matrix and force vectors:

1. Stiffness matrix We observed from (341):

$$\mathbf{K} = \int_{\mathcal{D}} \mathbf{B}^{\mathrm{T}} \mathbf{D} \mathbf{B} \, \mathrm{dv}$$

$$\mathcal{D} = \cup_{I=1}^{n_{\mathrm{e}}} e_I = \text{ Discretization domain}$$

- First, we break domain of integration using $\mathcal{D} = \cup_{I=1}^{n_{\mathrm{e}}} e_I$ (disjoint union):

$$\mathbf{K} = \sum_{I=1}^{n_{\mathrm{e}}} \mathbf{K}^{e_I} \tag{344}$$

$$\mathbf{K}^{e_I} = \int_{e_I} \mathbf{B}^{\mathrm{T}} \mathbf{D} \mathbf{B} \, \mathrm{dv}$$

$$n_{\mathrm{e}} = \text{ Number of elements}$$

- Second, having $\mathbf{N} = \mathbf{N}^e \mathbf{C}^e \Rightarrow \mathbf{B} = L_m(\mathbf{N}) = L_m(\mathbf{N}^e \mathbf{C}^e) = \mathbf{B}^e \mathbf{C}^e$ within element e we have:

$$\mathbf{K}^e = \int_e \mathbf{B}^{\mathrm{T}} \mathbf{D} \mathbf{B} \, \mathrm{dv} = \int_e (\mathbf{B}^e \mathbf{C}^e)^{\mathrm{T}} \mathbf{D} (\mathbf{B}^e \mathbf{C}^e) \, \mathrm{dv} \Rightarrow \mathbf{K}^e = \mathbf{C}^{e\,\mathrm{T}} \mathbf{k}^e \mathbf{C}^e, \quad \Rightarrow \tag{345}$$

$$\mathbf{K} = \sum_{I=1}^{n_{\mathrm{e}}} \mathbf{K}^{e_I} = \sum_{I=1}^{n_{\mathrm{e}}} \mathbf{C}^{e_I\,\mathrm{T}} \mathbf{k}^{e_I} \mathbf{C}^{e_I} \quad \text{for} \tag{346a}$$

$$\mathbf{k}^e = \int_e \mathbf{B}^{e\,\mathrm{T}} \mathbf{D} \mathbf{B}^e \, \mathrm{dv} \tag{346b}$$

Local to Global: Source term force vector

2. Source term force vector We observed from (342a):

$$\mathbf{F}_r = \left(\mathbf{N}^{\mathrm{T}}, \mathbf{r}\right)_r = \int_{\mathcal{D}} \mathbf{N}^{\mathrm{T}}.\mathbf{r}\,\mathrm{dv}$$

- As stiffness matrix, we first break domain of integration using $\mathcal{D} = \cup_{I=1}^{n_{\mathrm{e}}} e_I$ (disjoint union):

$$\mathbf{F}_r = \sum_{I=1}^{n_{\mathrm{e}}} \mathbf{F}_r^{e_I} \quad \text{where} \tag{347}$$

$$\mathbf{F}_r^{e_I} = \int_{e_I} \mathbf{N}^{\mathrm{T}}\mathbf{r}\,\mathrm{dv} \tag{348}$$

- Second, having $\mathbf{N} = \mathbf{N}^e\mathbf{C}^e$ within element e we have:

$$\mathbf{F}_r^e = \int_e \mathbf{N}^{\mathrm{T}}\mathbf{r}\,\mathrm{dv} = \int_e (\mathbf{N}^e\mathbf{C}^e)^{\mathrm{T}}\mathbf{r}\,\mathrm{dv} = \mathbf{C}^{e\,\mathrm{T}}\mathbf{f}_r^e, \quad \Rightarrow \tag{349}$$

$$\mathbf{F}_r = \sum_{I=1}^{n_{\mathrm{e}}} \mathbf{F}_r^{e_I} = \sum_{I=1}^{n_{\mathrm{e}}} \mathbf{C}^{e_I\,\mathrm{T}}\mathbf{f}_r^{e_I} \quad \text{for} \tag{350a}$$

$$\mathbf{f}_r^e = \int_e \mathbf{N}^{e\,\mathrm{T}}.\mathbf{r}\,\mathrm{dv} \tag{350b}$$

Local to Global: Natural BC force vector

1. Natural BC force vector From (342b) ($\vec{\mathbf{N}}$ = normal vector not the shape function):

$$\mathbf{F}_N = \left(\mathbf{N}^{\mathrm{T}}, \bar{\mathbf{F}}\right)_N = \int_{\partial \mathcal{D}_f} \mathbf{N}^{\mathrm{T}} \bar{\mathbf{F}}.\vec{\mathbf{N}} \, \mathrm{ds}$$

- We first break domain of integration using $\partial \mathcal{D}_f = \cup_{I=1}^{n_{\mathrm{e}}} \partial e_f^I$ (disjoint union) where,

$$\partial e_f^I = \partial \mathcal{D}_f \cap \partial e_I \tag{351}$$

is the part of the boundary of e_I on natural boundary $\partial \mathcal{D}_f$.

$$\mathbf{F}_N = \sum_{I=1}^{n_{\mathrm{e}}} \mathbf{F}_N^{e_I} \quad \text{where}$$

$$\mathbf{F}_N^{e_I} = \int_{\partial e_f^I} \mathbf{N}^{\mathrm{T}} \bar{\mathbf{F}}.\vec{\mathbf{N}} \, \mathrm{ds} \tag{352}$$

- Second, having $\mathbf{N} = \mathbf{N}^e \mathbf{C}^e$ within element e we have:

$$\mathbf{F}_N^e = \int_{\partial e_f} \mathbf{N}^{\mathrm{T}} \bar{\mathbf{F}}.\vec{\mathbf{N}} \, \mathrm{ds} = \int_{\partial e_f} (\mathbf{N}^e \mathbf{C}^e)^{\mathrm{T}} \bar{\mathbf{F}}.\vec{\mathbf{N}} \, \mathrm{ds} = \mathbf{C}^{e\,\mathrm{T}} \mathbf{f}_N^e, \quad \Rightarrow \tag{353}$$

$$\mathbf{F}_N = \sum_{I=1}^{n_{\mathrm{e}}} \mathbf{F}_N^{e_I} = \sum_{I=1}^{n_{\mathrm{e}}} \mathbf{C}^{e_I\,\mathrm{T}} \mathbf{f}_N^{e_I} \quad \text{for} \tag{354a}$$

$$\mathbf{f}_N^e = \int_{\partial e_f} \mathbf{N}^{e\,\mathrm{T}} \bar{\mathbf{F}}.\vec{\mathbf{N}} \, \mathrm{ds} \tag{354b}$$

Local to Global: Essential BC force vector

4. Essential BC force vector We observed from (342c):

$$\mathbf{F}_D = \mathbf{K}_{fp}\bar{\mathbf{a}} \quad \text{where } \mathbf{K}_{fp} = \int_{\mathcal{D}} \mathbf{B}^{\mathrm{T}}\mathbf{D}\bar{\mathbf{B}}\,\mathrm{dv} \tag{355}$$

- First, we break domain of integration using $\mathcal{D} = \cup_{I=1}^{n_e} e_I$ (disjoint union):

$$\mathbf{K}_{fp} = \sum_{I=1}^{n_e} \mathbf{K}_{fp}^{e_I} \tag{356}$$

$$\mathbf{K}_{fp}^{e_I} = \int_{e_I} \mathbf{B}^{\mathrm{T}}\mathbf{D}\bar{\mathbf{B}}\,\mathrm{dv}$$

- Second, having $\mathbf{B} = \mathbf{B}^e\mathbf{C}^e$ and similarly $\bar{\mathbf{B}} = \mathbf{B}^e\bar{\mathbf{C}}^e$ within element e we have:

$$\mathbf{K}_{fp}^e = \int_e \mathbf{B}^{\mathrm{T}}\mathbf{D}\bar{\mathbf{B}}\,\mathrm{dv} = \int_e (\mathbf{B}^e\mathbf{C}^e)^{\mathrm{T}}\mathbf{D}(\mathbf{B}^e\bar{\mathbf{C}}^e)\,\mathrm{dv} \Rightarrow \mathbf{K}_{fp}^e = \mathbf{C}^{e\,\mathrm{T}}\mathbf{k}^e\bar{\mathbf{C}}^e,$$

$$\text{as before} \quad \mathbf{k}^e = \int_e \mathbf{B}^{e\,\mathrm{T}}\mathbf{D}\mathbf{B}^e\,\mathrm{dv} \quad \Rightarrow (\text{(355) and (356)}) \tag{357}$$

$$\mathbf{F}_D = \sum_{I=1}^{n_e}\{\mathbf{K}_{fp}^{e_I}\bar{\mathbf{a}}\} = \sum_{I=1}^{n_e} \mathbf{C}^{e_I\,\mathrm{T}}\mathbf{f}_D^{e_I} \text{ where} \tag{358a}$$

$$\mathbf{f}_D^e = \mathbf{k}^e\bar{\mathbf{C}}^e\bar{\mathbf{a}} \tag{358b}$$

Local to Global: Essential BC force vector

- We can further simplify $\mathbf{f}_D^e$ in (358b) by noting that $\bar{\mathbf{C}}^e\bar{\mathbf{a}}$ is the element level $\mathbf{a}^e$ with 0 values used for dofs that are free:

$$\bar{\mathbf{C}}^e\bar{\mathbf{a}} = \mathbf{a}^e \tag{359}$$

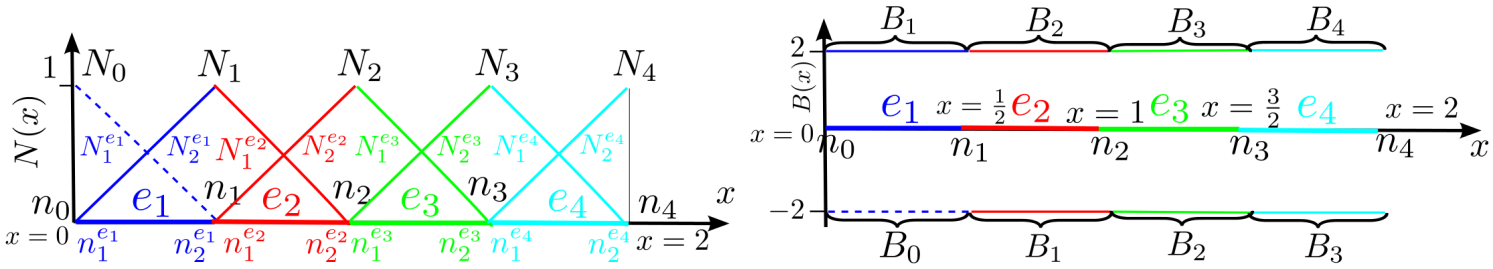
- Thus, the essential BC force simplifies to,

$$\mathbf{F}_D = \sum_{I=1}^{n_e} \mathbf{C}^{e_I\,\mathrm{T}}\mathbf{f}_D^{e_I} \quad \text{for} \tag{360a}$$

$$\mathbf{f}_D^e = \mathbf{k}^e\mathbf{a}^e \tag{360b}$$

- In practice, only certain rows and columns of this matrix product are performed.
  1. Only free rows: As eventually only forces for free dofs are added to system force vector.
  2. Only prescribed columns: In the product of free rows of $\mathbf{k}^e$ and $\mathbf{a}^e$ the free dof. are ignored as they should not be incorporated in this matrix product (that is why we put zero value for free dofs in $\mathbf{a}^e$ for the product $\mathbf{f}_D^e = \mathbf{k}^e\mathbf{a}^e$).
- That is our local matrix vector product is in accord to global relation: $\mathbf{F}_D = \mathbf{K}_{fp}\bar{\mathbf{a}}$ (free dof for rows, prescribed dofs for columns).

Local to Global: Essential BC force vector



- To better illustrate $\bar{\mathbf{C}}^e \bar{\mathbf{a}} = \mathbf{a}^e$, consider our 1D example:

$$\bar{\mathbf{C}}^{e_1} = \begin{bmatrix} 1 \\ 0 \end{bmatrix} \quad \bar{\mathbf{a}} = [1] \text{ (prescribed displacement for node 0)} \quad \bar{\mathbf{C}}^{e_1} \bar{\mathbf{a}} = \begin{bmatrix} 1 \\ 0 \end{bmatrix}$$

  which is equal to $\mathbf{a}_1^e = \begin{bmatrix} 1 \\ 0 \end{bmatrix}$ formed by setting the correct displacement values for prescribed dofs and zero for free ones which are not determined yet.

- For e_3 we have:

$$\bar{\mathbf{C}}^{e_3} = \begin{bmatrix} 0 \\ 0 \end{bmatrix} \quad \bar{\mathbf{a}} = [1] \quad \bar{\mathbf{C}}^{e_3} \bar{\mathbf{a}} = \begin{bmatrix} 0 \\ 0 \end{bmatrix}$$

  which is equal to $\mathbf{a}_3^e = \begin{bmatrix} 0 \\ 0 \end{bmatrix}$ again formed by setting zeros for not yet solved free dofs and correct prescribed dofs (none for this element).

Summary: Global vs Local system: equations

- From (343), (346), (350), (354), (360) we summarize element to global system relations as,

$$\mathbf{K} = \sum_{I=1}^{n_e} \mathbf{K}^{eI} = \sum_{I=1}^{n_e} \mathbf{C}^{eI\,\mathrm{T}} \mathbf{k}^{eI}\, \mathbf{C}^{eI} \tag{361a}$$

$$\mathbf{F} = \mathbf{F}_n + \mathbf{F}_e, \text{ where element force contributions } \mathbf{F}_e := \sum_{I=1}^{n_e} \mathbf{C}^{eI\,\mathrm{T}} \mathbf{f}_e^{eI} \tag{361b}$$

- Local stiffness matrix, $\mathbf{k}^e$, is given by (*cf.* (346b) and (340)):

$$\mathbf{k}^e = \int_e \mathbf{B}^{e\,\mathrm{T}} \mathbf{D} \mathbf{B}^e \,\mathrm{dv}, \quad \text{where} \quad \mathbf{B}^e = L_m(\mathbf{N}^e) \tag{362}$$

- Local force vector, $\mathbf{f}_e^e$, is the assembly of all element level forces:

$$\mathbf{f}_e^e := \mathbf{f}_r^e + \mathbf{f}_N^e - \mathbf{f}_D^e \tag{363a}$$

$$\mathbf{f}_r^e = \int_e \mathbf{N}^{e\,\mathrm{T}}.\mathbf{r}\,\mathrm{dv} \tag{363b}$$

$$\mathbf{f}_N^e = \int_{\partial e_f} \mathbf{N}^{e\,\mathrm{T}} \bar{\mathbf{F}}.\,\mathrm{ds} \tag{363c}$$

$$\mathbf{f}_D^e = \mathbf{k}^e \mathbf{a}^e \tag{363d}$$

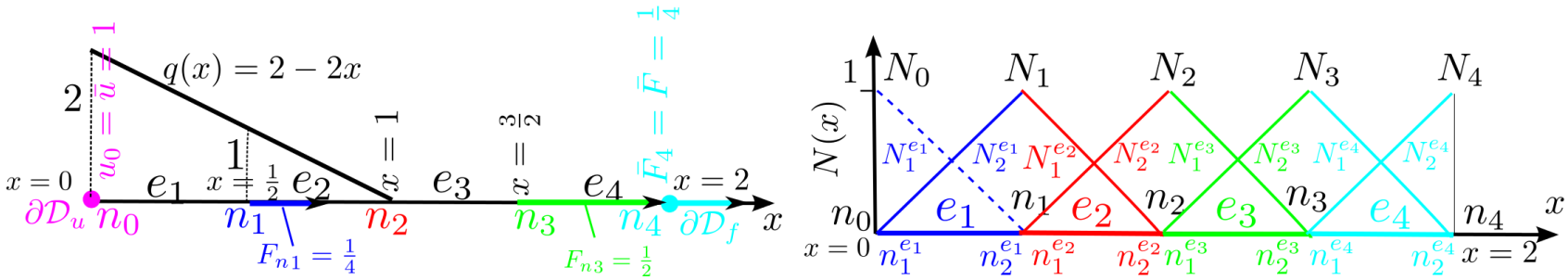
Summary: Global vs Local system: equations

- In $\mathbf{C}^{e\mathrm{T}}\mathbf{k}^e\mathbf{C}^e$ rows and columns of $\mathbf{k}^e$ are added to particular places in global matrix based on the map between local and global dofs: $\mathbf{M}^e$. We do not actually perform the product by boolean matrices in $\mathbf{C}^{e\mathrm{T}}\mathbf{k}^e\mathbf{C}^e$.
- Similarly, in $\mathbf{C}^{e\mathrm{T}}\mathbf{f}_e^e$ we do not actually compute the product operation and use $\mathbf{M}^e$ to add components of $\mathbf{f}_e^e$ to right places in $\mathbf{F}_e$.
- Nodal forces are global in nature and do not have a local counterpart.
- All other loads (source term, natural BC, and essential BC) are lumped into $\mathbf{f}_e^e$ and eventually added to $\mathbf{F}$ through $\mathbf{F}_e$.

Summary: Global vs Local system

	Term	Global	Local
Numbers	Number of elements	n_{e}	-
	Number of nodes	n_{n}	n_{n}^{e}
	Number of free dof.	n_{f}	n_{f}^{e}
	Number of prescribed dof.	n_{p}	n_{p}^{e}
	Number of dof. (unknowns)	$n_{\mathrm{dof}} = n_{\mathrm{f}} + n_{\mathrm{p}}$	$n_{\mathrm{dof}}^{e} = n_{\mathrm{f}}^{e} + n_{\mathrm{p}}^{e}$
	Number of dof. per node	n_{dpn}	
$\mathbf{N}$ related	Shape functions	free: $\mathbf{N}(1 \times n_{\mathrm{f}})$ & prescribed: $\bar{\mathbf{N}}(1 \times n_{\mathrm{p}})$	$\mathbf{N}^{e}$ $(1 \times n_{\mathrm{dof}}^{e})$ (both free and prescribed)
	Kinematic matrix	free: $\mathbf{B}(1 \times n_{\mathrm{f}})$ & prescribed: $\bar{\mathbf{B}}(1 \times n_{\mathrm{p}})$	$\mathbf{B}^{e}$ $(1 \times n_{\mathrm{dof}}^{e})$ (both free and prescribed)
	Free dof connectivity matrix	$\mathbf{C}^{e}$: $(n_{\mathrm{dof}}^{e} \times n_{\mathrm{f}})$, defined by $\mathbf{N} = \mathbf{N}^{e}\mathbf{C}^{e}$ within element e $(\Rightarrow \mathbf{B} = \mathbf{B}^{e}\mathbf{C}^{e})$	
	Prescribed dof connectivity matrix	$\bar{\mathbf{C}}^{e}$: $(n_{\mathrm{dof}}^{e} \times n_{\mathrm{p}})$, defined by $\bar{\mathbf{N}} = \mathbf{N}^{e}\bar{\mathbf{C}}^{e}$ within element e $(\Rightarrow \bar{\mathbf{B}} = \mathbf{B}^{e}\bar{\mathbf{C}}^{e})$	
Linear equation	Nodal values	$\mathbf{a}$ free dof.s $(n_{\mathrm{f}} \times 1)$	$\mathbf{a}^{e}$ $(n_{\mathrm{dof}}^{e} \times 1)$ (both free and prescribed)
	Prescribed nodal values	$\bar{\mathbf{a}}$ $(n_{\mathrm{p}} \times 1)$	Not needed
	Stiffness (conductivity) matrix	$\mathbf{K} = \int_{\mathcal{D}} \mathbf{B}^{\mathrm{T}}\mathbf{D}\mathbf{B}\,\mathrm{dv}$ $(n_{\mathrm{f}} \times n_{\mathrm{f}})$	$\mathbf{k}^{e} = \int_{e} \mathbf{B}^{e\,\mathrm{T}}\mathbf{D}\mathbf{B}^{e}\,\mathrm{dv}$ $(n_{\mathrm{dof}}^{e} \times n_{\mathrm{dof}}^{e})$
	Prescribed to free stiffness matrix	$\mathbf{K}_{\mathrm{fp}} = \int_{\mathcal{D}} \mathbf{B}^{\mathrm{T}}\mathbf{D}\bar{\mathbf{B}}\,\mathrm{dv}$ $(n_{\mathrm{f}} \times n_{\mathrm{p}})$	Not needed
	Force (load): Source term	$\mathbf{F}_r = \int_{\mathcal{D}} \mathbf{N}^{\mathrm{T}}\mathbf{r}\,\mathrm{dv}$ $(n_{\mathrm{f}} \times 1)$	$\mathbf{f}_r^{e} = \int_{e} \mathbf{N}^{e\,\mathrm{T}}\mathbf{r}\,\mathrm{dv}$ $(n_{\mathrm{dof}}^{e} \times 1)$
	Force (load): Natural BC	$\mathbf{F}_N = \int_{\partial\mathcal{D}_f} \mathbf{N}^{\mathrm{T}}\bar{\mathbf{f}}.\mathbf{N}\,\mathrm{ds}$ $(n_{\mathrm{f}} \times 1)$	$\mathbf{f}_N^{e} = \int_{\partial e_f} \mathbf{N}^{e\,\mathrm{T}}\bar{\mathbf{f}}\,\mathrm{ds}.\mathbf{N}$ $(n_{\mathrm{dof}}^{e} \times 1)$
	Force (load): Essential BC	$\mathbf{F}_D = \mathbf{K}_{\mathrm{pf}}\bar{\mathbf{a}}$ $(n_{\mathrm{f}} \times 1)$	$\mathbf{f}_D^{e} = \mathbf{k}^{e}\mathbf{a}^{e}$ $(n_{\mathrm{dof}}^{e} \times 1)$
	Force (load): Nodal	$\mathbf{F}_n$: specified at free dof $(n_{\mathrm{f}} \times 1)$	Not Applicable
	Force (load): all but Nodal	$\mathbf{F}_e = \mathbf{F}_r + \mathbf{F}_N - \mathbf{F}_D$ $(n_{\mathrm{f}} \times 1)$	$\mathbf{f}_e^{e} = \mathbf{f}_r^{e} + \mathbf{f}_N^{e} - \mathbf{f}_D^{e}$ $(n_{\mathrm{dof}}^{e} \times 1)$
	Force (load): Total	$\mathbf{F} = \mathbf{F}_e + \mathbf{F}_n$ $(n_{\mathrm{f}} \times 1)$	Not Applicable

Assembly of the global system: Bar example

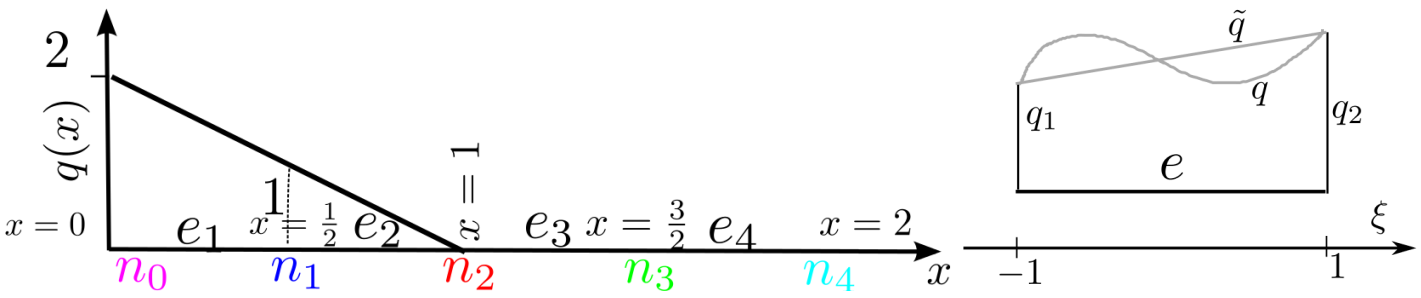


- Previously, in (327) we derived global stiffness matrix by an element-centered approach where integrals were broken to element level from the beginning. But, as we discussed that approach was not completely local as element level integrations were still tied up to global numbering.
- We remedied this problem by deriving entirely local, element-level, stiffness matrices and force vectors.
- We want to show this approach on the same problem we solved before.
- Local stiffness matrix: Since, $E = 1, A = 1, L = \frac{1}{2}$ for all elements, $\mathbf{k}^e$ is given by (328):

$$\mathbf{k}^e = \frac{(1).(1)}{1/2} \begin{bmatrix} 1 & -1 \\ -1 & 1 \end{bmatrix} = \begin{bmatrix} 2 & -2 \\ -2 & 2 \end{bmatrix}$$

- For element forces:
  - $\mathbf{f}_r^e$: Source term (body force) force discussed on the next slide.
  - $\mathbf{f}_N^e$: Natural BC force is only applied at $\partial\mathcal{D}_f = \{2\}$ and can be treated as a global nodal force along with nonzero forces applied at n_1 and n_3: $\mathbf{F}_N = [\frac{1}{4}\ 0\ \frac{1}{2}\ \frac{1}{4}]^{\mathrm{T}}$.
  - $\mathbf{f}_D^e$: Essential BC force will only be nonzero in e_1, since $\partial\mathcal{D}_u = \{0\}$ with $u_0 = 1$.

Assembly of global system: Bar example



- Another aspect of this problem is the body force (source term) that is applied on elements e_1 and e_2.
- If we approximate q by a linear function $\tilde{q}$ within an element e we have

$$\tilde{q} = N_1 q_1 + N_2 q_2 = \mathbf{N}^e \begin{bmatrix} q_1 \\ q_2 \end{bmatrix} \text{ and} \tag{364}$$

$$\mathbf{f}_r^e = \int_e \mathbf{N}^{e\,\mathrm{T}}.q \,\mathrm{d}x \approx \int_e \mathbf{N}^{e\,\mathrm{T}}.\tilde{q} \,\mathrm{d}x = \{\int_e \mathbf{N}^{e\,\mathrm{T}}.\mathbf{N}^e \,\mathrm{d}x\} \begin{bmatrix} q_1 \\ q_2 \end{bmatrix} \quad \textit{cf.}\ (363b)$$

- We define the matrix $\mathbf{r}^e := \int_e \mathbf{N}^{e\,\mathrm{T}}.\mathbf{N}^e \,\mathrm{d}x$ and later show that:

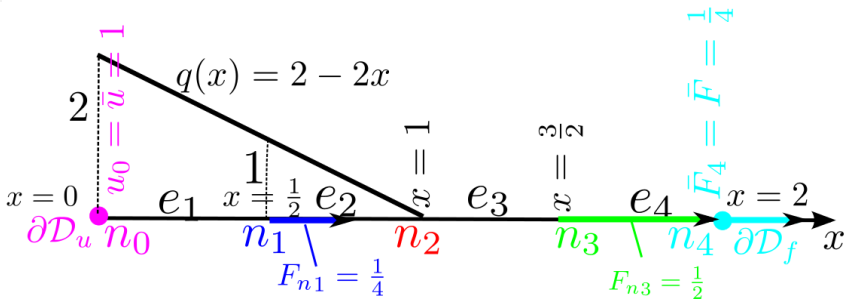
$$\mathbf{r}^e := \int_e \mathbf{N}^{e\,\mathrm{T}}.\mathbf{N}^e \,\mathrm{d}x = \frac{L}{6}\begin{bmatrix} 2 & 1 \\ 1 & 2 \end{bmatrix} \quad \Rightarrow \mathbf{r}^e = \frac{1}{12}\begin{bmatrix} 2 & 1 \\ 1 & 2 \end{bmatrix} \text{ for } e_1, \ldots, e_4 \tag{365}$$

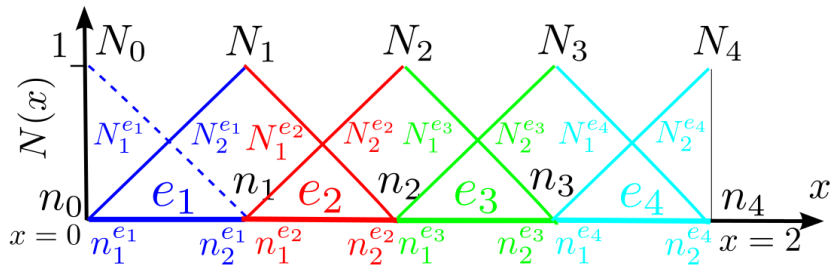
- Thus,

$$\mathbf{f}_r^e \approx \mathbf{r}^e \begin{bmatrix} q_1 \\ q_2 \end{bmatrix}, \quad \text{(exact for linear } q \text{)}$$

- As shown in the figure, for our sample problem q is linear in e_1 and e_2 and the above relation is exact.

Assembly of global system: Bar example



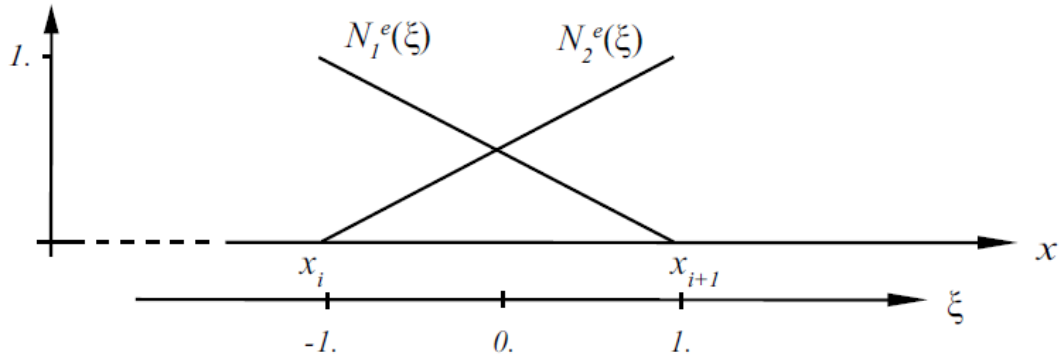


e	e_1	e_2	e_3	e_4
$\mathbf{k}^e$	$\begin{matrix} & \bar{1} & 1 \\ \bar{1} & 2 & -2 \\ 1 & -2 & 2 \end{matrix}$	$\begin{matrix} & 1 & 2 \\ 1 & 2 & -2 \\ 2 & -2 & 2 \end{matrix}$	$\begin{matrix} & 2 & 3 \\ 2 & 2 & -2 \\ 3 & -2 & 2 \end{matrix}$	$\begin{matrix} & 3 & 4 \\ 3 & 2 & -2 \\ 4 & -2 & 2 \end{matrix}$
$\mathbf{f}_r^e$	$\mathbf{r}^{e1}\mathbf{q}^{e1} = \begin{bmatrix} \frac{1}{6} & \frac{1}{12} \\ \frac{1}{12} & \frac{1}{6} \end{bmatrix} \begin{bmatrix} 2 \\ 1 \end{bmatrix} = \begin{matrix} \bar{1} \\ 1 \end{matrix} \begin{bmatrix} \frac{5}{12} \\ \frac{4}{12} \end{bmatrix}$	$\mathbf{r}^{e2}\mathbf{q}^{e2} = \begin{bmatrix} \frac{1}{6} & \frac{1}{12} \\ \frac{1}{12} & \frac{1}{6} \end{bmatrix} \begin{bmatrix} 1 \\ 0 \end{bmatrix} = \begin{matrix} 1 \\ 2 \end{matrix} \begin{bmatrix} \frac{2}{12} \\ \frac{1}{12} \end{bmatrix}$	$\mathbf{f}_r^{e3} = \mathbf{r}^{e3}\mathbf{q}^{e3} = \begin{bmatrix} 0 \\ 0 \end{bmatrix}$	$\mathbf{f}_r^{e4} = \mathbf{r}^{e4}\mathbf{q}^{e4} = \begin{bmatrix} 0 \\ 0 \end{bmatrix}$
$\mathbf{f}_D^e$	$\mathbf{k}^{e1}\mathbf{a}_1^e = \begin{bmatrix} 2 & -2 \\ -2 & 2 \end{bmatrix} \begin{bmatrix} 1 \\ 0 \end{bmatrix} = \begin{matrix} \bar{1} \\ 1 \end{matrix} \begin{bmatrix} 2 \\ -2 \end{bmatrix}$	$\mathbf{k}^{e2}\mathbf{a}_2^e = \begin{bmatrix} 0 \\ 0 \end{bmatrix}$	$\mathbf{k}^{e3}\mathbf{a}_3^e = \begin{bmatrix} 0 \\ 0 \end{bmatrix}$	$\mathbf{k}^{e4}\mathbf{a}_4^e = \begin{bmatrix} 0 \\ 0 \end{bmatrix}$
$\mathbf{f}_e^e$	$\mathbf{f}_e^{e1} = \mathbf{f}_r^{e1} + \mathbf{f}_N^{e1} - \mathbf{f}_D^{e1} = \begin{matrix} \bar{1} \\ 1 \end{matrix} \begin{bmatrix} -\frac{19}{12} \\ \frac{28}{12} \end{bmatrix}$	$\mathbf{f}_e^{e2} = \mathbf{f}_r^{e2} + \mathbf{f}_N^{e2} - \mathbf{f}_D^{e2} = \begin{matrix} 1 \\ 2 \end{matrix} \begin{bmatrix} \frac{2}{12} \\ \frac{1}{12} \end{bmatrix}$	$\mathbf{f}_e^{e3} = \begin{matrix} 2 \\ 3 \end{matrix} \begin{bmatrix} 0 \\ 0 \end{bmatrix}$	$\mathbf{f}_e^{e4} = \begin{matrix} 3 \\ 4 \end{matrix} \begin{bmatrix} 0 \\ 0 \end{bmatrix}$

$$\mathbf{K} = \begin{bmatrix} 2+2 & -2 & 0 & 0 \\ -2 & 2+2 & -2 & 0 \\ 0 & -2 & 2+2 & -2 \\ 0 & 0 & -2 & 2 \end{bmatrix} = \begin{bmatrix} 4 & -2 & 0 & 0 \\ -2 & 4 & -2 & 0 \\ 0 & -2 & 4 & -2 \\ 0 & 0 & -2 & 2 \end{bmatrix}, \mathbf{F} = \mathbf{F}_n + \mathbf{F}_e = \begin{bmatrix} \frac{1}{4} \\ 0 \\ \frac{1}{2} \\ \frac{1}{4} \end{bmatrix} + \begin{bmatrix} \frac{28}{12} + \frac{2}{12} \\ \frac{1}{12} \\ 0 \\ 0 \end{bmatrix} = \begin{bmatrix} \frac{11}{4} \\ \frac{1}{12} \\ \frac{1}{2} \\ \frac{1}{4} \end{bmatrix} \Rightarrow$$

$$\mathbf{a} = \mathbf{K}^{-1}\mathbf{F} = \begin{bmatrix} \frac{43}{24} \\ \frac{53}{24} \\ \frac{31}{12} \\ \frac{65}{24} \end{bmatrix}$$ which matches our solution from global approach (321)

Local coordinate system

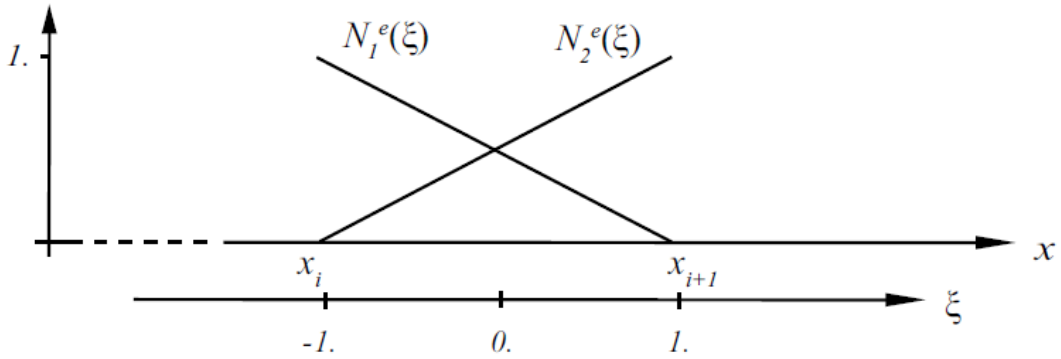


- It is customary to use a local coordinate system for the integration of element stiffness matrix and force vectors.
- For example the element stiffness matrix is given by,

$$\mathbf{k}^e = \int_e \mathbf{B}^{e\,\mathrm{T}} EA \mathbf{B}^e \mathrm{d}x = \int_e \begin{bmatrix} B_1^e \\ B_2^e \end{bmatrix} EA \begin{bmatrix} B_1^e & B_2^e \end{bmatrix} \mathrm{d}x = \int_{x_i}^{x_{i+1}} \begin{bmatrix} \frac{\mathrm{d}N_1^e}{\mathrm{d}x} \\ \frac{\mathrm{d}N_2^e}{\mathrm{d}x} \end{bmatrix} EA \begin{bmatrix} \frac{\mathrm{d}N_1^e}{\mathrm{d}x} & \frac{\mathrm{d}N_2^e}{\mathrm{d}x} \end{bmatrix} \mathrm{d}x \tag{366}$$

- If we express N_1^e and N_2^e in terms of local coordinate system ξ we need to do the following to evaluate $\mathbf{k}^e$:
  1. Express $\mathbf{N}^e$ in terms of ξ.
  2. Express x in terms of ξ.
  3. Obtain a relation between $\frac{\mathrm{d}N_1^e}{\mathrm{d}\xi}$ and $\frac{\mathrm{d}N_1^e}{\mathrm{d}x}$.
  4. Obtain a relation between $\mathrm{d}x$ and $\mathrm{d}\xi$.
- While the use of local coordinate system ξ seems cumbersome, we will later see that all the elements of the same type and different geometries will be expressed and integrated in a unified manner, which is a major advantage.

Stiffness matrix: Local coordinate system



- In the local coordinate system shown, the left side of the bar corresponds to $\xi = -1$ and the right side to $\xi = 1$. Using the property $N_i^e(n_j) = \delta_{ij}$ we observe:

$$N_1^e(\xi) = (1 - \xi)/2 \tag{367a}$$

$$N_2^e(\xi) = (1 + \xi)/2 \tag{367b}$$

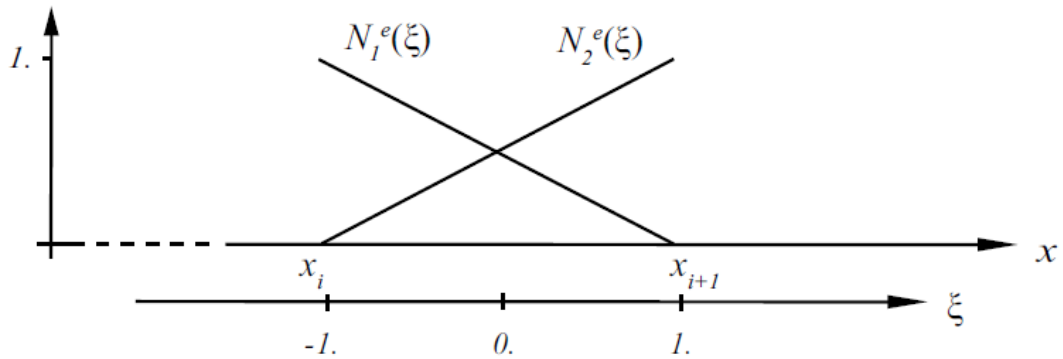
- Next, we look for a relation between x and ξ. We know that the relation between the two is linear:

$$\left.\begin{array}{lll} x & = A\xi + B & \\ \xi & = -1 & \Rightarrow x = x_i \\ \xi & = 1 & \Rightarrow x = x_{i+1} \end{array}\right\} \Rightarrow \left.\begin{array}{ll} B - A & = x_i \\ B + A & = x_{i+1} \end{array}\right\} \Rightarrow x = \frac{x_{i+1} - x_i}{2}\xi + \frac{x_{i+1} + x_i}{2}$$

- Instead of the approach above, we can directly use the $N_i^e(n_j) = \delta_{ij}$ property of the shape functions to form this map:

$$x = x_i N_1^e(\xi) + x_{i+1} N_2^e(\xi) = \frac{1 - \xi}{2} x_i + \frac{1 + \xi}{2} x_{i+1} \tag{368}$$

Stiffness matrix: Local coordinate system



- Denoting,

$$x_m^e := \frac{x_{i+1} + x_i}{2} \tag{369a}$$

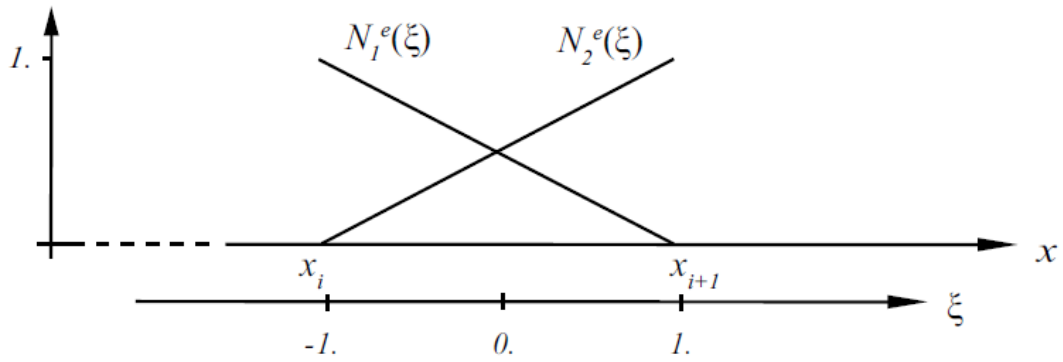
$$L^e := x_{i+1} - x_i \tag{369b}$$

we observe both approaches yield:

$$x = \frac{L^e}{2}\xi + x_m^e \tag{370}$$

- Later, in the context of iso-, sup-, and super-parametric elements we observe how similar to the second approach in the last slide, we can use shape functions to manufacture the map between x and ξ. In fact, shape functions can be used to express solution fields $\mathbf{u}$, geometry $\mathbf{x}$, source term $\mathbf{r}$ and many other things within an element.

Stiffness matrix: Local coordinate system



- As the second stage, we compute $\mathbf{B}^e$:

$$\mathbf{B}^e = [\frac{\mathrm{d}N_1^e}{\mathrm{d}x} \quad \frac{\mathrm{d}N_2^e}{\mathrm{d}x}]$$

- Since, $\mathbf{N}^e$ is expressed in terms of ξ in (367), we have,

$$\mathbf{B}^e = [\frac{\mathrm{d}N_1^e}{\mathrm{d}\xi}/\frac{\mathrm{d}x}{\mathrm{d}\xi} \quad \frac{\mathrm{d}N_2^e}{\mathrm{d}\xi}/\frac{\mathrm{d}x}{\mathrm{d}\xi}]$$

- From (370) we get,

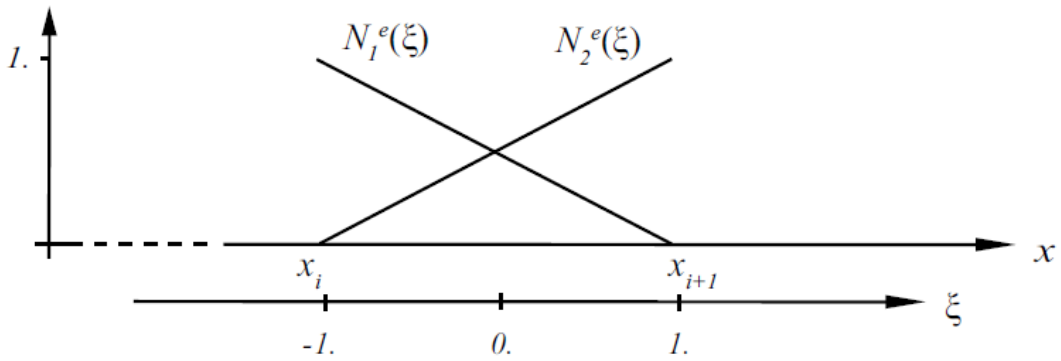
$$\frac{\mathrm{d}x}{\mathrm{d}\xi} = \frac{L^e}{2} \quad \Rightarrow$$

$$\mathbf{B}^e = [-\frac{1}{2}.\frac{2}{L^e} \quad \frac{1}{2}.\frac{2}{L^e}] = [-\frac{1}{L^e} \quad \frac{1}{L^e}] \tag{371}$$

and

$$\mathrm{d}x = \frac{L^e}{2}\mathrm{d}\xi \tag{372}$$

Stiffness matrix: Local coordinate system



- Finally, we plug (371) and (372) into (366) to obtain,

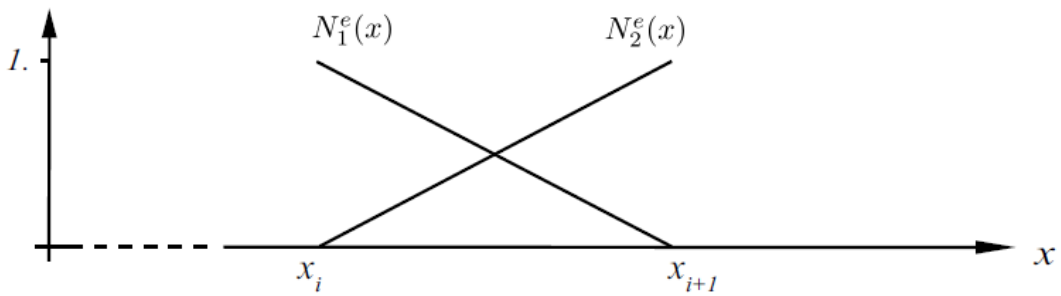
$$\mathbf{k}^e = \int_{-1}^{1} \begin{bmatrix} -\frac{1}{L^e} \\ \frac{1}{L^e} \end{bmatrix} E(\xi)A(\xi) \begin{bmatrix} -\frac{1}{L^e} & \frac{1}{L^e} \end{bmatrix} \{\frac{L^e}{2} \mathrm{d}\xi\} \quad \Rightarrow$$

$$\boxed{\mathbf{k}^e = \frac{1}{2L^e} \int_{-1}^{1} E(\xi)A(\xi)\, \mathrm{d}\xi \begin{bmatrix} 1 & -1 \\ -1 & 1 \end{bmatrix}} \tag{373}$$

- If A and E are constant along the bar, we have:

$$\boxed{\mathbf{k}^e = \frac{AE}{L^e} \begin{bmatrix} 1 & -1 \\ -1 & 1 \end{bmatrix} \quad \text{(constant } A \text{ and } E\text{)}} \tag{374}$$

Stiffness matrix: Global coordinate system



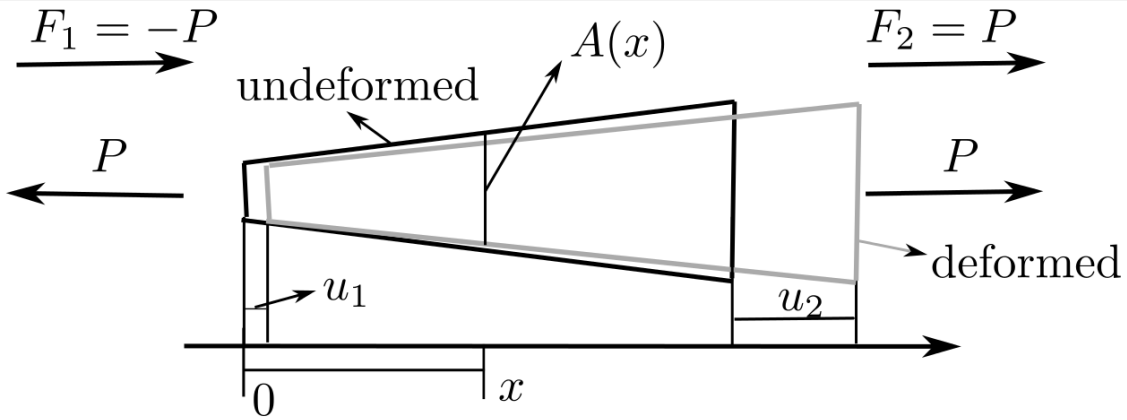
- In this simple bar element, it is in fact easier to derive the stiffness matrix without going to the local coordinate system ξ:

$$\left.\begin{array}{ll} N_1^e & = \frac{x_{i+1}-x}{L^e} \\ N_2^e & = \frac{x-x_i}{L^e} \end{array}\right\} \quad \Rightarrow \quad \left.\begin{array}{ll} B_1^e & = -\frac{1}{L^e} \\ B_2^e & = \frac{1}{L^e} \end{array}\right\} \quad \Rightarrow$$

$$\mathbf{k}^e = \int_{x_i}^{x_{i+1}} \begin{bmatrix} -\frac{1}{L^e} \\ \frac{1}{L^e} \end{bmatrix} A(x)E(x) \begin{bmatrix} -\frac{1}{L^e} & \frac{1}{L^e} \end{bmatrix} \mathrm{d}x = \frac{AE}{L^e} \begin{bmatrix} 1 & -1 \\ -1 & 1 \end{bmatrix} \text{ for constant} A, E$$

- The coordinate transformations we used in previous slides is used in the same way for more complicated elements in higher dimensions where the benefits of that approach are more apparent.

Stiffness matrix: Direct method



- The engineering concepts of finite element method were first based on directly obtaining relations between deformations and forces without restoring to shape functions (*e.g.*, assumed linear displacement field we used before).
- First, we observe that for static equilibrium $F_1 = -F_2$. We denote $P := F_2$.
- Second, we mechanics equations to obtain u_2 assuming that the displacement is equal to u_1 at the left end $(x = 0)$. Let $F(x)$ be the axial force at x:

$$F(x) = P \qquad \text{Axial force} \qquad (375a)$$

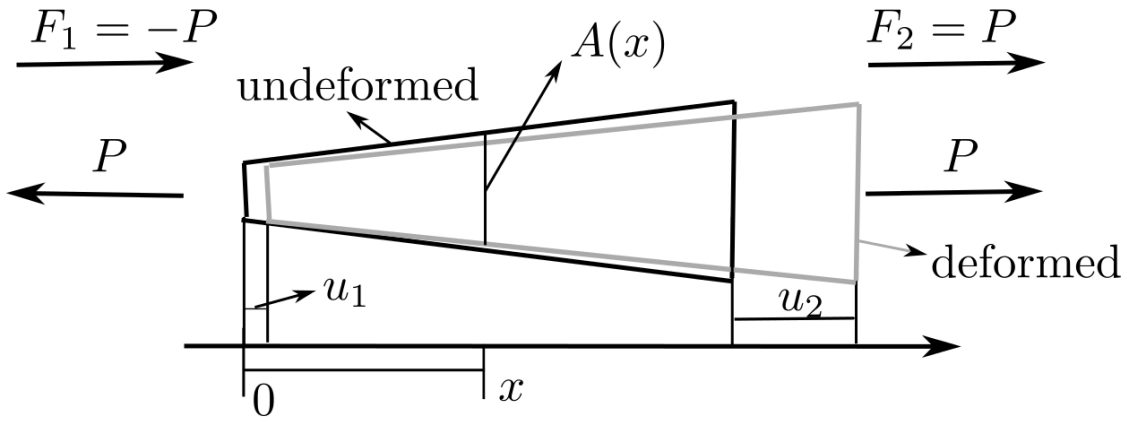
$$s(x) = \frac{F(x)}{A(x)} \qquad \text{Stress} \qquad (375b)$$

$$\epsilon(x) = \frac{s(x)}{E(x)} = \frac{F(x)}{A(x)E(x)} \qquad \text{Strain} \qquad (375c)$$

$$u(x) = u_1 + \int_0^x \epsilon(y)\,\mathrm{d}y = u_1 + \int_0^L \frac{F(y)}{A(y)E(y)}\,\mathrm{d}y \qquad \text{Displacement} \qquad (375d)$$

$$u_2 - u_1 = \int_0^L \frac{F(x)}{A(x)E(x)}\,\mathrm{d}x = P\int_0^L \frac{1}{A(x)E(x)}\,\mathrm{d}x \qquad \text{Displacement jump} \qquad (375e)$$

Stiffness matrix: Direct method



- Thus, noting that $F_2 = -F_1 = P$ we have,

$$\boxed{\mathbf{k}^e = \frac{1}{\int_0^L \frac{1}{E(x)A(x)}\,\mathrm{d}x} \begin{bmatrix} 1 & -1 \\ -1 & 1 \end{bmatrix}} \tag{376}$$

- If A and E are constant along the bar, we have:

$$\boxed{\mathbf{k}^e = \frac{AE}{L^e} \begin{bmatrix} 1 & -1 \\ -1 & 1 \end{bmatrix} \quad \text{(constant } A \text{ and } E\text{)}} \tag{377}$$

- We observe that (374) and (377) coincide. Why is that?
- Do (373) and (376) match in general? If not, why? Which one is approximate and which one is exact?

Stiffness matrix: Direct vs. FEM method

- Do (373) (FEM) and (376) (Direct)match in general? If not, why? Which one is approximate and which one is exact?

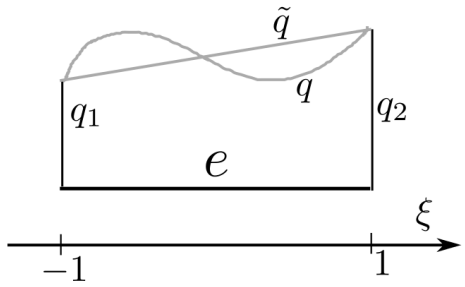
$$\mathbf{k}^e = \frac{1}{2L^e}\int_{-1}^{1} E(\xi)A(\xi)\,\mathrm{d}\xi \begin{bmatrix} 1 & -1 \\ -1 & 1 \end{bmatrix} \text{ FEM, } \mathbf{k}^e = \frac{1}{\int_0^L \frac{1}{E(x)A(x)}\,\mathrm{d}x} \begin{bmatrix} 1 & -1 \\ -1 & 1 \end{bmatrix} \text{ Direct}$$

  - No, they don't match in general.
  - This is because, in FEM approach displacement is approximated to be linear by using linear shape function where as as we see from (375d), in direct method displacement can be nonlinear when $A(x)E(x)$ is not constant.
  - Direct method is exact, FEM is approximate for nonconstant $A(x)E(x)$.
- We observe that (FEM) and (Direct) coincide for constant A and E. Why is that?

$$\mathbf{k}^e = \frac{AE}{L^e} \begin{bmatrix} 1 & -1 \\ -1 & 1 \end{bmatrix} \quad \text{(constant } AE \text{ for both Direct (377) and FEM (374) methods)}$$

  - They match because in this case the exact displacement field is linear, which can be exactly captured by linear shape functions in FEM.
- While direct method is simple enough for 1D bar example, its calculation is much more involved in general and mixing it with other quantities (*e.g.*, element forces) that may be derived using shape functions may lead to inconsistencies. . That is, why we just use shape functions to derive element forces and stiffness matrix.
- The error in stiffness matrices between direct and FEM approach is a part of discretization error, and does not influence the convergence rate of the method toward the exact solution (*e.g.*, convergence rate of 2 that we observed for linear bar elements in HW2).

Use of shape functions to approximate source term force



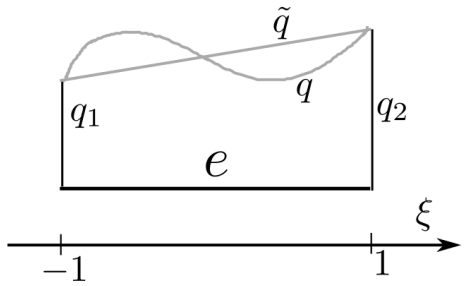
- Previously, we have used shape functions $\mathbf{N}^e$ to i) approximate solution; ii) obtain a relation between global and local coordinates (368); iii) and obtain a linear approximation of body force q for bar element (364).
- We want to derive the matrix (365) in local coordinate system.
- As we did in (364), q can be approximated by a linear $\tilde{q}$, using shape functions, as follows,

$$\tilde{q} = N_1 q_1 + N_2 q_2 = \mathbf{N}^e \begin{bmatrix} q_1 \\ q_2 \end{bmatrix} \text{ and}$$

- The source term vector is then approximated by,

$$\mathbf{f}_r^e = \int_e \mathbf{N}^{e\,\mathrm{T}}.q\,\mathrm{d}x \approx \int_e \mathbf{N}^e.\tilde{q}\,\mathrm{d}x = \{\int_e \mathbf{N}^{e\,\mathrm{T}}.\mathbf{N}^e\,\mathrm{d}x\} \begin{bmatrix} q_1 \\ q_2 \end{bmatrix} \quad \textit{cf.}\ (363\text{b})$$

Use of shape functions to approximate source term force



- We define the matrix,

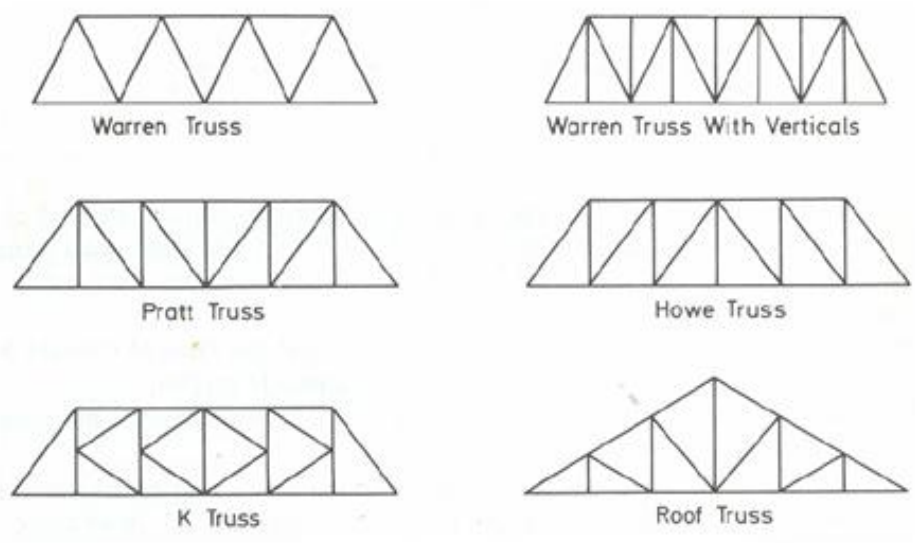
$$\mathbf{r}^e := \int_e \mathbf{N}^{e\,\mathrm{T}}.\mathbf{N}^e \,\mathrm{d}x$$

- Using the equations for shape functions in local coordinate system (367) and relation between $\mathrm{d}x$ and $\mathrm{d}\xi$ (372) we have,

$$\begin{aligned}\mathbf{r}^e &= \int_{-1}^{1} \begin{bmatrix}(1-\xi)/2\\(1+\xi)/2\end{bmatrix} \cdot \begin{bmatrix}(1-\xi)/2 & (1+\xi)/2\end{bmatrix} \{\frac{L^e}{2}\,\mathrm{d}\xi\} \\ &= \frac{L^e}{8}\int_{-1}^{1} \begin{bmatrix}1-2\xi+\xi^2 & 1-\xi^2\\ 1-\xi^2 & 1+2\xi+\xi^2\end{bmatrix} = \frac{L^e}{8}\begin{bmatrix}2+\frac{2}{3} & 2-\frac{2}{3}\\ 2-\frac{2}{3} & 2+\frac{2}{3}\end{bmatrix} \quad \Rightarrow\end{aligned}$$

$$\boxed{\mathbf{r}^e = \frac{L^e}{6}\begin{bmatrix}2 & 1\\ 1 & 2\end{bmatrix} \quad \text{and} \quad \mathbf{f}_r^e \approx \mathbf{r}^e \begin{bmatrix}q_1\\ q_2\end{bmatrix} \text{ exact for linear } q} \tag{378}$$

Trusses



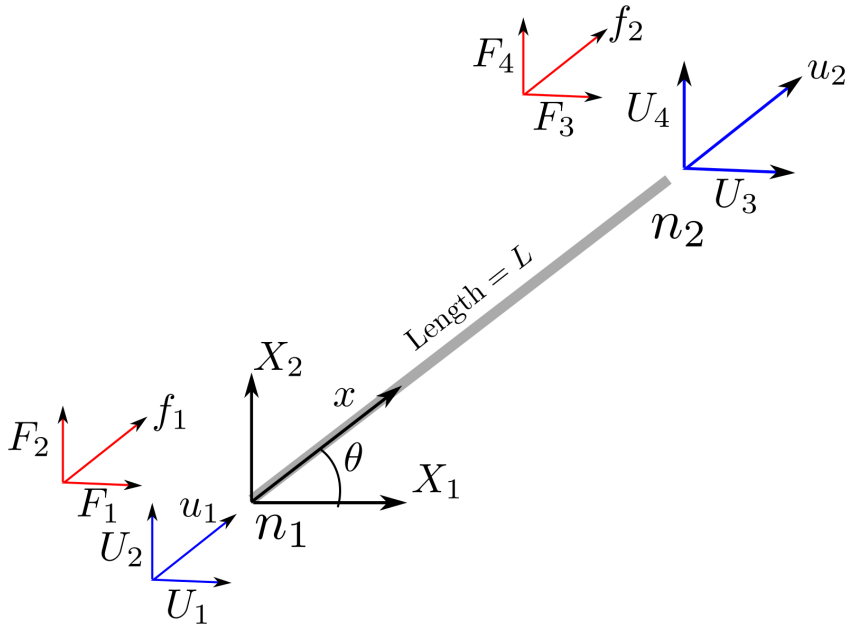
Types of simple Plane truss

- Trusses are 2D or 3D ensemble of bars.
- The main load transfer mechanism of these bars is axial force as the hinge connection at nodes prevent generation and transfer of moments.
- Although in bar elements we could have body force, in trusses we do not apply any type of load between the nodes (except the weight of bars themselves which may be neglected in many applications).
- Generally, top and bottom bars carry the moments and middle diagonal and vertical bars carry shear forces if we think of truss as a big bar.

Coordinate rotation for truss elements

We use lower- and upper-case to distinguish:

- Local (x) and global (X_1, X_2) coordinate systems.
- Local (u_1, u_2) and global (U_1, U_2, U_3, U_4) coordinate displacements.
- Local (f_1, f_2) and global (F_1, F_2, F_3, F_4) coordinate forces.
- Local $(\mathbf{k})$ and global $(\mathbf{K})$ coordinate stiffness matrices.

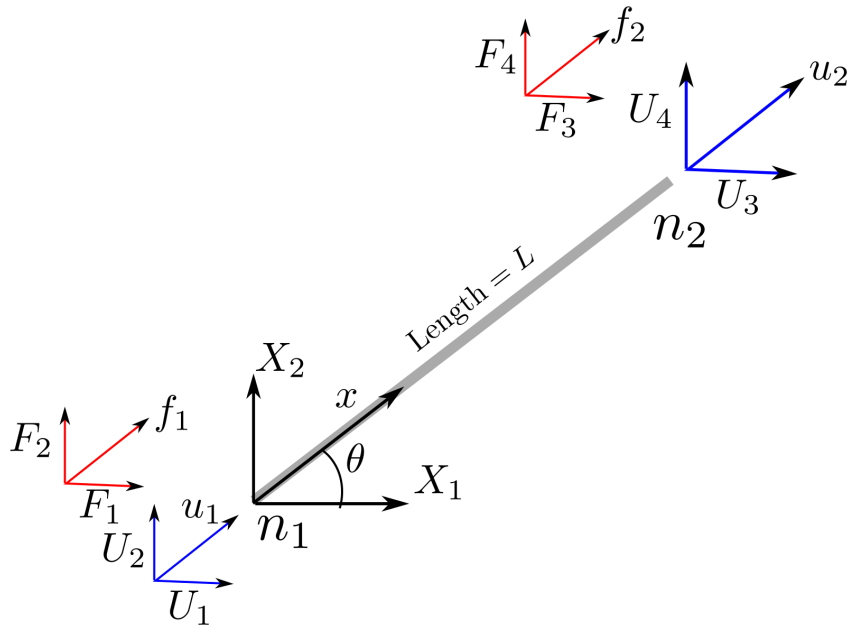


- Note that $\mathbf{K}$ is still a local stiffness matrix since it belongs to one element. This is opposed to our convention that we use $\mathbf{K}$ for system stiffness matrix and $\mathbf{K}$ is only used for element stiffness within this derivation.
- We already know the stiffness matrix between $\mathbf{u}$ and $\mathbf{f}$ in local coordinate system.
- We will use the transfer between $\mathbf{u} \longleftrightarrow \mathbf{U}$ and $\mathbf{f} \longleftrightarrow \mathbf{F}$ to obtain the stiffness $\mathbf{K}$ in global coordinate system.

Coordinate rotation for truss elements

- The stiffness matrix in local system is:

$$\mathbf{k} = \frac{AE}{L}\begin{bmatrix} 1 & -1 \\ -1 & 1 \end{bmatrix} \Rightarrow \mathbf{f} = \mathbf{k}\mathbf{u} \quad (379)$$



- Let $\mathbf{T_{uU}}$, a 2×4 matrix, be the transfer from $\mathbf{U}$ to $\mathbf{u}$:

$$\mathbf{u} = \mathbf{T_{uU}}\mathbf{U} \quad (380)$$

- Let $\mathbf{T_{Ff}}$, a 4×2 matrix, be the transfer from $\mathbf{f}$ to $\mathbf{F}$:

$$\mathbf{F} = \mathbf{T_{Ff}}\mathbf{f} \quad (381)$$

- Equations (379) and (380) yield:

$$\mathbf{f} = \{\mathbf{k}\mathbf{T_{uU}}\}\mathbf{U} \quad (382)$$

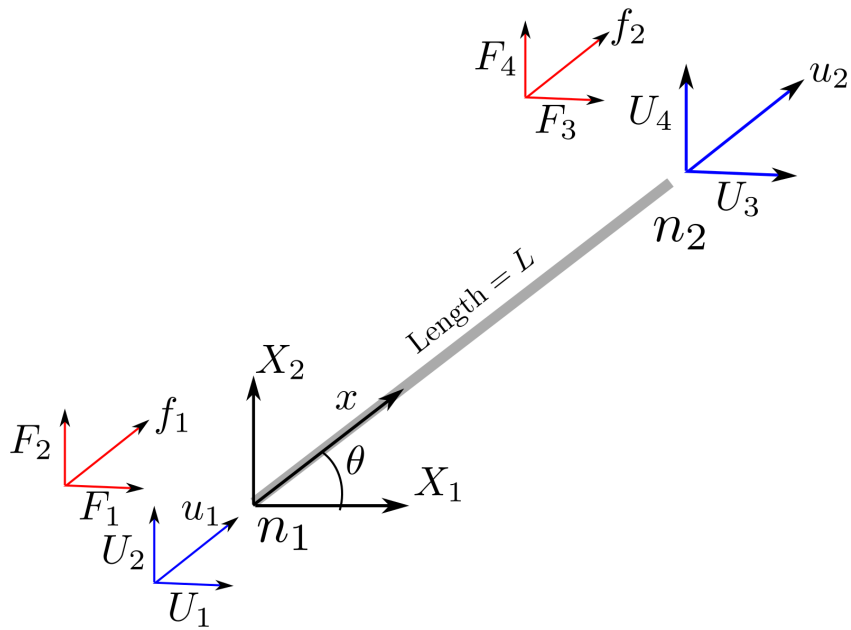
- We will later use the equation above to directly obtain the axial force in the bar (f_2) from $\mathbf{U}$.
- Equations (381) and (382) yield,

$$\mathbf{F} = \{\mathbf{T_{Ff}}\mathbf{k}\mathbf{T_{uU}}\}\mathbf{U} \quad \Rightarrow \quad \mathbf{K} = \mathbf{T_{Ff}}\mathbf{k}\mathbf{T_{uU}} \quad (383)$$

Coordinate rotation for truss elements

- Let us adopt the following shorthand notations:

$$c := \cos(\theta) \quad s := \sin(\theta) \tag{384}$$



- To obtain $\mathbf{T_{uU}}$ we observe that,

$$\left.\begin{aligned} u_1 &= cU_1 + sU_2 \\ u_2 &= cU_3 + sU_4 \end{aligned}\right\} \Rightarrow$$

$$\mathbf{T_{uU}} = \begin{bmatrix} c & s & 0 & 0 \\ 0 & 0 & c & s \end{bmatrix} \tag{385}$$

- First, to obtain $\mathbf{kT_{uU}}$ in (383) we observe (*cf.* (379), (385)),

$$\mathbf{T_{Ff}k} = \frac{AE}{L}\begin{bmatrix} 1 & -1 \\ -1 & 1 \end{bmatrix}\begin{bmatrix} c & s & 0 & 0 \\ 0 & 0 & c & s \end{bmatrix} \Rightarrow \begin{bmatrix} f_1 \\ f_2 \end{bmatrix} = \begin{bmatrix} c & s & -c & -s \\ -c & -s & c & s \end{bmatrix}\begin{bmatrix} U_1 \\ U_2 \\ U_3 \\ U_4 \end{bmatrix} \tag{386}$$

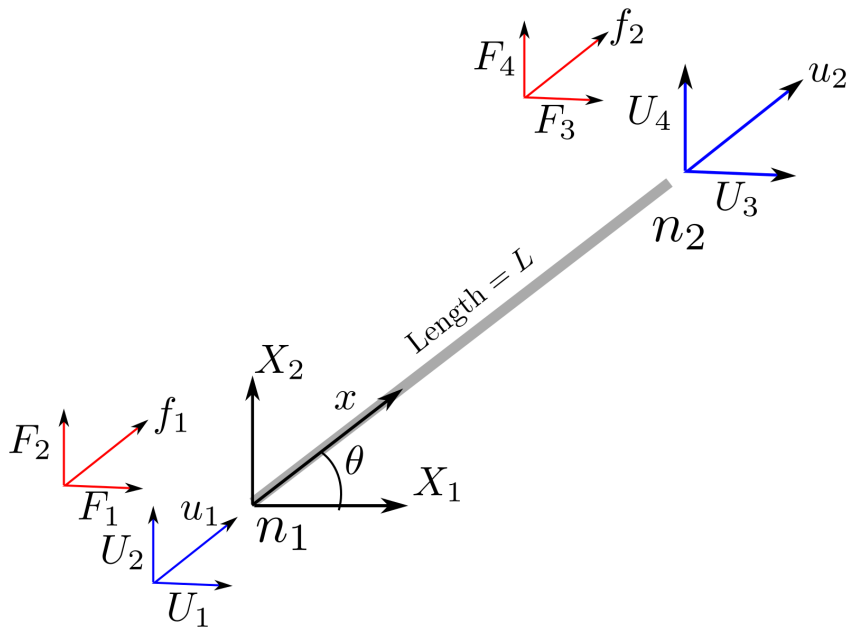
- Noting that f_2 corresponds to tensile axial force in the bar, which we denote by T we have,

$$\boxed{T = AE\left\{c(U_3 - U_1) + s(U_4 - U_2)\right\}} \tag{387}$$

Coordinate rotation for truss elements

- Subsequently, we obtain $\mathbf{T_{Ff}}$.
- We note that the forces f_1 and f_2 are axial and there are no forces normal to the bar to be broken down to F_1 to F_4.

$$\left.\begin{array}{ll} F_1 & = cf_1 \\ F_2 & = sf_1 \\ F_3 & = cf_2 \\ F_4 & = sf_2 \end{array}\right\} \quad \Rightarrow \mathbf{T_{Ff}} = \begin{bmatrix} c & 0 \\ s & 0 \\ 0 & c \\ 0 & s \end{bmatrix} \tag{388}$$



- We notice $\mathbf{T_{Ff}} = \mathbf{T_{uU}}^{\mathrm{T}}$ which we define as,

$$\mathbf{T} := \mathbf{T_{uU}} = \mathbf{T_{Ff}}^{\mathrm{T}} = \begin{bmatrix} c & s & 0 & 0 \\ 0 & 0 & c & s \end{bmatrix} \quad \Rightarrow \tag{389}$$

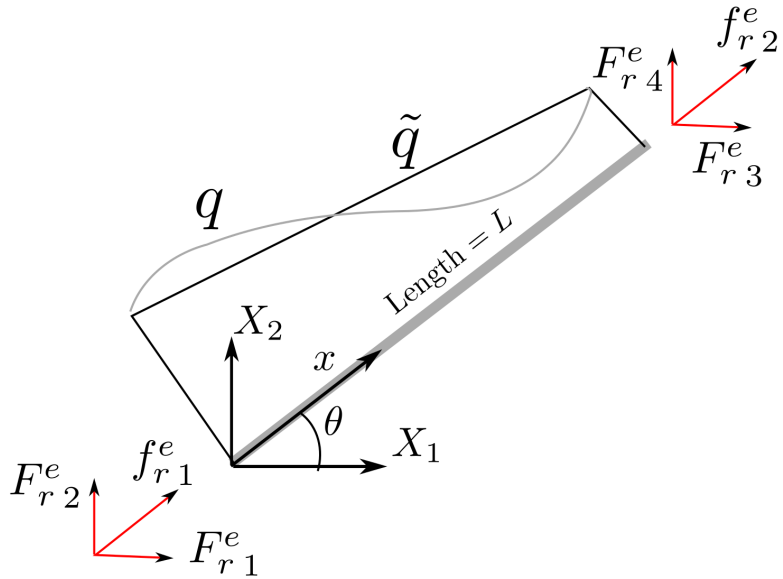
$$\mathbf{K} = \mathbf{T}^{\mathrm{T}}\mathbf{k}\mathbf{T} \Rightarrow \mathbf{K} = \frac{AE}{L}\left[\begin{array}{c|c} \mathbf{k}_b & -\mathbf{k}_b \\ \hline -\mathbf{k}_b & \mathbf{k}_b \end{array}\right], \text{ where } \mathbf{k}_b = \begin{bmatrix} c^2 & cs \\ cs & s^2 \end{bmatrix} \text{ that is}$$

$$\mathbf{K} = \frac{AE}{L}\begin{bmatrix} c^2 & cs & -c^2 & -cs \\ cs & s^2 & -cs & -s^2 \\ -c^2 & -cs & c^2 & cs \\ -cs & -s^2 & cs & s^2 \end{bmatrix} \tag{390}$$

Body force in rotated coordinate system

- While there is no body force exerted on truss members, it is interesting know how the body force we computed in local coordinate system transfers to global coordinate system.
- From (378) we have,

$$\mathbf{f}_r^e = \int_e \mathbf{N}^{e\,\mathrm{T}}.q\,\mathrm{d}x \approx \mathbf{r}^e \begin{bmatrix} q_1 \\ q_2 \end{bmatrix} \qquad (391)$$



- Similar to $\mathbf{F} = \mathbf{T}^{\mathrm{T}}\mathbf{U}$ we have,

$$\mathbf{F}_r^e = \mathbf{T}^{\mathrm{T}}\mathbf{f}_r^e$$

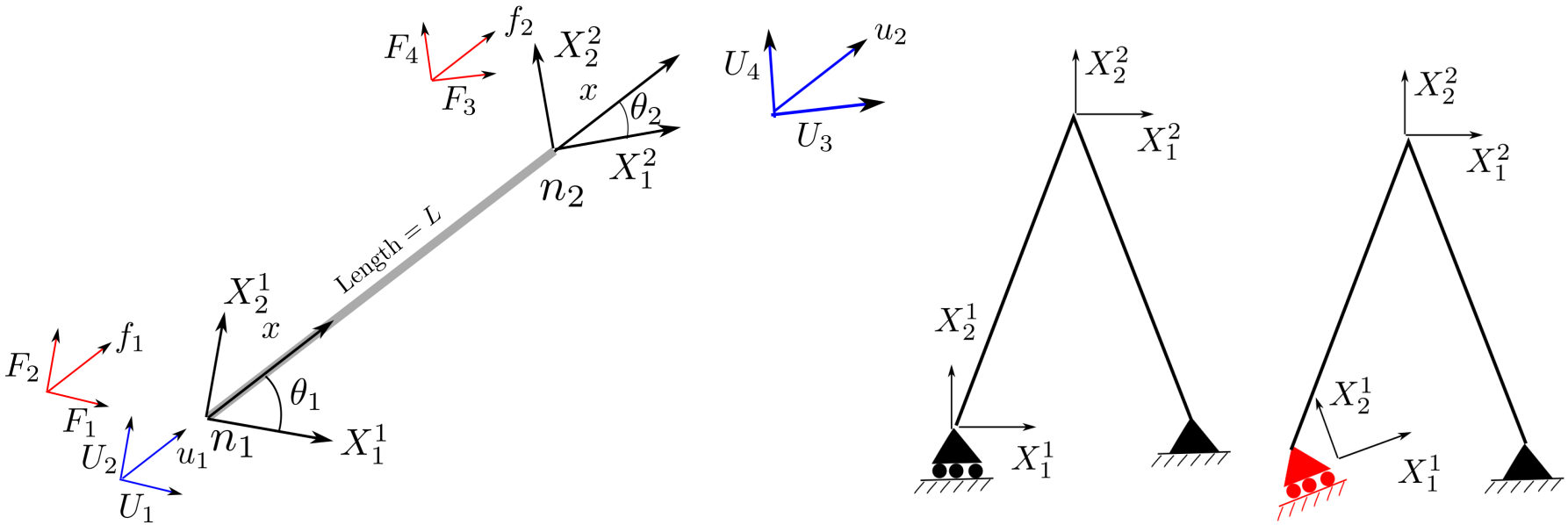
- Thus,

$$\mathbf{F}_r^e = \mathbf{T}^{\mathrm{T}} \int_e \mathbf{N}^{e\,\mathrm{T}}.q\,\mathrm{d}x \approx \mathbf{T}^{\mathrm{T}}\mathbf{r}^e \begin{bmatrix} q_1 \\ q_2 \end{bmatrix} \quad \text{that is}$$

$$\begin{bmatrix} F^e_{r\,1} \\ F^e_{r\,2} \\ F^e_{r\,3} \\ F^e_{r\,4} \end{bmatrix} = \begin{bmatrix} c & 0 \\ s & 0 \\ 0 & c \\ 0 & s \end{bmatrix} \int_e \begin{bmatrix} N_1^e \\ N_2^e \end{bmatrix} q\,\mathrm{d}x \approx \begin{bmatrix} c & 0 \\ s & 0 \\ 0 & c \\ 0 & s \end{bmatrix} \frac{L^e}{6} \begin{bmatrix} 2 & 1 \\ 1 & 2 \end{bmatrix} \begin{bmatrix} q_1 \\ q_2 \end{bmatrix} \quad \text{exact for linear } q \qquad (392)$$

- The transformation from local to global forces is general and applies to other forces, *e.g.*, $\mathbf{f}_N^e$, and other element types, *e.g.*, beams.

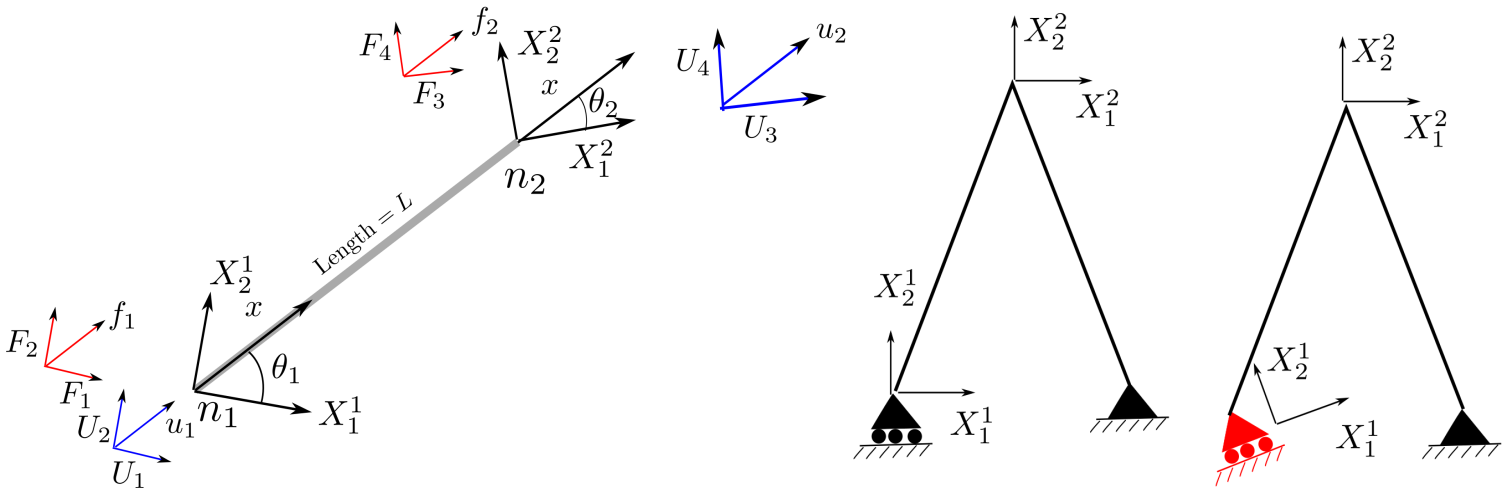
Truss element /two different coordinate systems



- In some instances we need to employ two different coordinate systems at the end points of a bar or in general coordinate system(s) that are not aligned with global coordinate system.
- For example the support highlighted in red in the right figure, do decouple displacement at the support and set the normal displacement to zero (Dirichlet BC) and tangential one free (Neumann BC) we need to employ the rotated coordinate system X_1^1, X_2^1.
- We have two different angles, θ_1 and θ_2. We define,

$$c_1 = \cos(\theta_1) \qquad s_1 = \sin(\theta_1)$$
$$c_2 = \cos(\theta_2) \qquad s_2 = \sin(\theta_2)$$

Truss element /two different coordinate systems



- As before $\mathbf{T} := \mathbf{T_{uU}} = \mathbf{T_{Ff}}$ and in this case is given by,

$$\mathbf{T} = \begin{bmatrix} c_1 & s_1 & 0 & 0 \\ 0 & 0 & c_2 & s_2 \end{bmatrix} \tag{393}$$

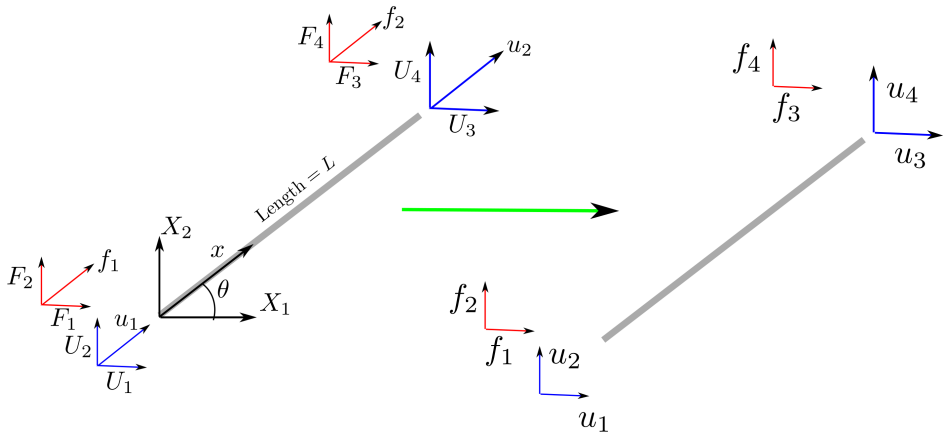
- Accordingly, from $\mathbf{K} = \mathbf{T}^{\mathrm{T}}\mathbf{k}\mathbf{T}$ we obtain,

$$\mathbf{K} = \begin{bmatrix} c_1^2 & c_1 s_1 & -c_1 c_2 & -c_1 s_2 \\ c_1 s_1 & s_1^2 & -c_2 s_1 & -s_1 s_2 \\ -c_1 c_2 & -c_2 s_1 & c_2^2 & c_2 s_2 \\ -c_1 s_2 & -s_1 s_2 & c_2 s_2 & s_2^2 \end{bmatrix} \tag{394}$$

- Finally the axial tensile force in the bar, which is the second line of $\mathbf{kT_{uU}} = \mathbf{kT}$ is (compare to one global coordinate in (387)):

$$\boxed{T = -c_1 U_1 - s_1 U_2 + c_2 U_3 + s_2 U_4} \tag{395}$$

Coordinate rotation for truss elements: Final notes



- While in our derivation we use upper-case for quantities in global coordinate system, and lower-case for coordinate system along the bar, we generally use upper-case for global quantities (stiffness, force vector, *etc.*) and lower-case for element level quantities.
- Now that we have obtained the local stiffness matrix in rotated coordinate system we will use lower-case for its stiffness matrix, and corresponding forces and displacements.
- Truss element is an example of an element which has more than one dof. per node.
- When there are more than one dof per node, the element node numbering is:

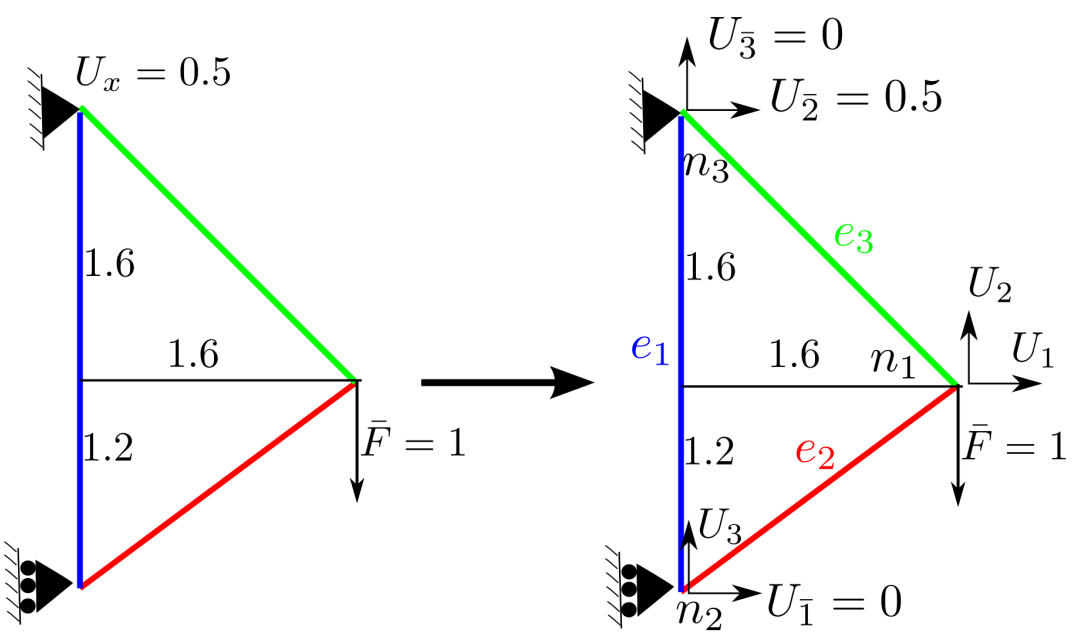
$$\text{element order of dofs} \;=\; \begin{bmatrix} (\text{dof 1 of } n_1) & \cdots & (\text{dof } n_{\text{dpn}} \text{ of } n_1) \\ \vdots & \ddots & \vdots \\ (\text{dof 1 of } n_{n_{\text{n}}^e}) & \cdots & (\text{dof } n_{\text{dpn}} \text{ of } n_{n_{\text{n}}^e}) \end{bmatrix} \tag{396}$$

where we cover rows and then columns. As before,

$n_{\text{dpn}} =$ number of dof per node $\qquad n_{\text{n}}^e =$ number of nodes per elements

- This explains the order of dof we used for truss element (u_1, u_2, u_3, u_4 in the right figure).

Truss Example

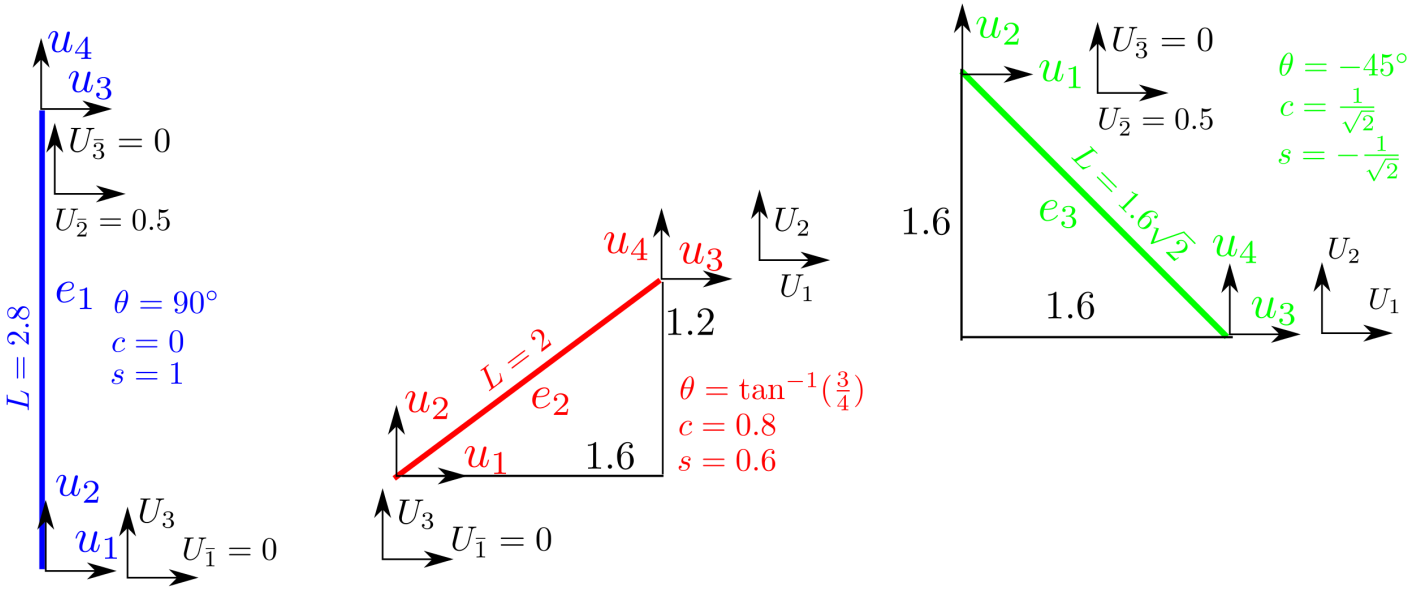


- In order to solve the truss on the left, we need to number nodes and elements.
- Each node has two dofs (x and y displacements).
- Ordering of dofs is as follows:
  - Start from n_1 to n_3.
  - For each node increment free as well as prescribed dof count based on node conditions.

- Final element and node numbering is shown in the figure on the right.
- Initial node and element numbering is arbitrary in a sense that all result in the same physical solution.
- Node and dof numbering affects the band width of the stiffness matrix.
- In general, FE software optimizes such numbering to obtain the minimum stiffness matrix band width. This minimizes the solution cost.
- In this example material and section properties are:

$$A^{e_1} = 1 \qquad E^{e_1} = 1$$
$$A^{e_2} = 1 \qquad E^{e_2} = 1$$
$$A^{e_3} = 1 \qquad E^{e_3} = 1$$

Truss Example



- Table below summarizes parameters for each element:

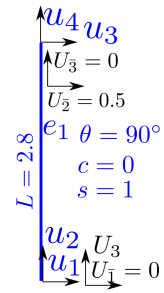
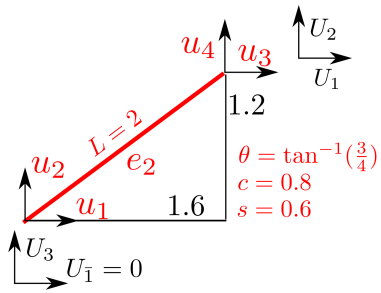
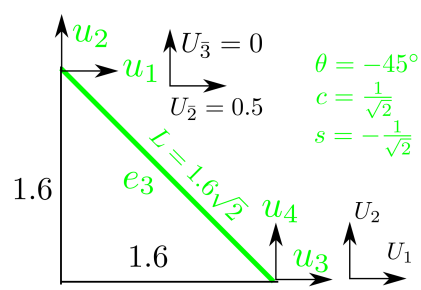
e	L^e	θ	c	s	$\mathbf{M}_t^e$
e_1	2.8	$90°$	0	1	$[\bar{1}\ \ 3\ \ \bar{2}\ \ \bar{3}]$
e_2	2	$\tan^{-1}(\frac{3}{4})$	0.8	0.6	$[\bar{1}\ \ 3\ \ 1\ \ 2]$
e_3	$1.6\sqrt{2}$	$-45°$	$\frac{1}{\sqrt{2}}$	$-\frac{1}{\sqrt{2}}$	$[\bar{2}\ \ \bar{3}\ \ 1\ \ 2]$

- Local stiffness matrices are given by (390):

$$\mathbf{k}^e = \frac{AE}{L}\left[\begin{array}{c|c} \mathbf{k}_b & -\mathbf{k}_b \\ \hline -\mathbf{k}_b & \mathbf{k}_b \end{array}\right],\ \mathbf{k}_b = \begin{bmatrix} c^2 & cs \\ cs & s^2 \end{bmatrix} : \mathbf{k}^e = \frac{AE}{L}\begin{bmatrix} c^2 & cs & -c^2 & -cs \\ cs & s^2 & -cs & -s^2 \\ -c^2 & -cs & c^2 & cs \\ cs & s^2 & -cs & -s^2 \end{bmatrix}$$

- As mentioned for trusses generally $\mathbf{f}_r^e = 0$ (no body force), similar to bars we lump natural BC into nodal forces, and finally $\mathbf{f}_D^e = \mathbf{k}^e\mathbf{a}^e$.

Truss example: Assembly of global system

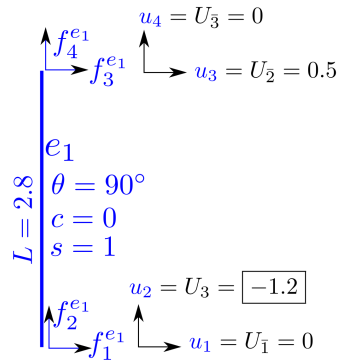
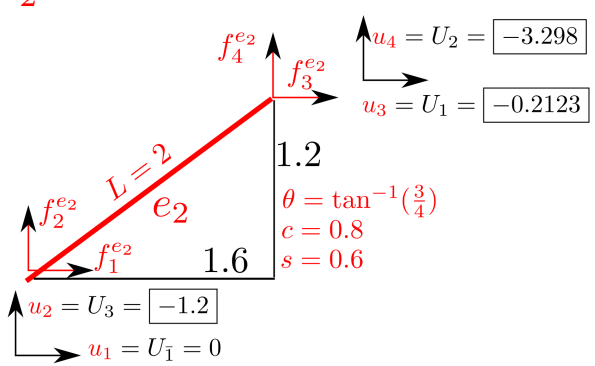
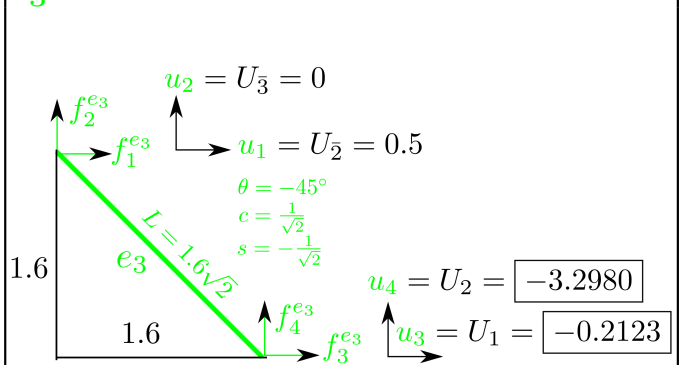
e	e_1	e_2	e_3
			
$\mathbf{k}^e$	$\mathbf{k}^{e1} = \frac{(1)(1)}{2.8}\begin{bmatrix} 0&0&0&0 \\ 0&1&0&-1 \\ 0&0&0&0 \\ 0&-1&0&1 \end{bmatrix}$ $= \begin{array}{c} \\ \bar{1} \\ 3 \\ \bar{2} \\ \bar{3} \end{array}\begin{array}{c} \begin{array}{cccc} \bar{1} & 3 & \bar{2} & \bar{3} \end{array} \\ \begin{bmatrix} 0&0&0&0 \\ 0&0.3571&0&-0.3571 \\ 0&0&0&0 \\ 0&-0.3571&0&0.3571 \end{bmatrix} \end{array}$	$\mathbf{k}^{e2} = \frac{(1)(1)}{2}\begin{bmatrix} 0.64&0.48&-0.64&-0.48 \\ 0.48&0.36&-0.48&-0.36 \\ -0.64&-0.48&0.64&0.48 \\ -0.48&-0.36&0.48&0.36 \end{bmatrix}$ $= \begin{array}{c} \\ \bar{1} \\ 3 \\ 1 \\ 2 \end{array}\begin{array}{c} \begin{array}{cccc} \bar{1} & 3 & 1 & 2 \end{array} \\ \begin{bmatrix} 0.32&0.24&-0.32&-0.24 \\ 0.24&0.18&-0.24&-0.18 \\ -0.32&-0.24&0.32&0.24 \\ -0.24&-0.18&0.24&0.18 \end{bmatrix} \end{array}$	$\mathbf{k}^{e3} = \frac{(1)(1)}{1.6\sqrt{2}}\begin{bmatrix} 0.5&-0.5&-0.5&0.5 \\ -0.5&0.5&0.5&-0.5 \\ -0.5&0.5&0.5&-0.5 \\ 0.5&-0.5&-0.5&0.5 \end{bmatrix}$ $= \begin{array}{c} \\ \bar{2} \\ \bar{3} \\ 1 \\ 2 \end{array}\begin{array}{c} \begin{array}{cccc} \bar{2} & \bar{3} & 1 & 2 \end{array} \\ \begin{bmatrix} 0.221&-0.221&-0.221&0.221 \\ -0.221&0.221&0.221&-0.221 \\ -0.221&0.221&0.221&-0.221 \\ 0.221&-0.221&-0.221&0.221 \end{bmatrix} \end{array}$
$\mathbf{f}_D^e$	$\mathbf{k}^{e1}\mathbf{a}_1^e = \begin{bmatrix} 0&0&0&0 \\ 0&0.3571&0&-0.3571 \\ 0&0&0&0 \\ 0&-0.3571&0&0.3571 \end{bmatrix}\begin{bmatrix} 0 \\ \textcircled{0} \\ 0.5 \\ 0 \end{bmatrix} = \begin{array}{c} \\ \bar{1} \\ 3 \\ \bar{2} \\ \bar{3} \end{array}\begin{bmatrix} 0 \\ 0 \\ 0 \\ 0 \end{bmatrix}$	$\mathbf{k}^{e2}\mathbf{a}_2^e = \begin{bmatrix} 0.32&0.24&-0.32&-0.24 \\ 0.24&0.18&-0.24&-0.18 \\ -0.32&-0.24&0.32&0.24 \\ -0.24&-0.18&0.24&0.18 \end{bmatrix}\begin{bmatrix} 0 \\ \textcircled{0} \\ \textcircled{0} \\ \textcircled{0} \end{bmatrix} = \begin{array}{c} \\ \bar{1} \\ 3 \\ 1 \\ 2 \end{array}\begin{bmatrix} 0 \\ 0 \\ 0 \\ 0 \end{bmatrix}$	$\mathbf{k}^{e3}\mathbf{a}_3^e = \begin{bmatrix} 0.221&-0.221&-0.221&0.221 \\ -0.221&0.221&0.221&-0.221 \\ -0.221&0.221&0.221&-0.221 \\ 0.221&-0.221&-0.221&0.221 \end{bmatrix}\begin{bmatrix} 0.5 \\ 0 \\ \textcircled{0} \\ \textcircled{0} \end{bmatrix} = \begin{array}{c} \\ \bar{2} \\ \bar{3} \\ 1 \\ 2 \end{array}\begin{bmatrix} 0.1105 \\ -0.1105 \\ -0.1105 \\ 0.1105 \end{bmatrix}$
$\mathbf{f}_e^e$	$\mathbf{f}_e^{e1} = \mathbf{f}_r^{e1} + \mathbf{f}_N^{e1} - \mathbf{f}_D^{e1} = \begin{array}{c} \\ \bar{1} \\ 3 \\ \bar{2} \\ \bar{3} \end{array}\begin{bmatrix} 0 \\ 0 \\ 0 \\ 0 \end{bmatrix}$	$\mathbf{f}_e^{e2} = \mathbf{f}_r^{e2} + \mathbf{f}_N^{e2} - \mathbf{f}_D^{e2} = \begin{array}{c} \\ \bar{1} \\ 3 \\ 1 \\ 2 \end{array}\begin{bmatrix} 0 \\ 0 \\ 0 \\ 0 \end{bmatrix}$	$\mathbf{f}_e^{e3} = \mathbf{f}_r^{e3} + \mathbf{f}_N^{e3} - \mathbf{f}_D^{e3} = \begin{array}{c} \\ \bar{2} \\ \bar{3} \\ 1 \\ 2 \end{array}\begin{bmatrix} -0.1105 \\ 0.1105 \\ 0.1105 \\ -0.1105 \end{bmatrix}$

$$\mathbf{K} = \begin{bmatrix} 0.32+0.221 & 0.24-0.221 & -0.24 \\ 0.24-0.221 & 0.18+0.221 & -0.18 \\ -0.24 & -0.18 & 0.3571+0.18 \end{bmatrix} = \begin{bmatrix} 0.5410 & 0.019 & -0.24 \\ 0.019 & 0.401 & -0.18 \\ -0.24 & -0.18 & 0.5371 \end{bmatrix}$$

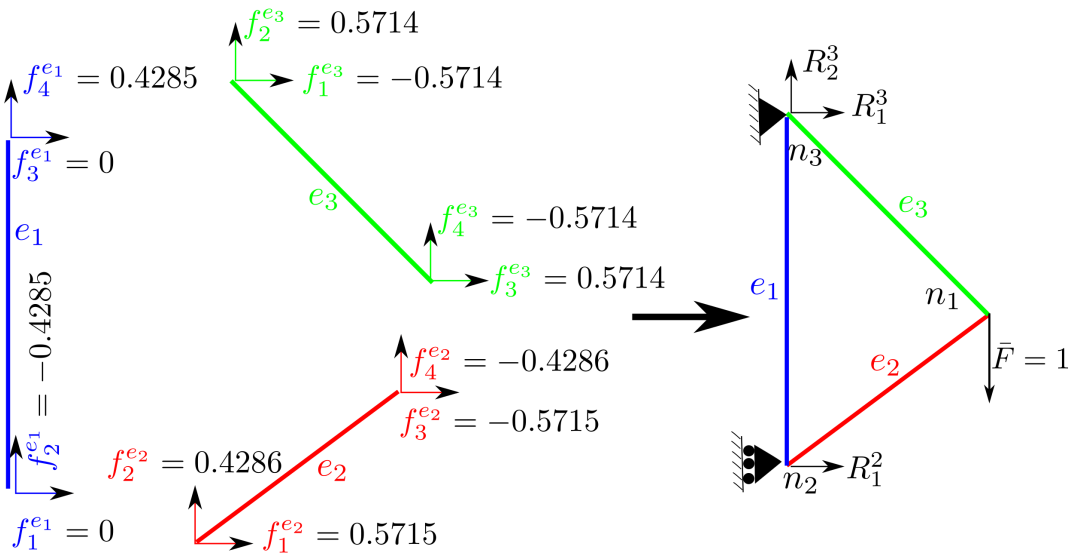
$$\mathbf{F} = \mathbf{F}_N + \mathbf{F}_e = \begin{bmatrix} 0 \\ -1 \\ 0 \end{bmatrix} + \begin{bmatrix} 0.1105 \\ -0.1105 \\ 0 \end{bmatrix} = \begin{bmatrix} 0.1105 \\ -1.1105 \\ 0 \end{bmatrix} \Rightarrow \quad \boxed{\mathbf{U} = \mathbf{K}^{-1}\mathbf{F} = \begin{bmatrix} U_1 \\ U_2 \\ U_3 \end{bmatrix} = \begin{bmatrix} -0.2123 \\ -3.2980 \\ -1.200 \end{bmatrix}}$$

- Numbers encircled in the computation of essential BC force are displacements corresponding to free dofs. As mentioned before, in reality we do not consider them in computation of this force, but in hand calculation we just put zero for those values.

Truss Example: Axial force and element local forces

e	$e1$	$e2$	$e3$
			
$\mathbf{u}^e$	$\begin{bmatrix} 0 \\ -1.2 \\ 0.5 \\ 0 \end{bmatrix}$	$\begin{bmatrix} 0 \\ -1.2 \\ -0.2123 \\ -3.2980 \end{bmatrix}$	$\begin{bmatrix} 0.5 \\ 0 \\ -0.2123 \\ -3.2980 \end{bmatrix}$
T^e	$T^{e1} = \frac{1\times1}{2.8}\{0 \times (0.5 - 0) + 1 \times (0 + 1.2)\} = 0.4286$	$T^{e2} = \frac{1\times1}{2}\{0.8 \times (-0.2123 - 0) + 0.6 \times (-3.2928 + 1.2)\} = -0.7128$	$T^{e3} = \frac{1\times1}{1.6\sqrt{2}}\{\frac{1}{\sqrt{2}} \times (-0.2123 - 0.5) - \frac{1}{\sqrt{2}} \times (-3.2928 - 0)\} = 0.8064$
$-f^e$	$\mathbf{k}^{e1}\mathbf{a}_1^e = \begin{bmatrix} 0 & 0 & 0 & 0 \\ 0 & 0.357 & 0 & -0.357 \\ 0 & 0 & 0 & 0 \\ 0 & -0.357 & 0 & 0.357 \end{bmatrix} \begin{bmatrix} 0 \\ -1.2 \\ 0.5 \\ 0 \end{bmatrix} = \begin{matrix} \bar{1} \\ 3 \\ \bar{2} \\ \bar{3} \end{matrix} \begin{bmatrix} 0 \\ -0.4285 \\ 0 \\ 0.4285 \end{bmatrix}$	$\mathbf{k}^{e2}\mathbf{a}_2^e = \begin{bmatrix} 0.32 & 0.24 & -0.32 & -0.24 \\ 0.24 & 0.18 & -0.24 & -0.18 \\ -0.32 & -0.24 & 0.32 & 0.24 \\ -0.24 & -0.18 & 0.24 & 0.18 \end{bmatrix} \begin{bmatrix} 0 \\ -1.2 \\ -0.2123 \\ -3.2980 \end{bmatrix} = \begin{matrix} \bar{1} \\ 3 \\ 1 \\ 2 \end{matrix} \begin{bmatrix} 0.5715 \\ 0.4286 \\ -0.5715 \\ -0.4286 \end{bmatrix}$	$\mathbf{k}^{e3}\mathbf{a}_3^e = \begin{bmatrix} 0.221 & -0.221 & -0.221 & 0.221 \\ -0.221 & 0.221 & 0.221 & -0.221 \\ -0.221 & 0.221 & 0.221 & -0.221 \\ 0.221 & -0.221 & -0.221 & 0.221 \end{bmatrix} \begin{bmatrix} 0.5 \\ 0 \\ -0.2123 \\ -3.2980 \end{bmatrix} = \begin{matrix} \bar{2} \\ \bar{3} \\ 1 \\ 2 \end{matrix} \begin{bmatrix} -0.5714 \\ 0.5714 \\ 0.5714 \\ -0.5714 \end{bmatrix}$

Truss Example: Reaction Forces



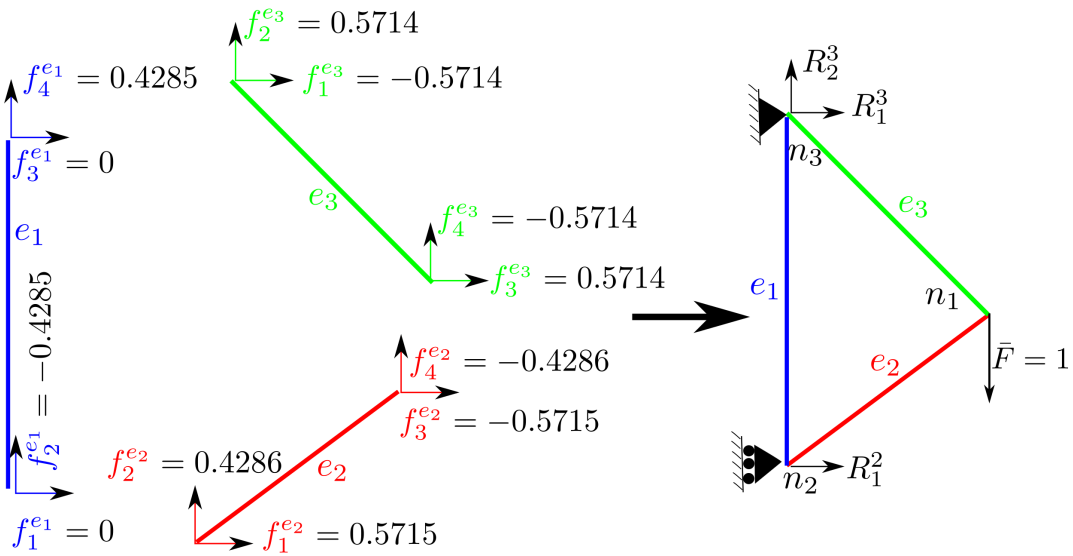
- First, we compute reaction forces by adding up forces from individual elements that contribute to reaction forces:

$$R_1^2 = f_1^{e_1} + f_1^{e_2} = 0 + 0.5715 = 0.5715 \tag{397a}$$

$$R_1^3 = f_3^{e_1} + f_1^{e_3} = 0 + -0.5714 = -0.5714 \tag{397b}$$

$$R_2^3 = f_4^{e_1} + f_2^{e_3} = 0.4285 + 0.5714 = 0.9999 \tag{397c}$$

Truss Example: verification of forces at free dofs



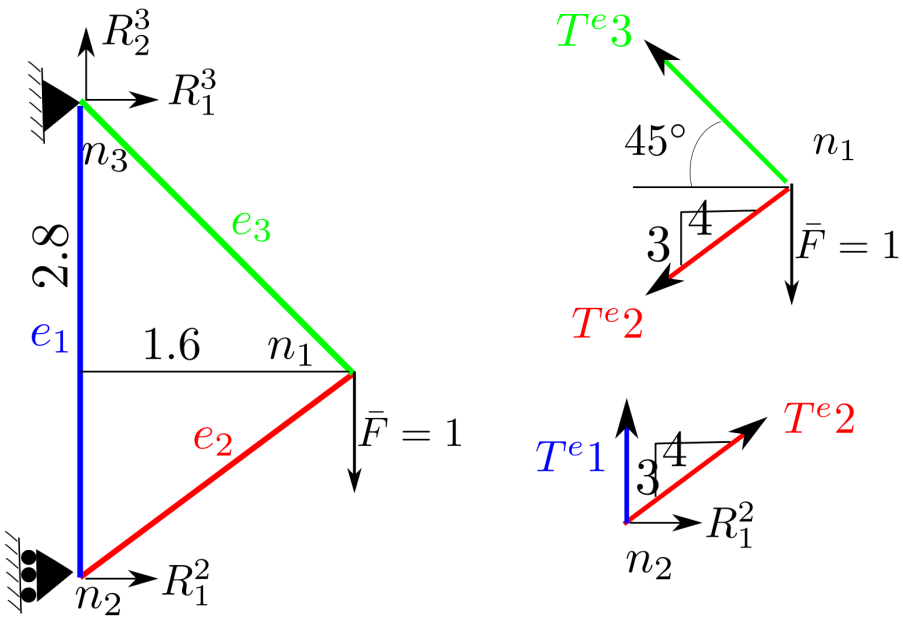
- Also, if we want to double-check our calculations on **free dofs**. This step is not needed and it may be done as a verification for hand calculations:

$$F_1^1 = f_3^{e_2} + f_3^{e_3} = -0.5715 + 0.5714 = -0.0001 \tag{398a}$$

$$F_2^1 = f_4^{e_2} + f_4^{e_3} = -0.4286 + -0.5714 = -1 = \bar{F} \tag{398b}$$

$$R_2^2 = f_2^{e_1} + f_2^{e_2} = -0.4285 + 0.4286 = 0.0001 \tag{398c}$$

Truss Example: Direct solution method



- Since this is a statically determinate structure, we can easily solve the forces and verify our FEM forces.

$$\Sigma F_2 = 0 \quad \Rightarrow \quad R_2^3 - \bar{F} = 0 \quad \Rightarrow \quad R_2^3 = 1 \tag{399a}$$

$$\Sigma M_{n_3} = 0 \quad \Rightarrow \quad 2.8R_1^2 - 1.6\bar{F} = 0 \quad \Rightarrow \quad R_1^2 = \frac{4}{7} = 0.5714 \tag{399b}$$

$$\Sigma F_1 = 0 \quad \Rightarrow \quad R_1^2 + R_1^3 = 0 \quad \Rightarrow \quad R_1^3 = -\frac{4}{7} = -0.5714 \tag{399c}$$

$$\Sigma F_1 = 0(@n_2) \quad \Rightarrow \quad R_1^2 + \frac{4}{5}T^e2 = 0 \quad \Rightarrow \quad T^e2 = -\frac{5}{7} = -0.7143 \tag{399d}$$

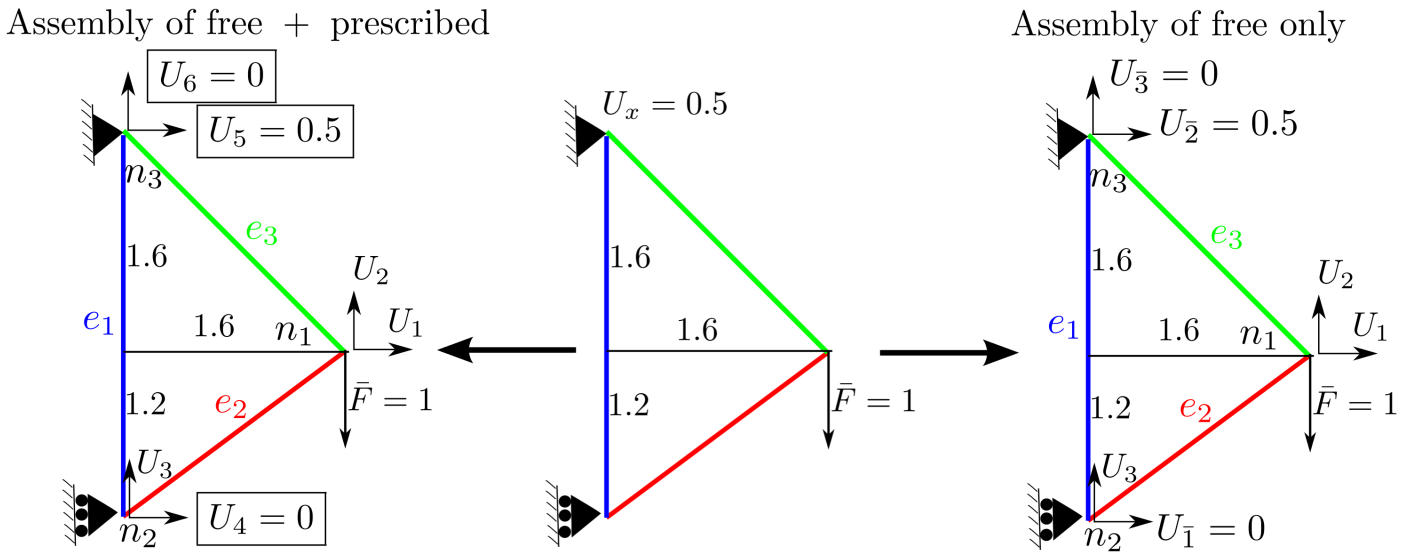
$$\Sigma F_2 = 0(@n_2) \quad \Rightarrow \quad T^e1 + \frac{3}{5}T^e2 = 0 \quad \Rightarrow \quad T^e1 = \frac{3}{7} = 0.4286 \tag{399e}$$

$$\Sigma F_1 = 0(@n_1) \quad \Rightarrow \quad -\frac{4}{5}T^e2 - \frac{1}{\sqrt{2}}T^e3 = 0 \quad \Rightarrow \quad T^e3 = \frac{4}{7}\sqrt{2} = 0.8081 \tag{399f}$$

Truss Example: Observations

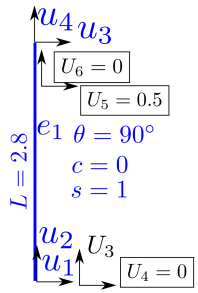
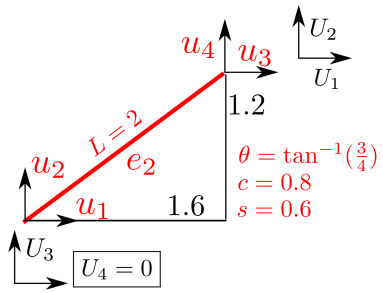
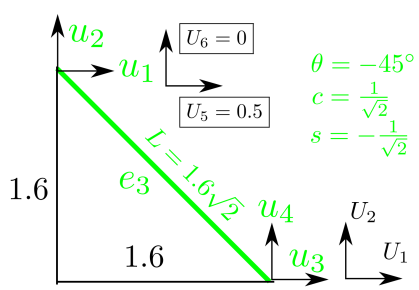
- The fact that our free dof forces in (398) are the values enforced is a necessary (but not sufficient) check on the correctness of our FEM solutions.
- We observe that exact (direct solutions in (399)) match our FEM solution in the last table and (397).
- The reason FEM and exact solutions match is that FEM shape functions (linear displacement within each bar) can capture the exact solution. In general, whenever, FEM approximate solution space can capture exact solution, FEM recovers the exact solution.
- Small error between FEM and direct method or about 0.0001 errors in some reaction and sum of forces in FEM method are finite numerical precision error which are different from discretization errors caused by approximating an infinite solution space by FEM shape functions. The former is caused by working with finite number of digits in our calculations.

Assembly of free + prescribed dofs vs. free only



- All we covered so far was the **assembly of free dof only**.
- We can assemble all dofs (free + prescribed) as shown in figure on the left.
- The numbering of dof when assembling free + prescribed dof is exactly like before with the difference that **we first number free dof followed by prescribed dof** as shown in the figure. For each group (f & p) we start from node n_1 to n_{n_n}.

Truss example: Assembly of global system $(f + p)$

e	e_1	e_2	e_3
			
$\mathbf{k}^e$	$\mathbf{k}^{e_1} = \frac{(1)(1)}{2.8}$ $\begin{bmatrix} 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & -1 \\ 0 & 0 & 0 & 0 \\ 0 & -1 & 0 & 1 \end{bmatrix}$ $= \begin{array}{c} \\ 4 \\ 3 \\ 5 \\ 6 \end{array} \begin{array}{c} \begin{array}{cccc} 4 & 3 & 5 & 6 \end{array} \\ \begin{bmatrix} 0 & 0 & 0 & 0 \\ 0 & 0.3571 & 0 & -0.3571 \\ 0 & 0 & 0 & 0 \\ 0 & -0.3571 & 0 & 0.3571 \end{bmatrix} \end{array}$	$\mathbf{k}^{e_2} = \frac{(1)(1)}{2}$ $\begin{bmatrix} 0.64 & 0.48 & -0.64 & -0.48 \\ 0.48 & 0.36 & -0.48 & -0.36 \\ -0.64 & -0.48 & 0.64 & 0.48 \\ -0.48 & -0.36 & 0.48 & 0.36 \end{bmatrix}$ $= \begin{array}{c} \\ 4 \\ 3 \\ 1 \\ 2 \end{array} \begin{array}{c} \begin{array}{cccc} 4 & 3 & 1 & 2 \end{array} \\ \begin{bmatrix} 0.32 & 0.24 & -0.32 & -0.24 \\ 0.24 & 0.18 & -0.24 & -0.18 \\ -0.32 & -0.24 & 0.32 & 0.24 \\ -0.24 & -0.18 & 0.24 & 0.18 \end{bmatrix} \end{array}$	$\mathbf{k}^{e_3} = \frac{(1)(1)}{1.6\sqrt{2}}$ $\begin{bmatrix} 0.5 & -0.5 & -0.5 & 0.5 \\ -0.5 & 0.5 & 0.5 & -0.5 \\ -0.5 & 0.5 & 0.5 & -0.5 \\ 0.5 & -0.5 & -0.5 & 0.5 \end{bmatrix}$ $= \begin{array}{c} \\ 5 \\ 6 \\ 1 \\ 2 \end{array} \begin{array}{c} \begin{array}{cccc} 5 & 6 & 1 & 2 \end{array} \\ \begin{bmatrix} 0.221 & -0.221 & -0.221 & 0.221 \\ -0.221 & 0.221 & 0.221 & -0.221 \\ -0.221 & 0.221 & 0.221 & -0.221 \\ 0.221 & -0.221 & -0.221 & 0.221 \end{bmatrix} \end{array}$

- Global stiffness matrix:

$$\mathbf{K} = \begin{bmatrix} 0.32+0.221 & 0.24-0.221 & -0.24 & -0.32 & -0.221 & 0.221 \\ 0.24-0.221 & 0.18+0.221 & -0.18 & -0.24 & 0.221 & -0.221 \\ -0.24 & -0.18 & 0.3571+0.18 & 0+0.24 & 0 & -0.3571 \\ -0.32 & -0.24 & 0+0.24 & 0+0.32 & 0 & 0 \\ -0.221 & 0.221 & 0 & 0 & 0+0.221 & 0-0.221 \\ 0.221 & -0.221 & -0.3571 & 0 & 0-0.221 & 0.3571+0.221 \end{bmatrix}$$

$$= \begin{bmatrix} 0.5410 & 0.019 & -0.24 & -0.32 & -0.221 & 0.221 \\ 0.019 & 0.401 & -0.18 & -0.24 & 0.221 & -0.221 \\ -0.24 & -0.18 & 0.5371 & 0.24 & 0 & -0.3571 \\ -0.32 & -0.24 & 0.24 & 0.32 & 0 & 0 \\ -0.221 & 0.221 & 0 & 0 & 0.221 & -0.221 \\ 0.221 & -0.221 & -0.3571 & 0 & -0.221 & 0.5781 \end{bmatrix}$$

- Global force vector. The element vectors that would assemble to $\mathbf{F}_e$ are $\mathbf{f}_e^e = \mathbf{f}_r^e + \mathbf{f}_N^e$ and we **do not include $-\mathbf{f}_D^e$ in it as it would be directly taken care of subsequently**. Since all $= \mathbf{f}_r^e$ and $\mathbf{f}_N^e$ are zero, $\mathbf{F}_e$ is identically zero.

$$\mathbf{F} = \mathbf{F}_N + \mathbf{F}_e = \begin{bmatrix} 0 \\ -1 \\ 0 \\ R_1^2 \\ R_1^3 \\ R_2^3 \end{bmatrix} + \begin{bmatrix} 0 \\ 0 \\ 0 \\ 0 \\ 0 \\ 0 \end{bmatrix} = \begin{bmatrix} 0 \\ -1 \\ 0 \\ R_1^2 \\ R_1^3 \\ R_2^3 \end{bmatrix}$$

Truss example: Solution of global system $(f + p)$

- The global system is,

$\mathbf{KU} = \mathbf{F}$ where

$$\mathbf{K} = \left[\begin{array}{ccc|ccc} 0.5410 & 0.019 & -0.24 & -0.32 & -0.221 & 0.221 \\ 0.019 & 0.401 & -0.18 & -0.24 & 0.221 & -0.221 \\ -0.24 & -0.18 & 0.5371 & 0.24 & 0 & -0.3571 \\ \hline -0.32 & -0.24 & 0.24 & 0.32 & 0 & 0 \\ -0.221 & 0.221 & 0 & 0 & 0.221 & -0.221 \\ 0.221 & -0.221 & -0.3571 & 0 & -0.221 & 0.5781 \end{array}\right] = \left[\begin{array}{c|c} \mathbf{K}_{ff} & \mathbf{K}_{fp} \\ \hline \mathbf{K}_{pf} & \mathbf{K}_{pp} \end{array}\right]$$

$$\mathbf{F} = \begin{bmatrix} F_1 \\ F_2 \\ F_3 \\ F_4 \\ F_6 \\ F_6 \end{bmatrix} \begin{bmatrix} 0 \\ -1 \\ 0 \\ R_1^2 \\ R_1^3 \\ R_2^3 \end{bmatrix} = \left[\begin{array}{c} \mathbf{F}_f \\ \hline \mathbf{F}_p \end{array}\right]$$

- The unknown quantities $\mathbf{F}_p$ and $\mathbf{U}_f$ are highlighted.

Truss example: Solution of global system $(f + p)$

- To solve the system in previous slid, $\mathbf{KU} = \mathbf{F}$, we observe,

$$\mathbf{F}_f = \mathbf{K}_{ff}\mathbf{U}_f + \mathbf{K}_{fp}\mathbf{U}_p \tag{400a}$$
$$\mathbf{F}_p = \mathbf{K}_{pf}\mathbf{U}_f + \mathbf{K}_{pp}\mathbf{U}_p \tag{400b}$$

- Sizes of these matrices and vectors are:

$$\mathbf{F}_f : n_\mathrm{f} \times 1 \qquad \mathbf{F}_p : n_\mathrm{p} \times 1$$
$$\mathbf{K}_{ff} : n_\mathrm{f} \times n_\mathrm{f} \qquad \mathbf{K}_{fp} : n_\mathrm{f} \times n_\mathrm{p} \qquad \mathbf{K}_{pf} : n_\mathrm{p} \times n_\mathrm{f} \qquad \mathbf{K}_{pp} : n_\mathrm{p} \times n_\mathrm{p}$$
$$\mathbf{U}_f : n_\mathrm{f} \times 1 \qquad \mathbf{U}_p : n_\mathrm{p} \times 1$$
$$n_\mathrm{f} = \#\underline{\text{free}} \text{ dof} \qquad n_\mathrm{p} = \#\ \underline{\text{prescibed}} \text{ dof}$$

- In this particular problem $n_\mathrm{f} = 3$ and $n_\mathrm{p} = 3$ but clearly they can be distinct.
- To solve the unknowns $\mathbf{U}_f$ and $\mathbf{F}_p$ in (400), we first solve for $\mathbf{U}_f$ in (400a). Subsequently, we plug $\mathbf{U}_f$ in (400b):

$$\mathbf{U}_f = \mathbf{K}_{ff}{}^{-1}\left(\mathbf{F}_f - \mathbf{K}_{fp}\mathbf{U}_p\right) \quad \text{step A} \tag{401a}$$
$$\mathbf{F}_p = \mathbf{K}_{pf}\mathbf{U}_f + \mathbf{K}_{pp}\mathbf{U}_p \quad \text{step B} \tag{401b}$$

Truss example: Solution of global system $(f + p)$

- From our global system we observe,

$$\mathbf{K}_{ff} = \begin{bmatrix} 0.5410 & 0.019 & -0.24 \\ 0.019 & 0.401 & -0.18 \\ -0.24 & -0.18 & 0.5371 \end{bmatrix} \quad \mathbf{K}_{fp} = \mathbf{K}_{pf}^{\mathrm{T}} = \begin{bmatrix} -0.32 & -0.221 & 0.221 \\ -0.24 & 0.221 & -0.221 \\ 0.24 & 0 & -0.3571 \end{bmatrix}$$

$$\mathbf{K}_{pp} = \begin{bmatrix} 0.32 & 0 & 0 \\ 0 & 0.221 & -0.221 \\ 0 & -0.221 & 0.5781 \end{bmatrix} \quad \mathbf{F}_f = \begin{bmatrix} 0 \\ -1 \\ 0 \end{bmatrix}, \quad \mathbf{U}_p = \begin{bmatrix} 0 \\ 0.5 \\ 0 \end{bmatrix} \quad \Rightarrow$$

- First, we notice,

$$-\mathbf{K}_{fp}\mathbf{U}_p = \begin{bmatrix} 0.1105 \\ -0.1105 \\ 0 \end{bmatrix} \quad \Rightarrow \quad \mathbf{F}_f - \mathbf{K}_{fp}\mathbf{U}_p = \begin{bmatrix} 0.1105 \\ -1.1105 \\ 0 \end{bmatrix} \tag{402}$$

- Compare:
  1. $-\mathbf{K}_{fp}\mathbf{U}_p$ with $\mathbf{F}_e$ if free dof only approach (in this problem $\mathbf{F}_e$ is entirely obtained from $\mathbf{F}_D$ as $\mathbf{F}_r = 0$ and $\mathbf{F}_N = 0$).
  2. $\mathbf{F}_f - \mathbf{K}_{fp}\mathbf{U}_p$ with $\mathbf{F} = \mathbf{F}_N + \mathbf{F}_e$ in free dof only approach.

Truss example: Solution of global system $(f + p)$

- From (401a) we have,

$$\mathbf{U}_f = {\mathbf{K}_{ff}}^{-1}\left(\mathbf{F}_f - \mathbf{K}_{fp}\mathbf{U}_p\right) = \begin{bmatrix} 0.5410 & 0.019 & -0.24 \\ 0.019 & 0.401 & -0.18 \\ -0.24 & -0.18 & 0.5371 \end{bmatrix}^{-1} \begin{bmatrix} 0.1105 \\ -1.1105 \\ 0 \end{bmatrix} = \begin{bmatrix} -0.2123 \\ -3.2980 \\ 1.20 \end{bmatrix}$$

- As expected, this matches our solution from free dof assembled system.
- Next, we obtain prescribed dof forces (reaction forces) from second set of equations (401b):

$$\mathbf{F}_p = \mathbf{K}_{pf}\mathbf{U}_f + \mathbf{K}_{pp}\mathbf{U}_p = \begin{bmatrix} -0.32 & -0.24 & 24 \\ -0.221 & 0.221 & 0 \\ 0.221 & -0.221 & -0.3571 \end{bmatrix} \begin{bmatrix} -0.2123 \\ -3.2980 \\ 1.20 \end{bmatrix}$$

$$+ \begin{bmatrix} 0.32 & 0 & 0 \\ 0 & 0.221 & -0.221 \\ 0 & -0.221 & 0.5781 \end{bmatrix} \begin{bmatrix} 0 \\ 0.5 \\ 0 \end{bmatrix} \quad \Rightarrow \quad \mathbf{F}_p = \begin{bmatrix} R_1^2 \\ R_1^3 \\ R_2^3 \end{bmatrix} = \begin{bmatrix} 0.5714 \\ -0.5714 \\ 1.0 \end{bmatrix}$$

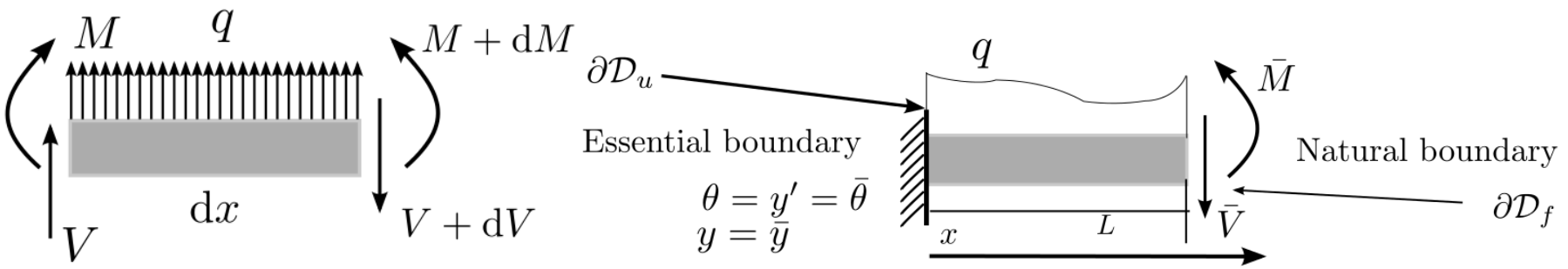
- This solution matches with free dof only approach in (398).

Comparison of f vs $f+p$ assembly approaches

	f	$f+p$	comment
$\mathbf{K}$	$n_{\mathrm{f}} \times n_{\mathrm{f}}$	$(n_{\mathrm{f}}+n_{\mathrm{p}}) \times (n_{\mathrm{f}}+n_{\mathrm{p}})$	In (f+p) approach we assemble a larger matrix equation
$-\mathbf{F}_D$	$-\mathbf{K}_{fp}\bar{\mathbf{a}}$ (342c)	$-\mathbf{K}_{fp}\mathbf{U}_p$ (401a)	while in both approached we compute $-\mathbf{K}_{fp}\mathbf{U}_p$ and add it to load vector, in f dof assembly, we only compute element level $\mathbf{f}_D^e = \mathbf{k}^e\mathbf{a}^e$ (*cf.* (360b)) and global $\mathbf{K}_{fp}$ is never explicitly assembled which is a great advantage.
$\mathbf{F}_p$	Added from individual elements prescribed forces	$\mathbf{K}_{pf}\mathbf{U}_f + \mathbf{K}_{pp}\mathbf{U}_p$	Again, $p+f$ is more straight forward but computationally more demanding

- In general, in computer codes we only assemble $\mathbf{K}_{ff}$.
- Assembly and solution of $\mathbf{K}$ in $(f+p)$ system, although may be simpler, it comes with additional computational costs.

Review: Euler Bernoulli Beam, Strong and WR statements



1. BVP pages 32 to 34: Strong form + boundary conditions (BCs shown on page 34).

$$\frac{\mathrm{d}^2 EI}{\mathrm{d}x^2}\left(\frac{\mathrm{d}^2 y}{\mathrm{d}x^2}\right) - q = 0 \quad \textit{cf.}\ (28)$$

2. Weighted Residual Statement, pages 44 to 45. WR for the figure is given by (61):

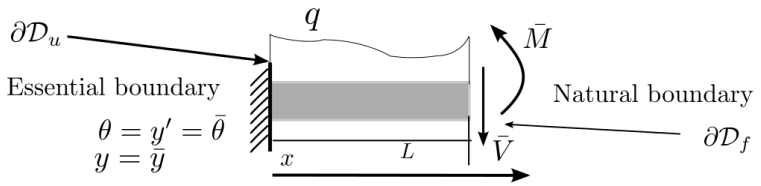
$$\text{Find } y \in \mathcal{V}^{\mathrm{WRS}} = \{u \in C^4(\mathcal{D}) \mid u(0) = \bar{y},\ \frac{\mathrm{d}u}{\mathrm{d}x}(0) = \bar{\theta}\},\ \text{such that,} \tag{403a}$$

$$\forall w \in \mathcal{W}^{\mathrm{WRS}} = C^1(\mathcal{D}) \text{ no need to enforce the homogeneous essential BCs for WRS} \tag{403b}$$

$$0 = \int_{\mathcal{D}} w\mathcal{R}_i(y)\,\mathrm{d}v + \int_{\partial\mathcal{D}_f} \mathbf{w}_f \mathcal{R}_f(y)\,\mathbf{ds}$$

$$= \int_0^L w\left(\frac{\mathrm{d}^2}{\mathrm{d}x^2}\left(EI\frac{\mathrm{d}^2 y}{\mathrm{d}x^2}\right) - q\right)\mathrm{d}x - \frac{\mathrm{d}w}{\mathrm{d}x}(\bar{M} - M(y))|_{x=L} + w(\bar{V} - V(y))|_{x=L} \tag{403c}$$

Review: Euler Bernoulli Beam, Weak statement



3. Weak statement , pages 51 to 57. For the figure (*cf.* (62)):

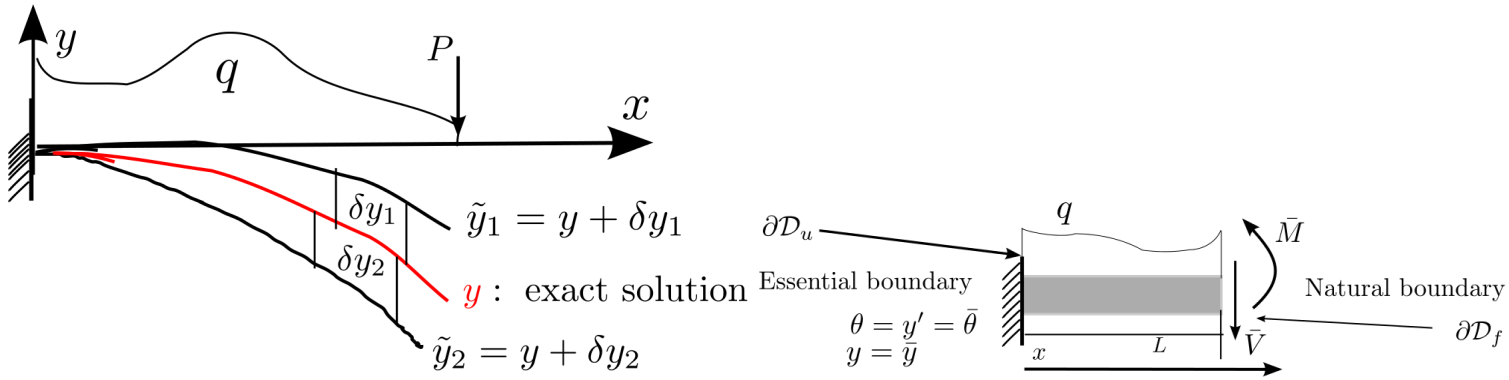
$$\text{Find } y \in \mathcal{V} = \{u \in C^2(\mathcal{D}) \mid u(0) = \bar{y},\ \frac{\mathrm{d}u}{\mathrm{d}x}(0) = \bar{\theta}\},\ \text{such that,} \tag{404a}$$

$$\forall w \in \mathcal{W} = \{u \in C^2(\mathcal{D}) \mid u(0) = 0,\ \frac{\mathrm{d}u}{\mathrm{d}x}(0) = 0\} \tag{404b}$$

$$0 = \int_0^L \left[\frac{\mathrm{d}^2 w}{\mathrm{d}x^2} EI \frac{\mathrm{d}^2 y}{\mathrm{d}x^2} - wq\right] \mathrm{d}x + \left\{-\frac{\mathrm{d}w}{\mathrm{d}x}\bar{M} + w\bar{V}\right\}_{x=L} \tag{404c}$$

- Weak Statement is well balanced; both y and w have a highest derivative of 2. In WRS highest derivatives are 4 and 1 for y and w, respectively.
- In Weak Statement, the integration by part cancels $V(y)$ and $M(y)$ at natural boundary.
- In Weak Statement, to get rid of terms introduced by integration by part at essential boundary **we require the weight function to satisfy homogeneous essential BCs**.
- The space of weight functions is often called $\mathcal{V}_0$ in weak statement as it is the same as solution space $\mathcal{V}$ with the exception of satisfying homogeneous (zero) version of essential BCs.
- Instead of C^2 conditions for y and w we require the more relaxed condition of H^2 (*i.e.*, 0th, 1st, 2nd derivatives are square integrable). This in turn means that C^1 continuity is sufficient for FEM trial functions.

Review: Euler Bernoulli Beam, Energy Method



- Energy method bypasses the strong form and weighted residual statements and directly yields the weak statement by minimizing the systems energy.
- Satisfaction of homogeneous essential BCs for weight functions in energy method (pages 71 to 73) has a very physical interpretation: Weight functions correspond to solution increment which should be zero at essential BC because all trial solutions satisfy essential BC (*cf.* figure above).
- Derivation of weak statement from energy method and comparison with balance law approach is given on pages 82 to 86.
- Virtual work statement for solid beam is given on page 88. As discussed, this is another form to express energy minimization statement.

$$\text{Find } y \in \mathcal{V} = \{v \in C^2([0,L]) \mid v(0) = \bar{y},\ \frac{\mathrm{d}v}{\mathrm{d}x}(0) = \bar{\theta}\},\ \text{s.t.},\ \forall \delta y \in \mathcal{W} = \{v \in C^2([0,L]) \mid v(0) = 0,\ \frac{\mathrm{d}v}{\mathrm{d}x}(0) = 0\}$$

$$\int_0^L \delta y''(x) E I y''(x)\,\mathrm{d}x = \int_0^L \delta y(x) q(x)\,\mathrm{d}x - \delta y(L)\bar{V} + \delta y'(L)\bar{M} \tag{405}$$

- We also discussed how we can go the opposite direction and obtain BC and strong from from the energy method; pages 90 to 92.

Review: Euler Bernoulli Beam, Function space

- From page 220 we observed that:

  Strong Form order $M = 2m$ $\Rightarrow$ Trial functions need to be H^m $\Rightarrow$

  FEM Trial functions of C^{m-1} form are sufficient

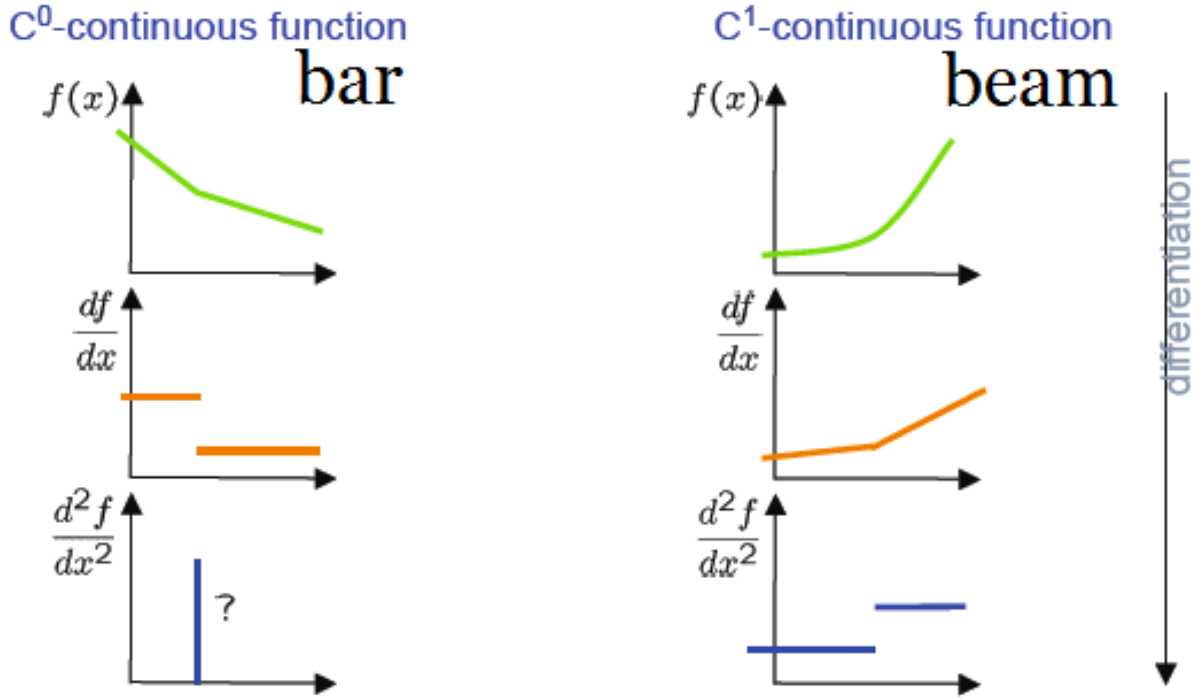
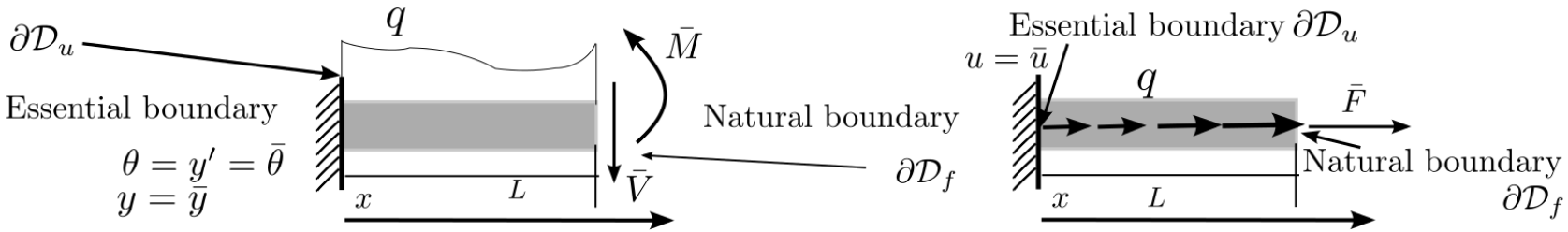


figure from F. Cirak

- C^1 for beam compared to C^0 for bar element is a major conceptual difference between the two physics.

FEM formulation of beam elements



- The weak statement for the problem in the figure can be written as (*cf.* (62)):

$$\text{Find } y \in \mathcal{V} = \{u \in C^2(\mathcal{D}) \mid u(0) = \bar{y},\ \frac{\mathrm{d}u}{\mathrm{d}x}(0) = \bar{\theta}\},\ \text{such that,} \tag{406a}$$

$$\forall w \in \mathcal{W} = \{u \in C^2(\mathcal{D}) \mid u(0) = 0,\ \frac{\mathrm{d}u}{\mathrm{d}x}(0) = 0\} \tag{406b}$$

$$\overbrace{\int_0^L \frac{\mathrm{d}^2 w}{\mathrm{d}x^2} EI \frac{\mathrm{d}^2 y}{\mathrm{d}x^2}\,\mathrm{d}x}^{\text{biliner form}} = \overbrace{\int_0^L wq\,\mathrm{d}x}^{\text{linear source term force}} + \overbrace{\left\{\frac{\mathrm{d}w}{\mathrm{d}x}\bar{M} - w\bar{V}\right\}_{x=L}}^{\text{natural BC force}} \tag{406c}$$

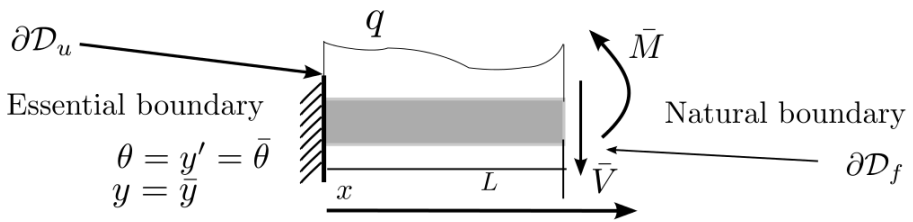
- This can be compared to weak statement for the bar element (right figure):

$$\text{Find } u \in \mathcal{V} = \{u \in C^1(\mathcal{D}) \mid u(0) = \bar{u}\},\ \text{such that,} \tag{407a}$$

$$\forall w \in \mathcal{W} = \{u \in C^1(\mathcal{D}) \mid u(0) = 0\} \tag{407b}$$

$$\overbrace{\int_0^L \frac{\mathrm{d}^1 w}{\mathrm{d}x^1} EA \frac{\mathrm{d}^1 y}{\mathrm{d}x^1}\,\mathrm{d}x}^{\text{biliner form}} = \overbrace{\int_0^L wq\,\mathrm{d}x}^{\text{linear source term force}} + \overbrace{\{w.\bar{F}\}_{x=L}}^{\text{natural BC force}} \tag{407c}$$

FEM formulation of beam elements



To cast beam problem into a general FEM framework and obtain force vectors we refer to (338):

$$\mathcal{A}(\mathbf{w},\mathbf{y}) = (\mathbf{w},\mathbf{r})_r + \left(\mathbf{w},\bar{\mathbf{F}}\right)_N$$

where for the beam problem:

1. Bilinear form:

$$\mathcal{A}(\mathbf{w},\mathbf{y}) = \int_{\mathcal{D}} L_m(\mathbf{w})\mathbf{D}L_m(\mathbf{y})\,\mathrm{dv} = \int_0^L \frac{\mathrm{d}^2 w}{\mathrm{d}x^2} EI \frac{\mathrm{d}^2 y}{\mathrm{d}x^2}\,\mathrm{d}x \;\Rightarrow\; \boxed{L_m(.) = \frac{\mathrm{d}^2(.)}{\mathrm{d}x^2}, \mathbf{D} = EI} \tag{408}$$

2. Source term force (similar to bar with q being vertical instead of horizontal):

$$(\mathbf{w},\mathbf{r})_r = \int_{\mathcal{D}} \mathbf{w}.\mathbf{r}\,\mathrm{dv} = \int_0^L w.q\,\mathrm{d}x \quad \boxed{\mathbf{r} = q} \tag{409}$$

3. Natural BC force. For this particular problem (in the figure):

$$\left(\mathbf{w},\bar{\mathbf{F}}\right)_N = \int_{\partial\mathcal{D}_f} \mathbf{w}\bar{\mathbf{F}} = \left\{\frac{\mathrm{d}w}{\mathrm{d}x}\bar{M} - w\bar{V}\right\}_{x=L} \tag{410}$$

As for the bar element, it is much more convenient (and general) to lump natural BCs into concentrated forces for beam element since this a 1D domain.

FEM formulation of beam elements

- According to our findings on previous slide and our general derivation of local stiffness matrix and force vectors we have (*cf.* page 295):
- Displacement-to-strain matrix $\mathbf{B}$:

$$\mathbf{B}^e = L_m(\mathbf{N}^e) = \frac{\mathrm{d}^2\mathbf{N}^e}{\mathrm{d}x^2} \tag{411}$$

transfer vertical displacement to curvature $\kappa = \frac{\mathrm{d}^2 y}{\mathrm{d}x^2}$(strain for beam problem):

- Material parameter:

$$\mathbf{D} = EI \tag{412}$$

- Local stiffness matrix, $\mathbf{k}^e$ (*cf.* (362)):

$$\mathbf{k}^e = \int_e \mathbf{B}^{e\,\mathrm{T}}\mathbf{D}\mathbf{B}^e \,\mathrm{dv} = \int_0^{L^e} \mathbf{B}^{e\,\mathrm{T}}\mathbf{D}\mathbf{B}^e \,\mathrm{d}x = \int_0^{L^e} \frac{\mathrm{d}^2\mathbf{N}^e}{\mathrm{d}x^2}^{\mathrm{T}} EI \frac{\mathrm{d}^2\mathbf{N}^e}{\mathrm{d}x^2} \,\mathrm{d}x \tag{413}$$

where L^e is the length of the element.

- Local force vector, $\mathbf{f}_e^e$, is the assembly of all element level forces (*cf.* (363)):

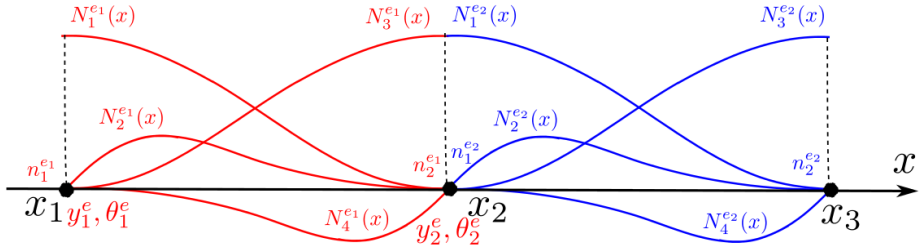
$$\mathbf{f}_e^e := \mathbf{f}_r^e - \mathbf{f}_D^e \tag{414a}$$

$$\mathbf{f}_r^e = \int_e \mathbf{N}^{e\,\mathrm{T}}.\mathbf{r} \,\mathrm{dv} = \int_0^{L^e} \mathbf{N}^{e\,\mathrm{T}}.q \,\mathrm{d}x \tag{414b}$$

$$\mathbf{f}_D^e = \mathbf{k}^e\mathbf{a}^e \tag{414c}$$

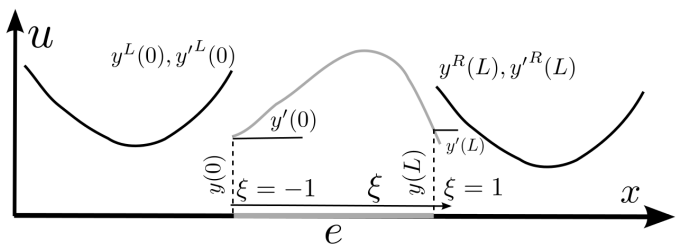
where as before $\mathbf{a}^e$ are element dofs with prescribed dofs set. Note that we have not added the natural BC force $\mathbf{f}_N^e$ as for 1D problems we can more conveniently incorporate that into concentrated forces.

FEM formulation of beam elements: Shape functions



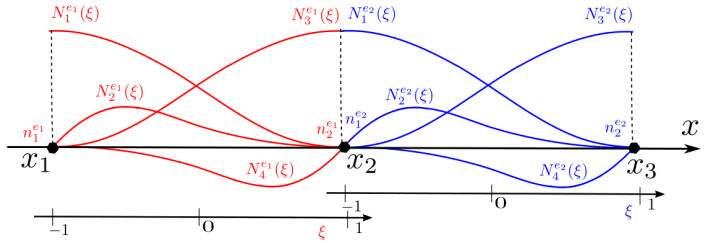
- Since beam displacement y^h should be C^1 continuous, dof at nodes are vertical displacement and rotation at each node.
- That is, there are two dofs per node as opposed to one (horizontal) displacement for the bar element.
- In the figure global versus local shape functions and dofs are shown.
- Numbering of dofs again is incremented first at node dofs then across nodes.
- As discussed before (*cf.* page 224) on each end we need two continuity condition (global C^1):

$$y(0) = y^L(0) \qquad y'(0) = y'^L(0)$$
$$y(L) = y^R(L) \qquad y'(L) = y'^R(L)$$



which implies that if we use polynomials to interpolate N_i they should be cubic (four coefficients).

FEM formulation of beam elements: Shape functions



- While there are many options for the local coordinate system for the beam (*e.g.*, $\xi = x - x_i$) we adopt the same coordinate system that we used for the bar element:

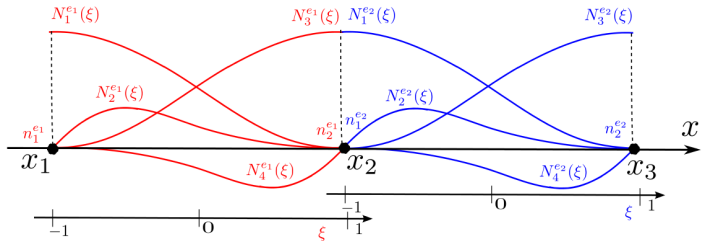
$$\left.\begin{aligned} x = x_i \quad &\rightarrow \xi = -1 \\ x = x_{i+1} \quad &\rightarrow \xi = 1 \end{aligned}\right\} \Rightarrow x = N_1^L x_i + N_2^L x_{i+1} = x_m^e + \frac{L^e}{2}\xi \Rightarrow \boxed{\mathrm{d}x = \frac{L^e}{2}\mathrm{d}\xi} \tag{415}$$

  where as in (369), $x_m^e = \frac{x_{i+1}+x_i}{2}$ and $L^e = x_{i+1} - x_i$ are the midpoint and length of element.
  Not that N_1^L and N_2^L used above for the interpolation between x and ξ are linear shape functions used for the bar element, but the cubic shape functions we will derive for beam element. They are only used to map local to global geometry (*cf.* (370) and (372)).
- Local dofs and shape functions of element are,

$$\mathbf{a}^e = \begin{bmatrix} y_1^e \\ \theta_1^e \\ y_2^e \\ \theta_2^e \end{bmatrix}, \quad \mathbf{N} = [N_1 \ N_2 \ N_3 \ N_4] \tag{416}$$

  where y_1^e and y_2^e are the vertical displacements and θ_1^e and θ_2^e are the slopes $\frac{\mathrm{d}y}{\mathrm{d}x}$ at the two end points.

FEM formulation of beam elements: Shape functions



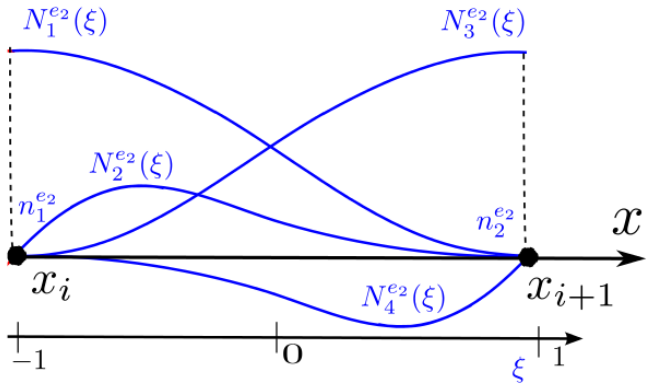
- Shape function N_i take the value of 1 at dof i and zero elsewhere:

$$\begin{array}{llllllll} N_1(\xi=-1) & =1 & \frac{\mathrm{d}N_1}{\mathrm{d}x}(\xi=-1) & =0 & N_1(\xi=1) & =0 & \frac{\mathrm{d}N_1}{\mathrm{d}x}(\xi=1) & =0 \\ N_2(\xi=-1) & =0 & \frac{\mathrm{d}N_2}{\mathrm{d}x}(\xi=-1) & =1 & N_2(\xi=1) & =0 & \frac{\mathrm{d}N_2}{\mathrm{d}x}(\xi=1) & =0 \\ N_3(\xi=-1) & =0 & \frac{\mathrm{d}N_3}{\mathrm{d}x}(\xi=-1) & =0 & N_3(\xi=1) & =1 & \frac{\mathrm{d}N_3}{\mathrm{d}x}(\xi=1) & =0 \\ N_4(\xi=-1) & =0 & \frac{\mathrm{d}N_4}{\mathrm{d}x}(\xi=-1) & =0 & N_4(\xi=1) & =0 & \frac{\mathrm{d}N_4}{\mathrm{d}x}(\xi=1) & =1 \end{array} \tag{417}$$

- Similar to bar element, since N_i are expressed in terms of ξ we need to transfer $\frac{\mathrm{d}N_i}{\mathrm{d}\xi}$ to $\frac{\mathrm{d}N_4}{\mathrm{d}x}$ (*cf.* (415)):

$$\mathrm{d}x = \frac{L^e}{2}\mathrm{d}\xi \quad \Rightarrow \quad \frac{\mathrm{d}N_i}{\mathrm{d}x} = \frac{\frac{\mathrm{d}N_i}{\mathrm{d}\xi}}{\frac{\mathrm{d}x}{\mathrm{d}\xi}} \quad \Rightarrow \quad \boxed{\frac{\mathrm{d}N_i}{\mathrm{d}x} = \frac{2}{L^e}\frac{\mathrm{d}N_i}{\mathrm{d}\xi}, \quad \frac{\mathrm{d}N_i}{\mathrm{d}\xi} = \frac{L^e}{2}\frac{\mathrm{d}N_i}{\mathrm{d}x}} \tag{418}$$

FEM formulation of beam elements: Shape functions



- From (417) and (418) we get:

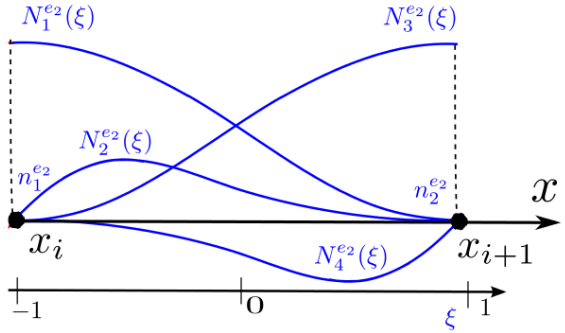
$$
\begin{array}{llll}
N_1(\xi=-1) = 1 & \frac{\mathrm{d}N_1}{\mathrm{d}\xi}(\xi=-1) = 0 & N_1(\xi=1) = 0 & \frac{\mathrm{d}N_1}{\mathrm{d}\xi}(\xi=1) = 0 \\
N_2(\xi=-1) = 0 & \frac{\mathrm{d}N_2}{\mathrm{d}\xi}(\xi=-1) = \frac{L^e}{2} & N_2(\xi=1) = 0 & \frac{\mathrm{d}N_2}{\mathrm{d}\xi}(\xi=1) = 0 \\
N_3(\xi=-1) = 0 & \frac{\mathrm{d}N_3}{\mathrm{d}\xi}(\xi=-1) = 0 & N_3(\xi=1) = 1 & \frac{\mathrm{d}N_3}{\mathrm{d}\xi}(\xi=1) = 0 \\
N_4(\xi=-1) = 0 & \frac{\mathrm{d}N_4}{\mathrm{d}\xi}(\xi=-1) = 0 & N_4(\xi=1) = 0 & \frac{\mathrm{d}N_4}{\mathrm{d}\xi}(\xi=1) = \frac{L^e}{2}
\end{array}
\tag{419}
$$

- Since each N_i has four conditions, we interpolate them with cubic polynomials:

$$
N_i = \alpha_0 + \alpha_1\xi + \alpha_2\xi^2 + \alpha_3\xi^3 \quad \Rightarrow \quad \frac{\mathrm{d}N_i}{\mathrm{d}\xi} = \alpha_1 + 2\alpha_2\xi + 3\alpha_3\xi^2 \tag{420}
$$

α_j are determined from the conditions in (419).

FEM formulation of beam elements: Shape functions



- For example, to determine N_1 from (419) and (420) we observe:

$$\left.\begin{array}{rl} N_1(\xi=-1) & =1 \\ \frac{\mathrm{d}N_1}{\mathrm{d}\xi}(\xi=-1) & =0 \\ N_1(\xi=1) & =0 \\ \frac{\mathrm{d}N_1}{\mathrm{d}\xi}(\xi=1) & =0 \end{array}\right\} \Rightarrow \left.\begin{array}{rl} \alpha_0-\alpha_1+\alpha_2-\alpha_3 & =1 \\ \alpha_1-2\alpha_2+3\alpha_3 & =0 \\ \alpha_0+\alpha_1+\alpha_2+\alpha_3 & =0 \\ \alpha_1+2\alpha_2+3\alpha_3 & =0 \end{array}\right\} \Rightarrow \begin{bmatrix} \alpha_0 \\ \alpha_1 \\ \alpha_2 \\ \alpha_3 \end{bmatrix} = \begin{bmatrix} \frac{1}{2} \\ -\frac{3}{4} \\ 0 \\ \frac{1}{4} \end{bmatrix}$$

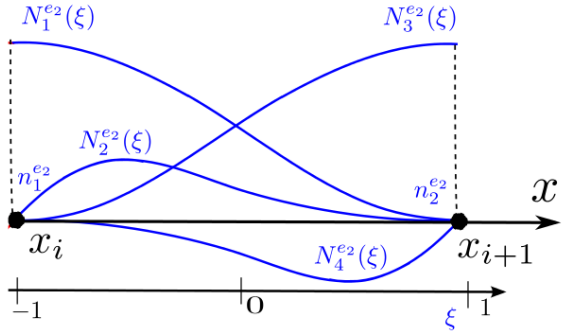
$$\boxed{N_1(\xi)=\frac{1}{4}(2-3\xi+\xi^3)}$$

- Similarly for N_2:

$$\left.\begin{array}{rl} N_2(\xi=-1) & =0 \\ \frac{\mathrm{d}N_2}{\mathrm{d}\xi}(\xi=-1) & =\frac{L^e}{2} \\ N_2(\xi=1) & =0 \\ \frac{\mathrm{d}N_2}{\mathrm{d}\xi}(\xi=1) & =0 \end{array}\right\} \Rightarrow \left.\begin{array}{rl} \alpha_0-\alpha_1+\alpha_2-\alpha_3 & =0 \\ \alpha_1-2\alpha_2+3\alpha_3 & =\frac{L^e}{2} \\ \alpha_0+\alpha_1+\alpha_2+\alpha_3 & =0 \\ \alpha_1+2\alpha_2+3\alpha_3 & =0 \end{array}\right\} \Rightarrow \begin{bmatrix} \alpha_0 \\ \alpha_1 \\ \alpha_2 \\ \alpha_3 \end{bmatrix} = \frac{L^e}{8}\begin{bmatrix} 1 \\ -1 \\ -1 \\ 1 \end{bmatrix}$$

$$\boxed{N_2(\xi)=\frac{L^e}{8}(1-\xi-\xi^2+\xi^3)}$$

FEM formulation of beam elements: Shape functions



- For N_3:

$$\left.\begin{array}{rl} N_3(\xi=-1) & =0 \\ \frac{\mathrm{d}N_3}{\mathrm{d}\xi}(\xi=-1) & =0 \\ N_3(\xi=1) & =1 \\ \frac{\mathrm{d}N_3}{\mathrm{d}\xi}(\xi=1) & =0 \end{array}\right\} \Rightarrow \left.\begin{array}{rl} \alpha_0-\alpha_1+\alpha_2-\alpha_3 & =0 \\ \alpha_1-2\alpha_2+3\alpha_3 & =0 \\ \alpha_0+\alpha_1+\alpha_2+\alpha_3 & =1 \\ \alpha_1+2\alpha_2+3\alpha_3 & =0 \end{array}\right\} \Rightarrow \begin{bmatrix} \alpha_0 \\ \alpha_1 \\ \alpha_2 \\ \alpha_3 \end{bmatrix} = \begin{bmatrix} \frac{1}{2} \\ \frac{3}{4} \\ 0 \\ -\frac{1}{4} \end{bmatrix}$$

$$\boxed{N_3(\xi)=\frac{1}{4}(2+3\xi-\xi^3)}$$

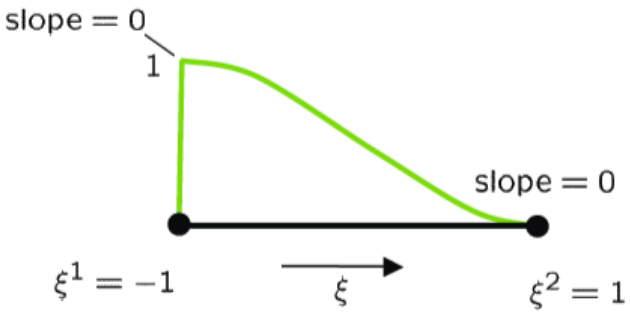
- and for N_4:

$$\left.\begin{array}{rl} N_4(\xi=-1) & =0 \\ \frac{\mathrm{d}N_4}{\mathrm{d}\xi}(\xi=-1) & =0 \\ N_4(\xi=1) & =0 \\ \frac{\mathrm{d}N_4}{\mathrm{d}\xi}(\xi=1) & =\frac{L^e}{2} \end{array}\right\} \Rightarrow \left.\begin{array}{rl} \alpha_0-\alpha_1+\alpha_2-\alpha_3 & =0 \\ \alpha_1-2\alpha_2+3\alpha_3 & = \\ \alpha_0+\alpha_1+\alpha_2+\alpha_3 & =0 \\ \alpha_1+2\alpha_2+3\alpha_3 & =\frac{L^e}{2} \end{array}\right\} \Rightarrow \begin{bmatrix} \alpha_0 \\ \alpha_1 \\ \alpha_2 \\ \alpha_3 \end{bmatrix} = \frac{L^e}{8}\begin{bmatrix} -1 \\ -1 \\ 1 \\ 1 \end{bmatrix}$$

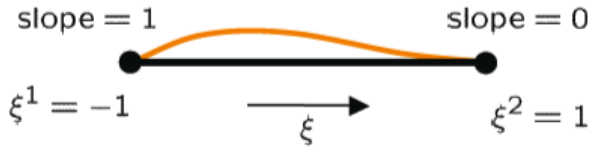
$$\boxed{N_4(\xi)=\frac{L^e}{8}(-1-\xi+\xi^2+\xi^3)}$$

FEM formulation of beam elements: Shape functions

- Shape functions of node 1

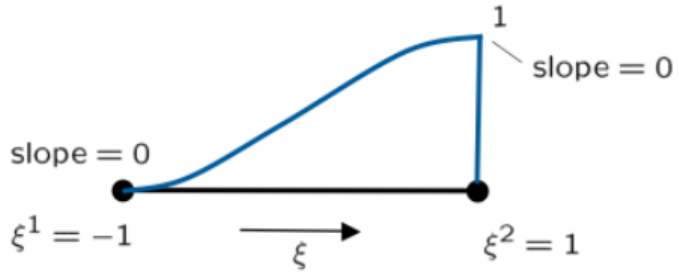


$$N_1^e(\xi) = \frac{1}{4}(1-\xi)^2(2+\xi)$$

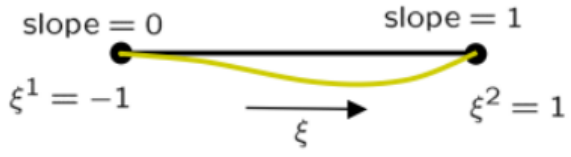


$$N_2^e(\xi) = \frac{l_e}{8}(1-\xi)^2(1+\xi)$$

- Shape functions of Node 2



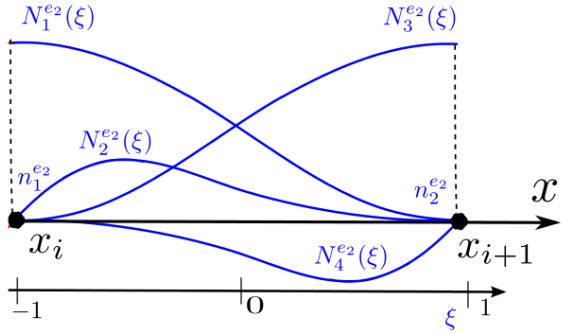
$$N_3^e(\xi) = \frac{1}{4}(1+\xi)^2(2-\xi)$$



$$N_4^e(\xi) = \frac{l_e}{8}(1+\xi)^2(\xi-1)$$

figure from F. Cirak

FEM formulation of beam elements: $\mathbf{B}^e$ matrix



- From (411) we have,

$$\mathbf{B}^e = L_m(\mathbf{N}^e) = \frac{\mathrm{d}^2\mathbf{N}^e}{\mathrm{d}x^2}$$

- The solution for shape functions can be summarized as,

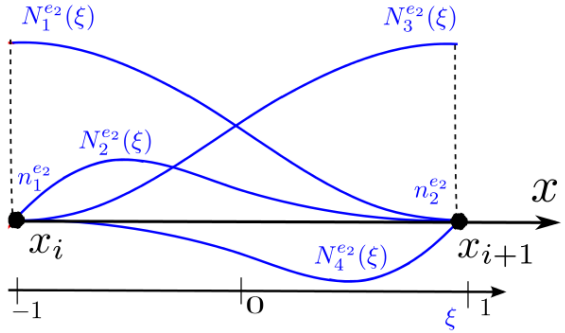
$$N_1^e(\xi) = \frac{1}{4}(2 - 3\xi + \xi^3) \qquad N_2^e(\xi) = \frac{L^e}{8}(1 - \xi - \xi^2 + \xi^3) \tag{421a}$$

$$N_3^e(\xi) = \frac{1}{4}(2 + 3\xi - \xi^3) \qquad N_4^e(\xi) = \frac{L^e}{8}(-1 - \xi + \xi^2 + \xi^3) \tag{421b}$$

- Since, $\mathbf{B}^e = \frac{\mathrm{d}^2\mathbf{N}^e}{\mathrm{d}x^2}$ and $\mathbf{N}^e$ is a function of ξ similar to bar element we need to do transform local derivatives to global derivatives (*cf.* (418)):

$$\mathbf{B}^e = \frac{\mathrm{d}^2\mathbf{N}^e}{\mathrm{d}x^2} = \frac{1}{\left(\frac{\mathrm{d}x}{\mathrm{d}\xi}\right)^2}\frac{\mathrm{d}^2\mathbf{N}^e}{\mathrm{d}\xi^2} = \frac{4}{L^{e\,2}}\frac{\mathrm{d}^2\mathbf{N}^e}{\mathrm{d}\xi^2} \tag{422}$$

FEM formulation of beam elements: Stiffness matrix



- According to (421) and (422) we have,

$$B_1^e(\xi) = \frac{6\xi}{L^{e\,2}} \qquad B_2^e(\xi) = \frac{-1+3\xi}{L^e} \tag{423a}$$

$$B_3^e(\xi) = -\frac{6\xi}{L^{e\,2}} \qquad B_4^e(\xi) = \frac{1+3\xi}{L^e} \tag{423b}$$

- That is,

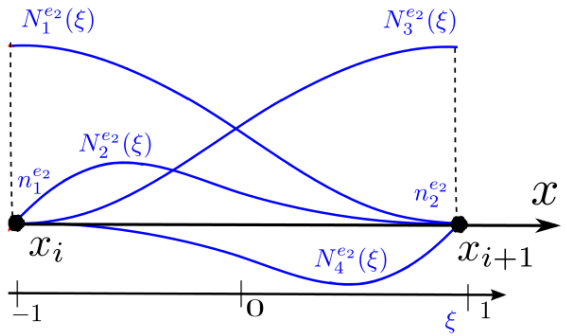
$$\boxed{\mathbf{B}^e = [B_1^e \;\; B_2^e \;\; B_3^e \;\; B_4^e] = [\frac{6\xi}{L^{e\,2}} \;\; \frac{-1+3\xi}{L^e} \;\; -\frac{6\xi}{L^{e\,2}} \;\; \frac{1+3\xi}{L^e}]} \tag{424}$$

- From (413) and $\mathrm{d}x = \frac{L^e}{2}\mathrm{d}\xi$ (*cf.* (415)) we have,

$$\mathbf{k}^e = \int_0^{L^e} \mathbf{B}^{e\,\mathrm{T}}\mathbf{D}\mathbf{B}^e \,\mathrm{d}x \quad \text{for } \mathbf{D} = EI \quad \Rightarrow$$

$$\mathbf{k}^e = \int_{-1}^{1} \mathbf{B}^e(\xi)^{\mathrm{T}} E(\xi) I(\xi) \mathbf{B}^e(\xi) \{\frac{L^e}{2}\mathrm{d}\xi\} \tag{425}$$

FEM formulation of beam elements: Stiffness matrix



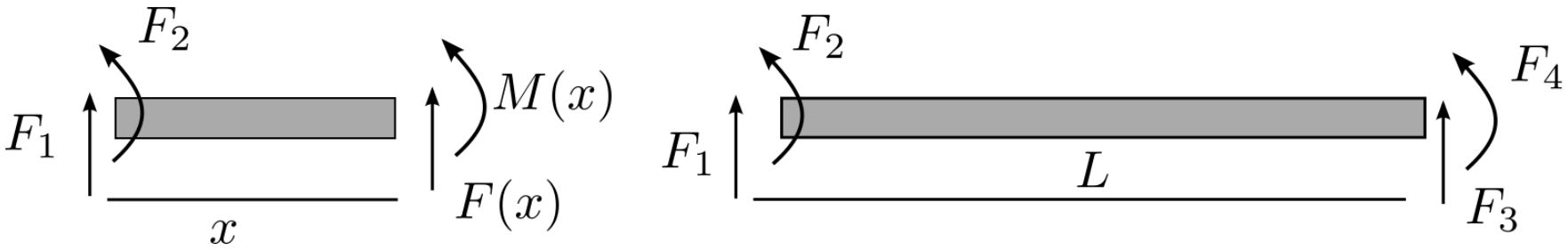
- From (424) and (425) we have,

$$\mathbf{k}^e = \frac{1}{2L^e}\int_{-1}^{1}\begin{bmatrix}\frac{6\xi}{L^e}\\ -1+3\xi\\ -\frac{6\xi}{L^e}\\ 1+3\xi\end{bmatrix} E(\xi)I(\xi)\begin{bmatrix}\frac{6\xi}{L^e} & -1+3\xi & -\frac{6\xi}{L^e} & 1+3\xi\end{bmatrix}\mathrm{d}\xi \tag{426}$$

- If E and I are constant, we can take those out of the equation and have:

$$\mathbf{k}^e = \frac{EI}{L^{e3}}\begin{bmatrix}12 & 6L^e & -12 & 6L^e\\ & 4L^{e2} & -6L^e & 2L^{e2}\\ & & 12 & -6L^e\\ \text{sym.} & & & 4L^{e2}\end{bmatrix} \quad \text{for constant } E \text{ and } I \tag{427}$$

beam element, Stiffness matrix: Direct method



- Similar to the development on pages 307 to 309 where we derived stiffness matrix from direct method and compared it to FEM method, we can derive the stiffness matrix directly for the beam element. Steps are as followed:
  1. Specify the number of independent forces needed. As shown in the right figure, we apply four forces F_1 to F_4 to the beam (F_1 and F_3 are vertical forces; F_2 and F_4 are moments). Two equations $\sum F_y = 0$ and $\sum M_z = 0$ imply that only $4 - 2 = 2$ of these quantities are independent. We choose F_1 and F_2 and observe $F_3 = -F_1$, $F_4 = F_1 L - F_2$.
  2. Obtain $M(x)$ from the left figure:

$$M(x) = F_1 x - F_2$$

  3. Compute curvature:

$$\kappa(x) = \frac{M(x)}{E(x)I(x)}$$

beam element, Stiffness matrix: Direct method

4. Compute rotation $\theta = \frac{\mathrm{d}y}{\mathrm{d}x}$

$$\theta(x) = \frac{\mathrm{d}y}{\mathrm{d}x} = \theta(0) + \int_0^x \kappa(\eta)\,\mathrm{d}\eta \quad \Rightarrow$$

$$\theta(x) = a_2^e + F_1\left\{\int_0^x \frac{\eta}{E(\eta)I(\eta)}\,\mathrm{d}\eta\right\} - F_2\left\{\int_0^x \frac{1}{E(\eta)I(\eta)}\,\mathrm{d}\eta\right\} \tag{428a}$$

$$\boxed{a_4^e = a_2^e + \int_0^{L^e} \frac{F_1\eta - F_2}{E(\eta)I(\eta)}\,\mathrm{d}\eta} \tag{428b}$$

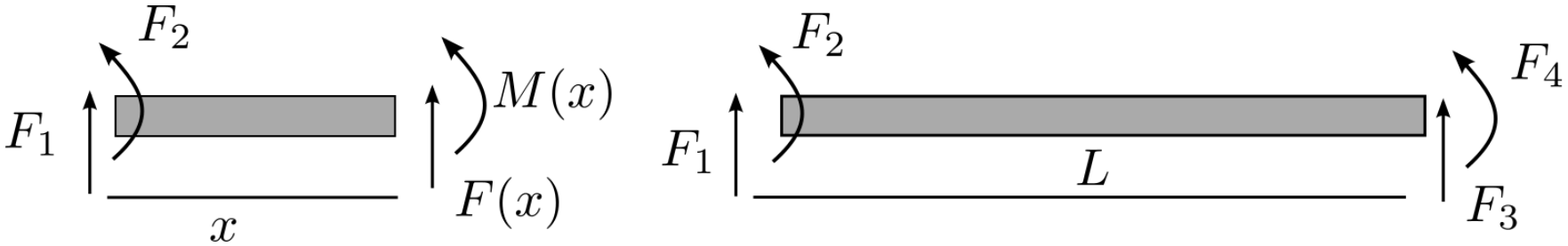
note that we have used $\theta(0) = a_2^e$ and $\theta(L^e) = a_4^e$.

5. Compute y using (428a):

$$\begin{aligned} y(x) &= y(0) + \int_0^x \frac{\mathrm{d}y}{\mathrm{d}x}(\xi)\,\mathrm{d}\xi \\ &= a_1^e + \int_0^x \left[a_2^e + F_1\left\{\int_0^\xi \frac{\eta}{E(\eta)I(\eta)}\,\mathrm{d}\eta\right\} - F_2\left\{\int_0^\xi \frac{1}{E(\eta)I(\eta)}\,\mathrm{d}\eta\right\}\right]\,\mathrm{d}\xi \\ &= a_1^e + a_2^e x + F_1\left\{\int_0^x\int_0^\xi \frac{\eta}{E(\eta)I(\eta)}\,\mathrm{d}\eta\,\mathrm{d}\xi\right\} - F_2\left\{\int_0^x\int_0^\xi \frac{1}{E(\eta)I(\eta)}\,\mathrm{d}\eta\,\mathrm{d}\xi\right\} \end{aligned} \tag{429a}$$

we have used $a_1^e = y(0)$.

beam element, Stiffness matrix: Direct method

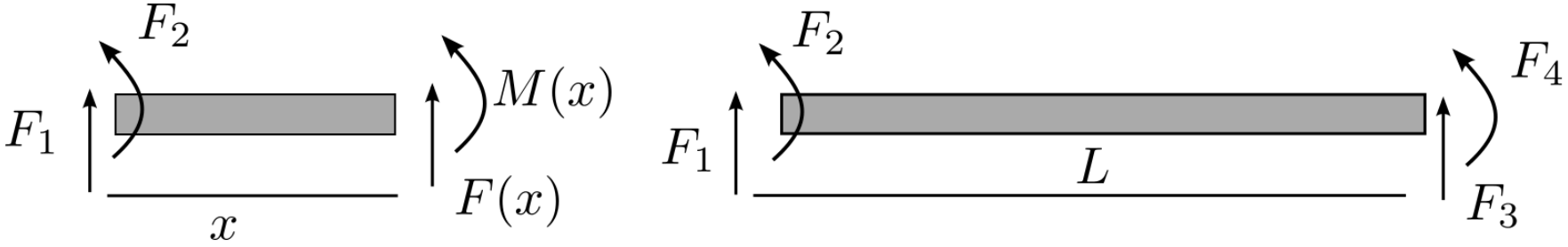


Plugging $x = L$ in (429a) to obtain $a_3^e = y(L^e)$ and rewriting (428b) we have:

$$a_3^e = a_1^e + a_2^e L^e + F_1 \left\{ \int_0^{L^e} \int_0^{\xi} \frac{\eta}{E(\eta)I(\eta)} \, \mathrm{d}\eta \, \mathrm{d}\xi \right\} - F_2 \left\{ \int_0^{L^e} \int_0^{\xi} \frac{1}{E(\eta)I(\eta)} \, \mathrm{d}\eta \, \mathrm{d}\xi \right\} \quad (430a)$$

$$a_4^e = a_2^e + \int_0^{L^e} \frac{F_1\eta - F_2}{E(\eta)I(\eta)} \, \mathrm{d}\eta \quad (430b)$$

beam element, Stiffness matrix: Direct method



Equation (430) can be written as:

$$\mathbf{C} = \mathbf{A}\mathbf{f}, \quad \text{where} \tag{431a}$$

$$\mathbf{C} = \begin{bmatrix} a_3^e - a_1^e - a_2^e L^e \\ a_4^e - a_2^e \end{bmatrix} \tag{431b}$$

$$\mathbf{A} = \begin{bmatrix} \int_0^{L^e} \int_0^{\xi} \frac{\eta}{E(\eta)I(\eta)} \, \mathrm{d}\eta \, \mathrm{d}\xi & -\int_0^{L^e} \int_0^{\xi} \frac{1}{E(\eta)I(\eta)} \, \mathrm{d}\eta \, \mathrm{d}\xi \\ \int_0^{L^e} \frac{\eta}{E(\eta)I(\eta)} \, \mathrm{d}\eta & -\int_0^{L^e} \frac{1}{E(\eta)I(\eta)} \, \mathrm{d}\eta \end{bmatrix} \tag{431c}$$

$$\mathbf{f} = \begin{bmatrix} F_1 \\ F_2 \end{bmatrix} \tag{431d}$$

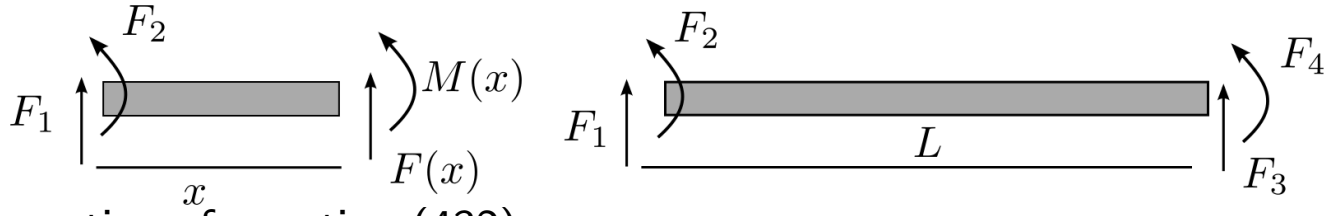
6. We can invert the relation (431a) to obtain forces F_1 to F_4 in terms of a_1^e to a_4^e:

$$\mathbf{f} = \mathbf{A}^{-1}\mathbf{C} \quad \Rightarrow \quad \begin{bmatrix} F_1 \\ F_2 \end{bmatrix} = \mathbf{A}^{-1} \begin{bmatrix} a_3^e - a_1^e - a_2^e L^e \\ a_4^e - a_2^e \end{bmatrix} \tag{432a}$$

$$F_3 = -F_1 \quad (\sum F_y = 0) \tag{432b}$$

$$F_4 = F_1 L^e - F_2 \quad (\sum M_z = 0) \tag{432c}$$

beam element, Stiffness matrix: Direct vs. FEM method



- Close inspection of equation (432):

$$\begin{bmatrix} F_1 \\ F_2 \end{bmatrix} = \mathbf{A}^{-1} \begin{bmatrix} a_3^e - a_1^e - a_2^e L^e \\ a_4^e - a_2^e \end{bmatrix}$$

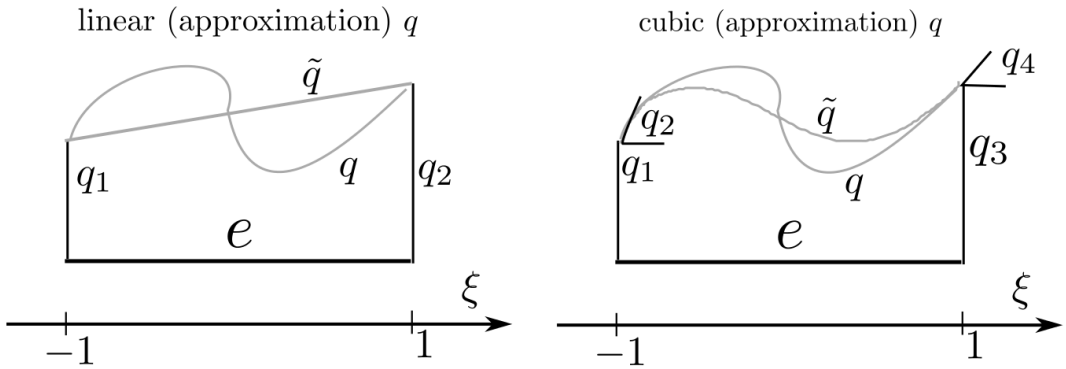
$$F_3 = -F_1$$

$$F_4 = F_1 L^e - F_2$$

reveals a a linear relation between four kinematic quantities a_1^e to a_4^e and F_1 to F_4 which is basically the 4×4 stiffness matrix obtained by direct method.

- We can easily show that for constant E and I direct and FEM methods yield the same stiffness matrix.
- As for the bar example, for nonuniform EI the cubic shape functions used for FEM displacement interpolation are no longer exact and FEM stiffness matrix is an approximation of the exact stiffness matrix obtained above.
- It is evident that the derivation of exact stiffness method is much more involved for the bar problem. In addition, as mentioned before the use of direct method (exact) stiffness method i) can lead to some inconsistencies if combined with some other terms based on FEM shape functions; ii) does not improve the solution convergence rate.
- For all three reasons above, the (possibly approximate) stiffness matrix obtained by FEM method is used in actual calculations.

Beam elements: Forces: A. Source term forces

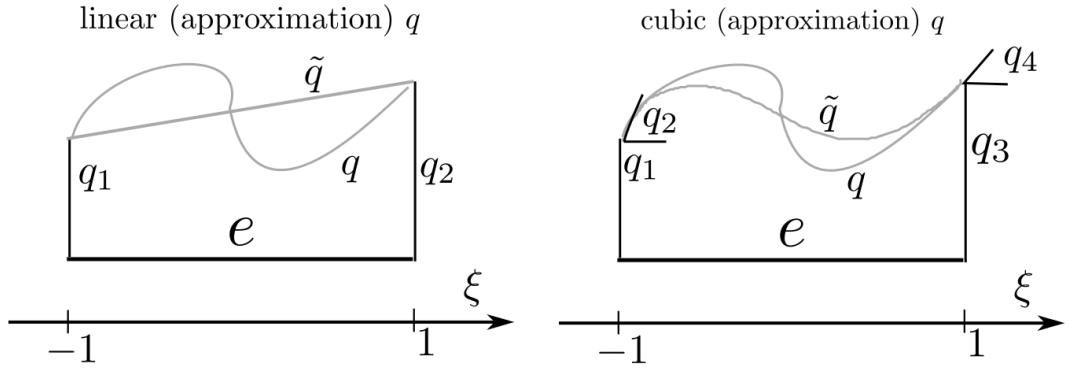


- From equation (414b)and (418) source term force is given by:

$$\mathbf{f}_r^e = \int_0^{L^e} \mathbf{N}^e(x)^{\mathrm{T}}.q(x)\,\mathrm{d}x = \int_{-1}^{1} \mathbf{N}^e(\xi)^{\mathrm{T}}.q(\xi)\left\{\frac{L^e}{2}\mathrm{d}\xi\right\}$$

- The are the following options to calculate $\mathbf{f}_r^e$:
  1. Direction integration: In this approach we directly integrate the equation above for a given $q(x)$. This integration can be done analytically or numerically. The latter is called quadrature which will be covered at subsequent sections. Either form of the integrals (in terms of x or ξ can be used)/
  2. Precomputed by assuming a specific form (approximation) for q called $\tilde{q}$: Similar to the bar example were we approximated q by a linear $\tilde{q}$ we can use finite element shape functions to approximate q for the beam element. Note that $\tilde{q}$ may be exact given the approximation space (linear, cubic, *etc.*) and the actual form of q.

Beam elements: Forces: A. Source term forces



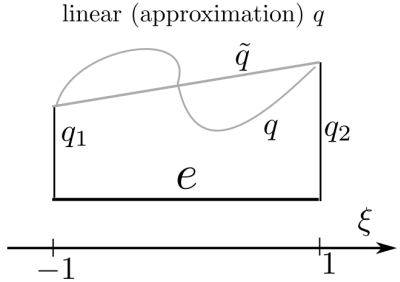
- Similar to the bar example, we discuss how we can compute the matrix $\mathbf{r}^e$ which transfers values of q at (some) nodal points to its corresponding force vector.
- We can use any approximation for q in $\mathbf{f}_r^e = \int_0^{L^e} \mathbf{N}^e(x)^{\mathrm{T}}.q(x)\,\mathrm{d}x$. The order of approximation does not need to be the same order as elements shape function (cubic for two node beam element). Two case are as follows:
  1. Linear: In this case, the function q is linearly interpolated by its end point values:

$$\tilde{q} = q_1 {N_L}_1^e + q_2 {N_L}_2^e = \mathbf{N}_L^e \begin{bmatrix} q_1 \\ q_2 \end{bmatrix}$$

To distinguish linear shape functions ${N_L}_i^e$ from the element solutions cubic shape functions N_1^e to N_4^e we decorate them with L:

$$\mathbf{N}_L^e = \begin{bmatrix} {N_L}_1^e & {N_L}_2^e \end{bmatrix} = \begin{bmatrix} \frac{1-\xi}{2} & \frac{1+\xi}{2} \end{bmatrix}$$

Beam elements: Forces: A. Source term forces



then the source term force is:

$$\mathbf{f}_r^e \approx \int_{-1}^{1} \mathbf{N}^e(\xi)^{\mathrm{T}}.\tilde{q}(\xi)\frac{L^e}{2}\mathrm{d}\xi = \mathbf{r}^e \begin{bmatrix} q_1 \\ q_2 \end{bmatrix} \quad \text{where}$$

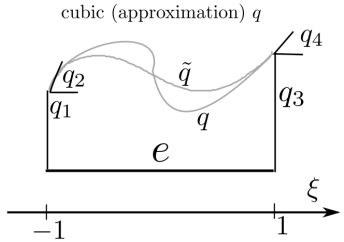
$$\mathbf{r}^e = \frac{L^e}{2}\int_{-1}^{1} \mathbf{N}^e(\xi)^{\mathrm{T}}.\mathbf{N}_L^e(\xi)\mathrm{d}\xi = \frac{L^e}{2}\int_{-1}^{1} \begin{bmatrix} \frac{1}{4}(2-3\xi+\xi^3) \\ \frac{L^e}{8}(1-\xi-\xi^2+\xi^3) \\ \frac{1}{4}(2+3\xi-\xi^3) \\ \frac{L^e}{8}(-1-\xi+\xi^2+\xi^3) \end{bmatrix} \cdot \begin{bmatrix} \frac{1-\xi}{2} & \frac{1+\xi}{2} \end{bmatrix} \mathrm{d}\xi \quad \Rightarrow$$

$$\boxed{\mathbf{f}_r^e \approx \mathbf{r}^e \begin{bmatrix} q_1 \\ q_2 \end{bmatrix} \quad \text{where } \mathbf{r}^e = L^e \begin{bmatrix} \frac{7}{20} & \frac{3}{20} \\ \frac{1}{20}L^e & \frac{1}{30}L^e \\ \frac{3}{20} & \frac{7}{20} \\ -\frac{1}{30}L^e & -\frac{1}{20}L^e \end{bmatrix} \quad \text{exact for linear } q} \qquad (433)$$

q_1 and q_2 are shown in the right figure. For constant $q(x) = \bar{q}$, $q_1 = q_2 = \bar{q}$, from (433) we get:

$$\mathbf{f}_r^e = \mathbf{r}^e \begin{bmatrix} \bar{q} \\ \bar{q} \end{bmatrix} = \begin{bmatrix} \frac{\bar{q}L^e}{2} \\ \frac{\bar{q}L^{e\,2}}{12} \\ \frac{\bar{q}L^e}{2} \\ -\frac{\bar{q}L^{e\,2}}{12} \end{bmatrix} \quad \text{constant } q(x) = \bar{q} \qquad (434)$$

Beam elements: Forces: A. Source term forces



2. Cubic: As shown in the figure q_1 and q_2 correspond to the value and slope of q at the n_1 and q_3 and q_4 to those at n_2, respectively. In this case, both solution and q are interpolated with cubic function and similar to linear approximation for q, $\mathbf{f}_r^e$ becomes:

$$\mathbf{f}_r^e \approx \int_{-1}^{1} \mathbf{N}^e(\xi)^{\mathrm{T}}.\tilde{q}(\xi)\frac{L^e}{2}\mathrm{d}\xi = \mathbf{r}^e \begin{bmatrix} q_1 \\ q_2 \\ q_3 \\ q_4 \end{bmatrix} \quad \text{where } \mathbf{r}^e = \frac{L^e}{2}\int_{-1}^{1} \mathbf{N}^e(\xi)^{\mathrm{T}}.\mathbf{N}^e(\xi)\mathrm{d}\xi$$

$$= \frac{L^e}{2}\int_{-1}^{1} \begin{bmatrix} \frac{1}{4}(2-3\xi+\xi^3) \\ \frac{L^e}{8}(1-\xi-\xi^2+\xi^3) \\ \frac{1}{4}(2+3\xi-\xi^3) \\ \frac{L^e}{8}(-1-\xi+\xi^2+\xi^3) \end{bmatrix} \cdot \begin{bmatrix} \frac{1}{4}(2-3\xi+\xi^3) \\ \frac{L^e}{8}(1-\xi-\xi^2+\xi^3) \\ \frac{1}{4}(2+3\xi-\xi^3) \\ \frac{L^e}{8}(-1-\xi+\xi^2+\xi^3) \end{bmatrix}^{\mathrm{T}} \mathrm{d}\xi \quad \Rightarrow$$

$$\mathbf{f}_r^e \approx \mathbf{r}^e \begin{bmatrix} q_1 \\ q_2 \\ q_3 \\ q_4 \end{bmatrix} \; : \; \mathbf{r}^e = L^e \begin{bmatrix} \frac{13}{35} & \frac{11}{210}L^e & \frac{9}{70} & -\frac{13}{420}L^e \\ \frac{11}{210}L^e & \frac{1}{105}L^{e2} & \frac{13}{420}L^e & -\frac{1}{140}L^{e2} \\ \frac{9}{70} & \frac{13}{420}L^e & \frac{13}{35} & -\frac{11}{210}L^e \\ -\frac{13}{420}L^e & -\frac{1}{140}L^{e2} & -\frac{11}{210}L^e & \frac{1}{105}L^{e2} \end{bmatrix} \quad \text{exact for cubic } q$$

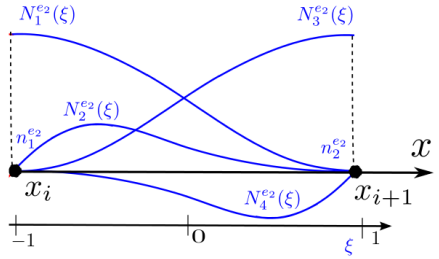
(435)

note that for constant $q(x) = \bar{q}$, $[q_1 \; q_2 \; q_3 \; q_4] = [1 \; 0 \; 1 \; 0]$ and we recover (434).

Beam elements: Forces: B. Natural BC force

- As discussed on page 364 and shown in (414a) for beam elements (and rotated beam elements in 2D and 3D frames) we do not compute natural BC force as such boundary is a set of points and corresponding forces can be treated as global nodal forces.

Beam elements: Forces: C. Essential BC force



- From (414c) on page 364, similar to all other element types, natural BC force is given by:

$$\mathbf{f}_D^e = \mathbf{k}^e \mathbf{a}^e$$

where $\mathbf{k}^e$ is given by (426):

$$\mathbf{k}^e = \frac{1}{2L^e} \int_{-1}^{1} \begin{bmatrix} \frac{6\xi}{L^e} \\ -1+3\xi \\ -\frac{6\xi}{L^e} \\ 1+3\xi \end{bmatrix} E(\xi) I(\xi) \begin{bmatrix} \frac{6\xi}{L^e} & -1+3\xi & -\frac{6\xi}{L^e} & 1+3\xi \end{bmatrix} \mathrm{d}\xi$$

and

$$\mathbf{a}^e = \begin{bmatrix} a_1^e \\ a_2^e \\ a_3^e \\ a_4^e \end{bmatrix}$$

with a_1^e, a_2^e being vertical displacement and rotation at n_1 and a_3^e, a_4^e those at n_2. As before, we set free dofs zero for this computation.

Beam elements: Element total Force

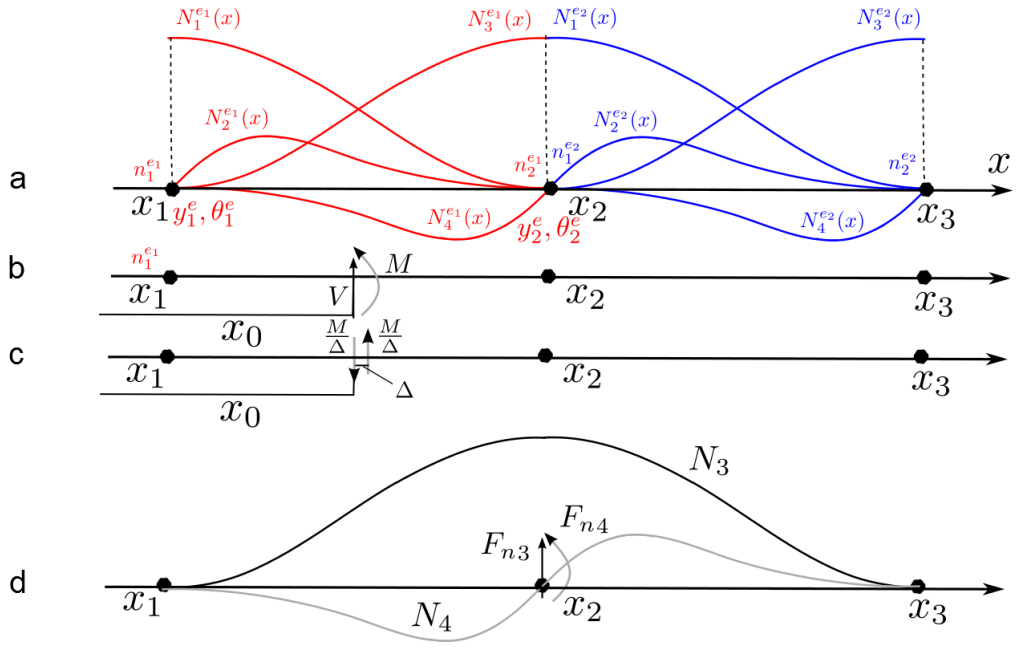
- From page discussion (*cf.* (414a))and previous slides:

$$\mathbf{f}_e^e = \mathbf{f}_r^e - \mathbf{f}_D^e$$

- We provided relations to obtain both $\mathbf{f}_r^e$ $\mathbf{f}_D^e$ in previous slides.
- The only remaining force, is the global dof forces which will be discussed next.

Beam elements: Concentrated load

- Our main goal is to obtain the contribution of concentrated vertical force and moment to global force vector for the beam problem.
- For each node, there are two dofs: vertical displacement and rotation. We expect the corresponding nodal forces to be concentrated vertical load and concentrated moment as they are dual quantities.



- In the figure (b) a concentrated vertical force V and moment M are exerted at point x.
- The source term force is:

$$\mathbf{F}_r = \int_{\mathcal{D}} \mathbf{N}^{\mathrm{T}}.\mathbf{r}\,\mathrm{dv},\ \mathcal{D} = [0\ L], \mathbf{r} = q \text{ (vertical distributed load)} \quad \Rightarrow$$

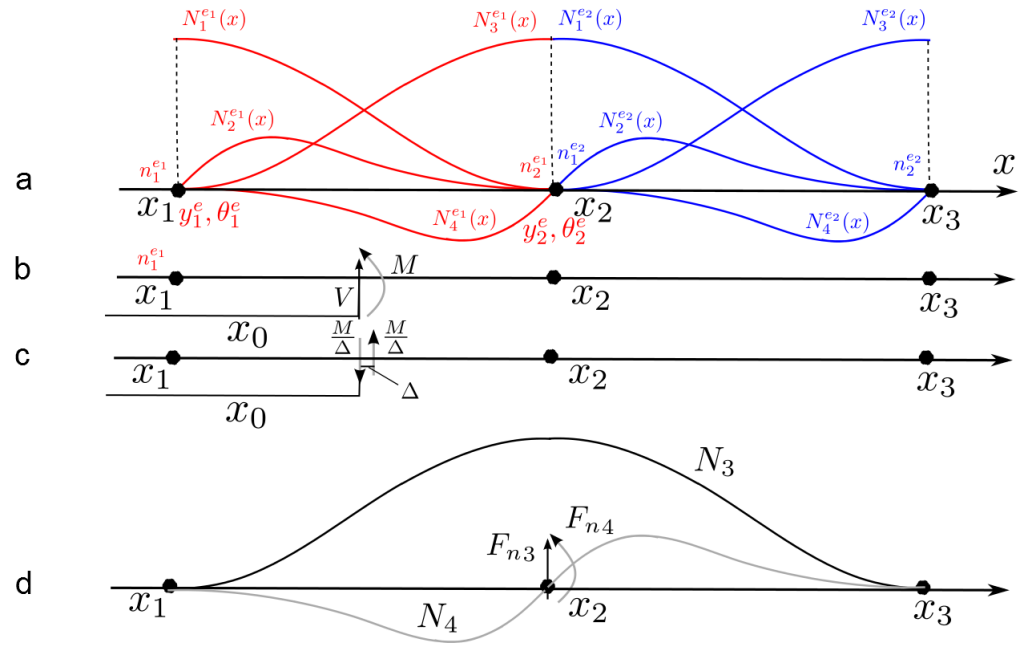
$$F_{r\,i} = \int_0^L N_i(x) q(x)\,\mathrm{d}x$$

Beam elements: Concentrated load

- Since, for a concentrated vertical load V applied at x_0 the corresponding q is $V\delta(x-x_0)$ the global force F_{ri} is:

$$F_{ri} = \int_0^L N_i(x)\{V\delta(x-x_0)\}\,\mathrm{d}x$$

$$F_{ri} = V N_i(x_0) \qquad (436)$$



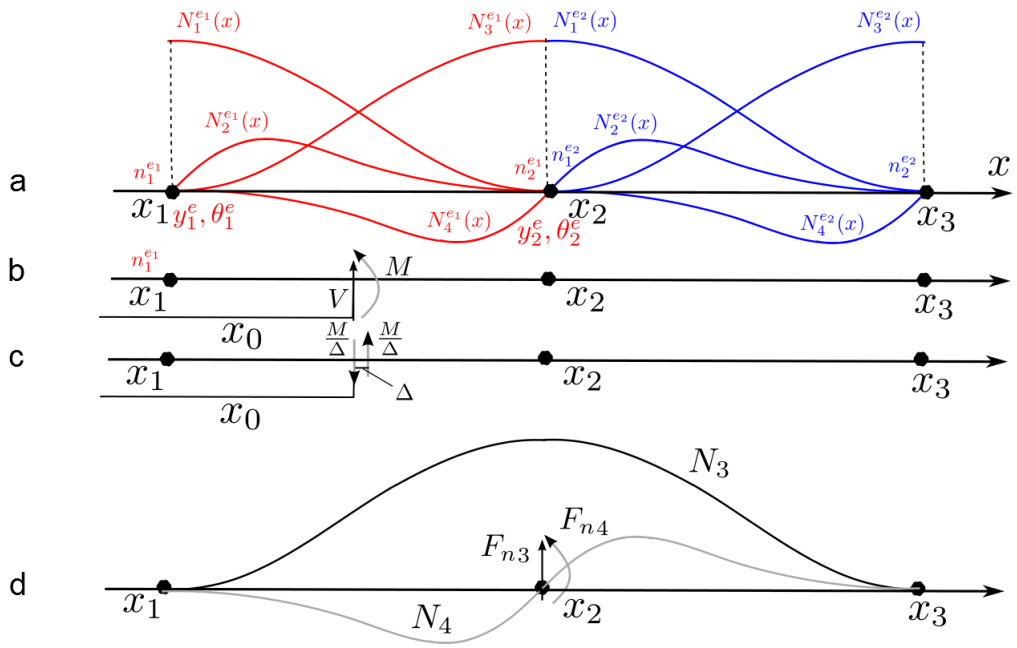
- If x_0 is inside an element (for example in figures (b) and (c)) the concentrated force can be added to $\mathbf{f}_r^e$:

$$\mathbf{f}_r^e = \int_0^{L^e} \mathbf{N}^{e\mathrm{T}}.q\,\mathrm{d}x = V \begin{bmatrix} N_1^e(\xi_0) \\ N_2^e(\xi_0) \\ N_3^e(\xi_0) \\ N_4^e(\xi_0) \end{bmatrix} \qquad (437)$$

- ξ_0 is the local coordinate corresponding to x_0.
- Off course, if in addition to V at x_0, there is the distributed load q applied on the beam, we add its contribution from the the one of variants on pages 358 to .

Beam elements: Concentrated load

- Next, we attempt to obtain global force corresponding to a moment M exerted at x_0.
- Moment M is not a force, even if considered as a delta function.
- We can approximate M with a couple of forces of magnitude $\frac{M}{\Delta}$ at distance Δ for $\Delta \ll 1$ (figure c):



couple system of forces $\frac{M}{\Delta}@\,(x_0+\Delta), -\frac{M}{\Delta}@\,x_0 \rightarrow$ moment $M@\,x_0$ as $\Delta \rightarrow 0$

- From (436) we already know the global force for a vertical load. Thus:

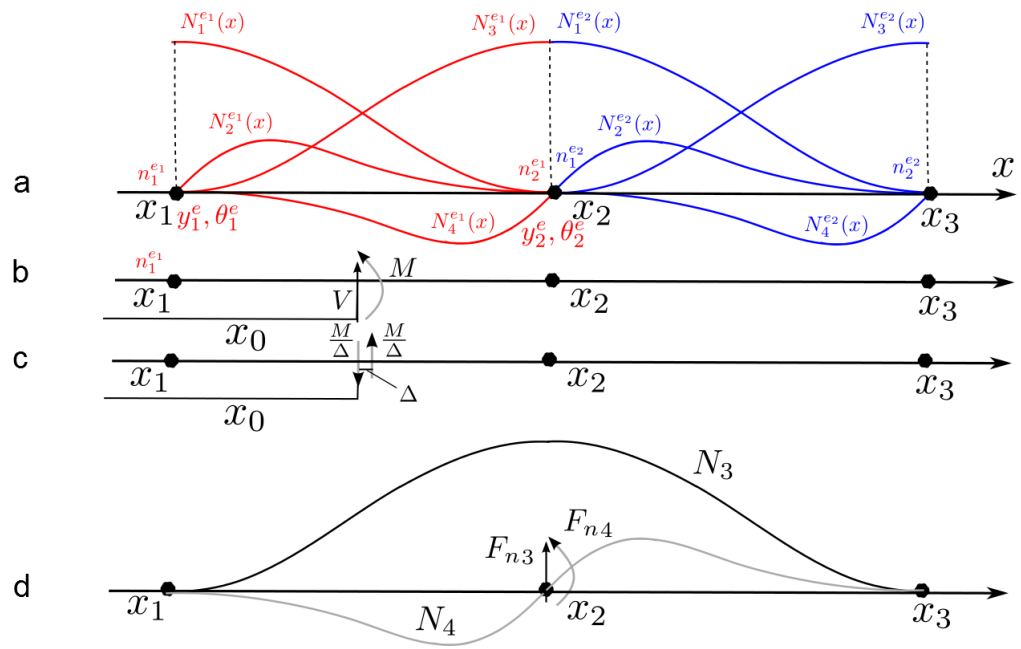
$$
\begin{aligned}
F_{ri}(M@\,x_0) &= \lim_{\Delta\to 0}\{F_{ri}(\frac{M}{\Delta}@\,x_0+\Delta) + F_{ri}(-\frac{M}{\Delta}@\,x_0)\} \\
&= \lim_{\Delta\to 0}\left\{\frac{M}{\Delta}N_i(x_0+\Delta) - \frac{M}{\Delta}N_i(x_0)\right\} = M\lim_{\Delta\to 0}\left(\frac{N_i(x_0+\Delta)-N_i(x_0)}{\Delta}\right) \\
&= M\frac{\mathrm{d}N_i}{\mathrm{d}x}(x_0)
\end{aligned}
\tag{438}
$$

Beam elements: Concentrated load

- Thus, as expected, from (438) we observe that the global force of moment is rotation of shape function at x_0.
- Equations (438) and (436) can be summarized as (figure (b)):

$$F_{ri} = V N_i(x_0) \text{ vertical force } V \; x_0 \tag{439a}$$

$$F_{ri} = M \frac{\mathrm{d}N_i}{\mathrm{d}x}(x_0) \text{ moment} M \; x_0 \tag{439b}$$



- Clearly, we superpose forces from all concentrated forces.
- For example, in figure (b) $F_{ri} = V N_i(x_0) + M \frac{\mathrm{d}N_i}{\mathrm{d}x}(x_0)$.
- If V and/or M are exerted within an element, *e.g.*, figure (b), their contributions to element source term force is (*cf.* (437)):

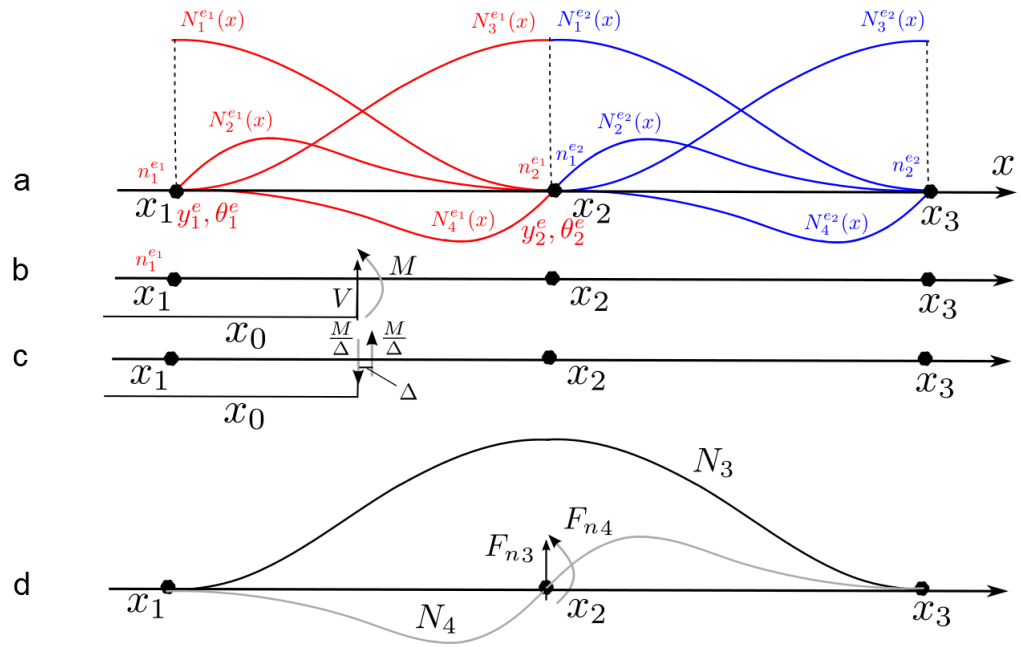
$$\mathbf{f}_r^e(V@\, x_0) = V \begin{bmatrix} N_1^e(\xi_0) \\ N_2^e(\xi_0) \\ N_3^e(\xi_0) \\ N_4^e(\xi_0) \end{bmatrix} \qquad \mathbf{f}_r^e(M@\, x_0) = M \begin{bmatrix} \frac{\mathrm{d}N_1^e}{\mathrm{d}x_e}(\xi_0) \\ \frac{\mathrm{d}N_2^e}{\mathrm{d}x_e}(\xi_0) \\ \frac{\mathrm{d}N_3^e}{\mathrm{d}x_e}(\xi_0) \\ \frac{\mathrm{d}N_4^e}{\mathrm{d}x}(\xi_0) \end{bmatrix} \tag{440}$$

Beam elements: Concentrated load

- How about concentrated forced and moments exerted at nodes?
- We should directly add their contribution to global nodal forces $\mathbf{F}_n$ (*cf.* (342d))

$$\mathbf{F}_n = \begin{bmatrix} F_{n1} & \cdots & F_{n n_\mathrm{f}} \end{bmatrix}^\mathrm{T}$$

- $\mathbf{F}_n$ is then added to force contribution from elements $\mathbf{F}_e$ to form total force $\mathbf{F} = \mathbf{F}_N + \mathbf{F}_e$ (*cf.* (361b))



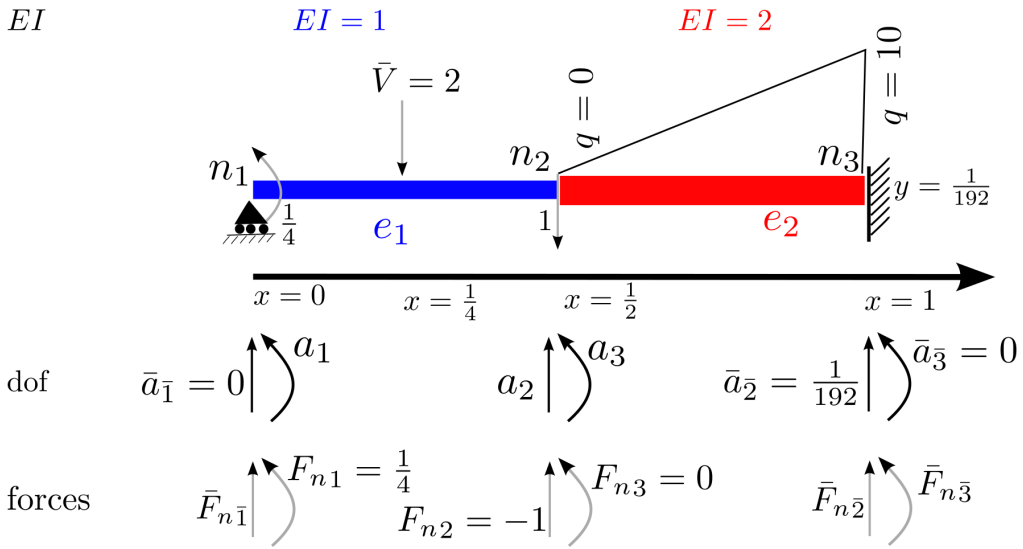
- Since shape functions take the value of one on their corresponding dof and zero at other dofs, from (439) F_{ni} corresponds to concentrated vertical force or moment for displacement or rotation dofs, respectively. Thus:
- For free dofs 1 to n_f:

$$\begin{aligned} \text{dof } i \text{ is vertical displacement:} \quad & F_{ni} = V_i (\text{vertical load at dof } i) \\ \text{dof } i \text{ is rotation:} \quad & F_{ni} = M_i (\text{moment at dof } i) \end{aligned} \tag{441}$$

- For example, in figure (d) on dofs 3 and 4 force F_{n3} and moment F_{n4} are applied, while there are no other nodal forces at free dofs. Thus,

$$\mathbf{F}_n = \begin{bmatrix} 0 & 0 & F_{n3} & F_{n4} & \cdots \end{bmatrix}^\mathrm{T}$$

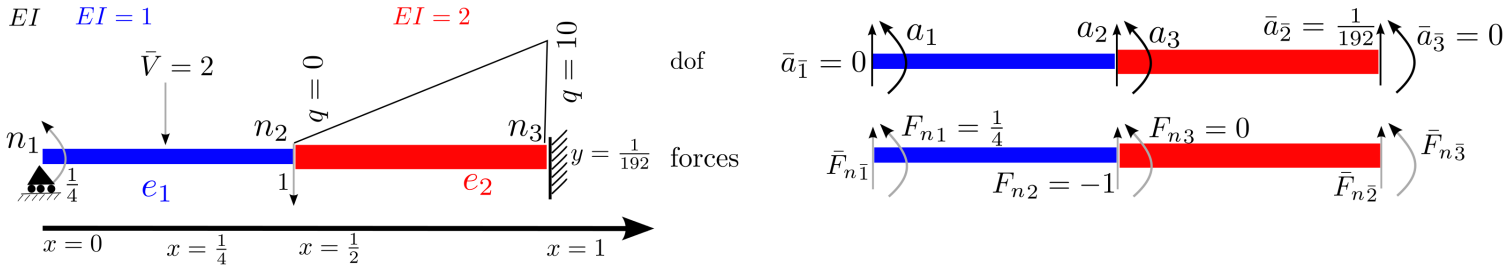
Beam Example



- The material properties of two elements are $EI = 1$, $EI = 2$, for elements e_1 and e_2, respectively.
- Lengths of elements are $L^{e_1} = \frac{1}{2}$ and $L^{e_2} = \frac{1}{2}$.
- Applied loads:
  1. Moment $\frac{1}{4}$ at $x = 0$.
  2. Downward $\bar{V}$ at $x = \frac{1}{4}$.
  3. Downward force of 1, $x = \frac{1}{2}$.
  4. Linear q: 0 to 10 on e_2.

- The supports and corresponding prescribed dofs are:
  1. Left: Roller, 0 prescribed vertical displacement ($\bar{a}_{\bar{1}} = 0$);
  2. Right: Fixed, prescribed vertical displacement ($\bar{a}_{\bar{2}} = \frac{1}{192}$); 0 rotation ($\bar{a}_{\bar{3}} = 0$).
- Free and prescribed dofs and forces are shown in the bottom two rows of the figure.
- Stiffness matrix: constant EI elements $\rightarrow$ stiffness matrices are computed from (427).
- Element forces:
  1. $\mathbf{f}_r^e$: i) e_1: concentrated $\bar{V}$ $\rightarrow$ first expression in (440); ii) e_2: linear q $\rightarrow$ (433).
  2. $\mathbf{f}_D^e$: i) e_1: zero prescribed dof values $\rightarrow$ $\mathbf{f}_D^e = 0$. ii) nonzero $a_3^e = \frac{1}{192}$ $\rightarrow$ $\mathbf{f}_D^e = \mathbf{k}^e\mathbf{a}^e$ *cf.* (414c).
  3. $\mathbf{f}_N^e$: 1D elements natural BC forces are incorporated into global nodal forces $\mathbf{F}_n$.
- Global force: $\mathbf{F}_n = \begin{bmatrix} \frac{1}{4} & -1 & 0 \end{bmatrix}^{\mathrm{T}}$ (*cf.* (439)).

Beam example: Assembly of global system



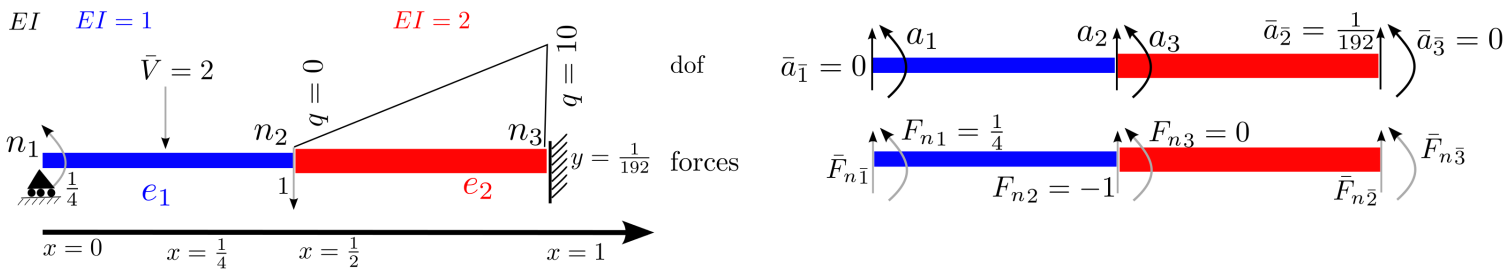
e	e_1	e_2
$\mathbf{k}^e$	$\mathbf{k}^{e1} = \frac{1}{(\frac{1}{2})^3}\begin{bmatrix} 12 & 3 & -12 & 3 \\ 3 & 1 & -3 & 0.5 \\ -12 & -3 & 12 & -3 \\ 3 & 0.5 & -3 & 1 \end{bmatrix}$ $= \begin{array}{c} \\ \bar{1} \\ 1 \\ 2 \\ 3 \end{array}\begin{array}{c} \begin{array}{cccc} \bar{1} & 1 & 2 & 3 \end{array} \\ \begin{bmatrix} 96 & 24 & -96 & 24 \\ 24 & 8 & -24 & 4 \\ -96 & -24 & 96 & -24 \\ 24 & 4 & -24 & 8 \end{bmatrix} \end{array}$	$\mathbf{k}^{e2} = \frac{2}{(\frac{1}{2})^3}\begin{bmatrix} 12 & 3 & -12 & 3 \\ 3 & 1 & -3 & 0.5 \\ -12 & -3 & 12 & -3 \\ 3 & 0.5 & -3 & 1 \end{bmatrix}$ $= \begin{array}{c} \\ 2 \\ 3 \\ \bar{2} \\ \bar{3} \end{array}\begin{array}{c} \begin{array}{cccc} 2 & 3 & \bar{2} & \bar{3} \end{array} \\ \begin{bmatrix} 192 & 48 & -192 & 48 \\ 48 & 16 & -48 & 8 \\ -192 & -48 & 192 & -48 \\ 48 & 8 & -48 & 16 \end{bmatrix} \end{array}$
$\mathbf{f}^e_r$	(440) (1st eqn) ($\xi = 0$) $\bar{V}\begin{bmatrix} N_1^{e1}(\xi_0) \\ N_2^{e1}(\xi_0) \\ N_3^{e1}(\xi_0) \\ N_4^{e1}(\xi_0) \end{bmatrix} = -2\begin{bmatrix} \frac{1}{2} \\ \frac{1}{8} \\ \frac{1}{2} \\ -\frac{1}{8} \end{bmatrix} = \begin{array}{c} \\ \bar{1} \\ 1 \\ 2 \\ 3 \end{array}\begin{bmatrix} -1 \\ -\frac{1}{8} \\ -1 \\ \frac{1}{8} \end{bmatrix}$	equation (433); $\mathbf{r}^e\ [q_1 \quad q_2]^{\mathrm{T}}$ $\frac{1}{2}\begin{bmatrix} \frac{7}{20} & \frac{3}{20} \\ \frac{1}{40} & \frac{1}{60} \\ \frac{3}{20} & \frac{7}{20} \\ -\frac{1}{60} & -\frac{1}{40} \end{bmatrix}\begin{bmatrix} 0 \\ 10 \end{bmatrix} = \begin{array}{c} \\ 2 \\ 3 \\ \bar{2} \\ \bar{3} \end{array}\begin{bmatrix} \frac{3}{4} \\ \frac{1}{12} \\ \frac{7}{4} \\ -\frac{1}{8} \end{bmatrix}$
$\mathbf{f}^e_D$	$\mathbf{k}^{e1}\mathbf{a}^e_1 =$ $\begin{bmatrix} 96 & 24 & -96 & 24 \\ 24 & 8 & -24 & 4 \\ -96 & -24 & 96 & -24 \\ 24 & 4 & -24 & 8 \end{bmatrix}\begin{bmatrix} 0 \\ 0 \\ 0 \\ 0 \end{bmatrix} = \begin{array}{c} \\ 1 \\ \bar{1} \\ 2 \\ 3 \end{array}\begin{bmatrix} 0 \\ 0 \\ 0 \\ 0 \end{bmatrix}$	$\mathbf{k}^{e2}\mathbf{a}^e_2 =$ $\begin{bmatrix} 192 & 48 & -192 & 48 \\ 48 & 16 & -48 & 8 \\ -192 & -48 & 192 & -48 \\ 48 & 8 & -48 & 16 \end{bmatrix}\begin{bmatrix} 0 \\ 0 \\ \frac{1}{192} \\ 0 \end{bmatrix} = \begin{array}{c} \\ 2 \\ 3 \\ \bar{2} \\ \bar{3} \end{array}\begin{bmatrix} -1 \\ -\frac{1}{4} \\ 1 \\ -\frac{1}{4} \end{bmatrix}$
$\mathbf{f}^e_e$	$\mathbf{f}^{e1}_e = \mathbf{f}^{e1}_r + \mathbf{f}^{e1}_N - \mathbf{f}^{e1}_D = \begin{array}{c} \\ \bar{1} \\ 1 \\ 2 \\ 3 \end{array}\begin{bmatrix} -1 \\ -\frac{1}{8} \\ -1 \\ \frac{1}{8} \end{bmatrix}$	$\mathbf{f}^{e2}_e = \mathbf{f}^{e2}_r + \mathbf{f}^{e2}_N - \mathbf{f}^{e2}_D = \begin{array}{c} \\ 2 \\ 3 \\ \bar{2} \\ \bar{3} \end{array}\begin{bmatrix} \frac{7}{4} \\ \frac{1}{3} \\ \frac{3}{4} \\ \frac{1}{8} \end{bmatrix}$

$$\mathbf{K} = \begin{bmatrix} 8 & -24 & 4 \\ & 96+192 & -24+48 \\ \text{sym.} & & 8+16 \end{bmatrix} = \begin{bmatrix} 8 & -24 & 4 \\ & 288 & 24 \\ \text{sym.} & & 24 \end{bmatrix}$$

$$\mathbf{F} = \mathbf{F}_n + \mathbf{F}_e = \begin{bmatrix} \frac{1}{4} \\ -1 \\ 0 \end{bmatrix} + \begin{bmatrix} -\frac{1}{8} \\ -1+\frac{7}{4} \\ \frac{1}{8}+\frac{1}{3} \end{bmatrix} = \begin{bmatrix} -\frac{1}{8} \\ -\frac{3}{4} \\ \frac{11}{24} \end{bmatrix} \Rightarrow$$

$$\mathbf{U} = \mathbf{K}^{-1}\mathbf{F} = \begin{bmatrix} a_1 \\ a_2 \\ a_3 \end{bmatrix} = \begin{bmatrix} -\frac{7}{1152} \\ -\frac{23}{6912} \\ \frac{3}{128} \end{bmatrix} = \begin{bmatrix} -.00607 \\ -.00332 \\ .023434 \end{bmatrix}$$

Beam Example: Resultant nodal forces



e	e_1	e_2
$\mathbf{u}^e$	$\begin{bmatrix} 0 \\ -\frac{7}{1152} \\ -\frac{23}{6912} \\ \frac{3}{128} \end{bmatrix}$	$\begin{bmatrix} -\frac{23}{6912} \\ \frac{3}{128} \\ \frac{1}{192} \\ 0 \end{bmatrix}$
$-f^e$	$\mathbf{k}^{e1}\mathbf{a}_1^e - \mathbf{f}_r^{e1} - \mathbf{f}_N^{e1} =$ $\begin{bmatrix} 96 & 24 & -96 & 24 \\ 24 & 8 & -24 & 4 \\ -96 & -24 & 96 & -24 \\ 24 & 4 & -24 & 8 \end{bmatrix} \begin{bmatrix} 0 \\ -\frac{7}{1152} \\ -\frac{23}{6912} \\ \frac{3}{128} \end{bmatrix} - \begin{bmatrix} -1 \\ -\frac{1}{8} \\ -1 \\ \frac{1}{8} \end{bmatrix} = \begin{matrix} \bar{1} \\ 1 \\ 2 \\ 3 \end{matrix} \begin{bmatrix} \frac{125}{72} \\ \frac{1}{4} \\ \frac{19}{72} \\ \frac{17}{144} \end{bmatrix}$	$\mathbf{k}^{e2}\mathbf{a}_2^e - \mathbf{f}_r^{e2} - \mathbf{f}_N^{e2} =$ $\begin{bmatrix} 192 & 48 & -192 & 48 \\ 48 & 16 & -48 & 8 \\ -192 & -48 & 192 & -48 \\ 48 & 8 & -48 & 16 \end{bmatrix} \begin{bmatrix} -\frac{23}{6912} \\ \frac{3}{128} \\ \frac{1}{192} \\ 0 \end{bmatrix} - \begin{bmatrix} \frac{3}{4} \\ \frac{1}{12} \\ \frac{7}{4} \\ -\frac{1}{8} \end{bmatrix} = \begin{matrix} 2 \\ 3 \\ \bar{2} \\ \bar{3} \end{matrix} \begin{bmatrix} -\frac{91}{72} \\ -\frac{17}{144} \\ -\frac{89}{72} \\ -\frac{7}{72} \end{bmatrix}$

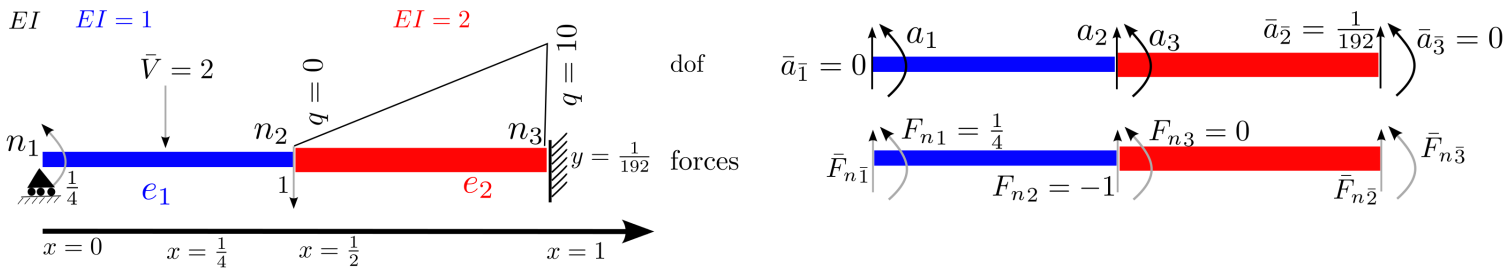
- **Prescribed forces $\mathbf{F}_p$ (Reaction forces)**:

$$\bar{F}_{n\bar{1}} = f_1^{e1} = \frac{125}{72} = 1.736111 \quad \text{vertical load at the left support} \tag{442a}$$

$$\bar{F}_{n\bar{2}} = f_3^{e2} = -\frac{89}{72} = -1.236111 \quad \text{vertical load at the right support} \tag{442b}$$

$$\bar{F}_{n\bar{3}} = f_4^{e2} = -\frac{7}{72} = -0.0972222 \quad \text{CCW moment at the right support} \tag{442c}$$

Beam Example: Verification of forces at free dofs



- **Free forces $\mathbf{F}_f$:**

$$F_{n1} = f_2^{e1} = \frac{1}{4} \quad = \text{moment exerted on the left support} \tag{443a}$$

$$F_{n2} = f_3^{e1} + f_1^{e2} = \frac{19}{72} - \frac{91}{72} = -1 \quad = \text{vertical load exerted at the center of the beam} \tag{443b}$$

$$F_{n3} = f_4^{e1} + f_2^{e2} = \frac{17}{144} - \frac{17}{144} = 0 \quad = \text{zero moment exerted at the center of the beam} \tag{443c}$$

- As mentioned before, this step is not necessary and may just be used as a necessary (not sufficient) condition for the correctness of hand calculations.

Beam Example: Calculation of y, θ, M, V within element

- After the solution of global free dofs, they are transferred to elements.
- Once element dofs are know, we have the Displacement in the entire elements:

$$y(\xi) = N_1^e(\xi)a_1^e + N_2^e(\xi)a_2^e + N_3^e(\xi)a_3^e + N_4^e(\xi)a_4^e.$$

element shape functions are given in (421).

- Rotation: Obtained by differentiating previous equation w.r.t. x & noting that $\frac{\mathrm{d}x}{\mathrm{d}\xi} = \frac{L^e}{2}$:

$$\theta(\xi) = \frac{\mathrm{d}y}{\mathrm{d}x}(\xi) = \frac{\frac{\mathrm{d}y}{\mathrm{d}\xi}(\xi)}{\frac{\mathrm{d}x}{\mathrm{d}\xi}(\xi)} = \frac{2}{L^e}\left\{\frac{\mathrm{d}N_1^e}{\mathrm{d}\xi}(\xi)a_1^e + \frac{\mathrm{d}N_2^e}{\mathrm{d}\xi}(\xi)a_2^e + \frac{\mathrm{d}N_3^e}{\mathrm{d}\xi}(\xi)a_3^e + \frac{\mathrm{d}N_4^e}{\mathrm{d}\xi}(\xi)a_4^e\right\}$$

- Moment is directly obtained by differentiating the above equation:

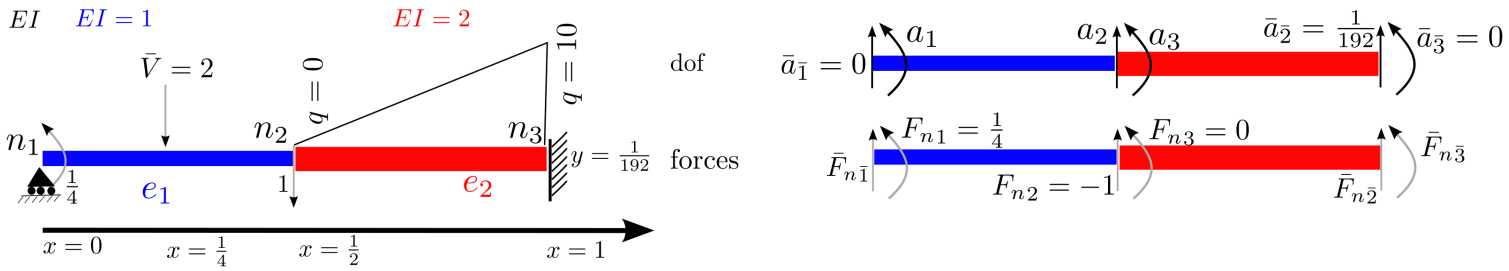
$$\begin{aligned} M(\xi) &= E(\xi)I(\xi)\frac{\mathrm{d}^2y}{\mathrm{d}x^2}(\xi) = E(\xi)I(\xi)\mathbf{B}^e(\xi) \\ &= E(\xi)I(\xi)\left\{B_1^e(\xi)a_1^e + B_2^e(\xi)a_2^e + B_3^e(\xi)a_3^e + B_4^e(\xi)a_4^e\right\} \quad \textit{cf.}\ (424)\ \text{for}\ \mathbf{B}^e \end{aligned}$$

- Shear force is obtained by differentiating M w.r.t. x. It's a similar process to deriving θ from y with the difference that if EI are not constant we need to take it into account. For constant EI we have:

$$V(\xi) = \frac{\mathrm{d}M}{\mathrm{d}x}(\xi) = \frac{\frac{\mathrm{d}M}{\mathrm{d}\xi}(\xi)}{\frac{\mathrm{d}x}{\mathrm{d}\xi}(\xi)} = \frac{2EI}{L^e}\left\{\frac{\mathrm{d}B_1^e}{\mathrm{d}\xi}(\xi)a_1^e + \frac{\mathrm{d}B_2^e}{\mathrm{d}\xi}(\xi)a_2^e + \frac{\mathrm{d}B_3^e}{\mathrm{d}\xi}(\xi)a_3^e + \frac{\mathrm{d}B_4^e}{\mathrm{d}\xi}(\xi)a_4^e\right\}$$

- To obtain these fields for the entire beam we evaluate these equations for all elements.

Beam Example: Exact solution



- This is an indeterminate system of equations (degree 1, *i.e.*, one unknown more than statistically determinate structure).
- The exact solution for this problem is:

$$y(x) = \begin{cases} -\frac{7}{1152}x - \frac{1}{8}x^2 + \frac{125}{432}x^3 - \frac{1}{3}\left\langle x - \frac{1}{4}\right\rangle_+^3 & x \le \frac{1}{2} \\ -\frac{13}{1728} + \frac{5}{576}x - \frac{1}{16}x^2 + \frac{125}{864}x^3 - \frac{1}{6}(x - \frac{1}{4})^3 - \frac{1}{12}\left(x - \frac{1}{2}\right)^3 + \frac{1}{12}\left(x - \frac{1}{2}\right)^5 & x > \frac{1}{2} \end{cases} \tag{444a}$$

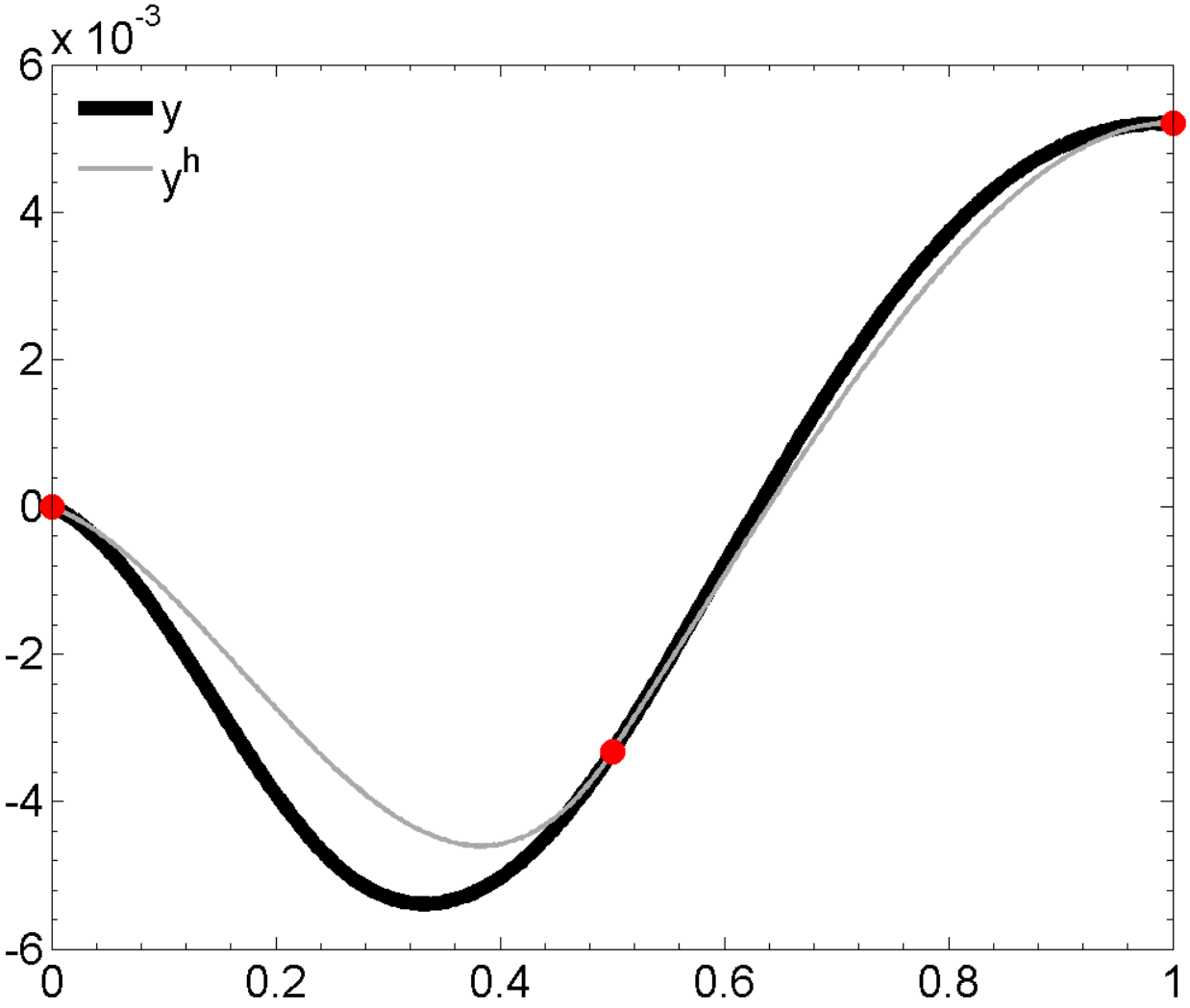
$$\theta(x) = \frac{\mathrm{d}y}{\mathrm{d}x} = \begin{cases} -\frac{7}{1152} - \frac{1}{4}x + \frac{125}{144}x^2 - \left\langle x - \frac{1}{4}\right\rangle_+^2 & x \le \frac{1}{2} \\ \frac{5}{576} - \frac{1}{8}x + \frac{125}{288}x^2 - \frac{1}{2}(x - \frac{1}{4})^2 - \frac{1}{4}\left(x - \frac{1}{2}\right)^2 + \frac{5}{12}\left(x - \frac{1}{2}\right)^4 & x > \frac{1}{2} \end{cases} \tag{444b}$$

$$M(x) = E(x)I(x)\frac{\mathrm{d}^2y}{\mathrm{d}x^2} = \begin{cases} -\frac{1}{4} + \frac{125}{72}x - 2\left\langle x - \frac{1}{4}\right\rangle_+ & x \le \frac{1}{2} \\ \frac{3}{4} - \frac{91}{72}x + \frac{10}{3}(x - \frac{1}{2})^3 & x > \frac{1}{2} \end{cases} \tag{444c}$$

$$V(x) = \frac{\mathrm{d}M}{\mathrm{d}x} = \frac{\mathrm{d}}{\mathrm{d}x}\left\{E(x)I(x)\frac{\mathrm{d}^2y}{\mathrm{d}x^2}\right\} = \begin{cases} \frac{125}{72}x - 2\mathrm{sign}\left(x - \frac{1}{4}\right) & x \le \frac{1}{2} \\ -\frac{91}{72}x + 10(x - \frac{1}{2})^2 & x > \frac{1}{2} \end{cases} \tag{444d}$$

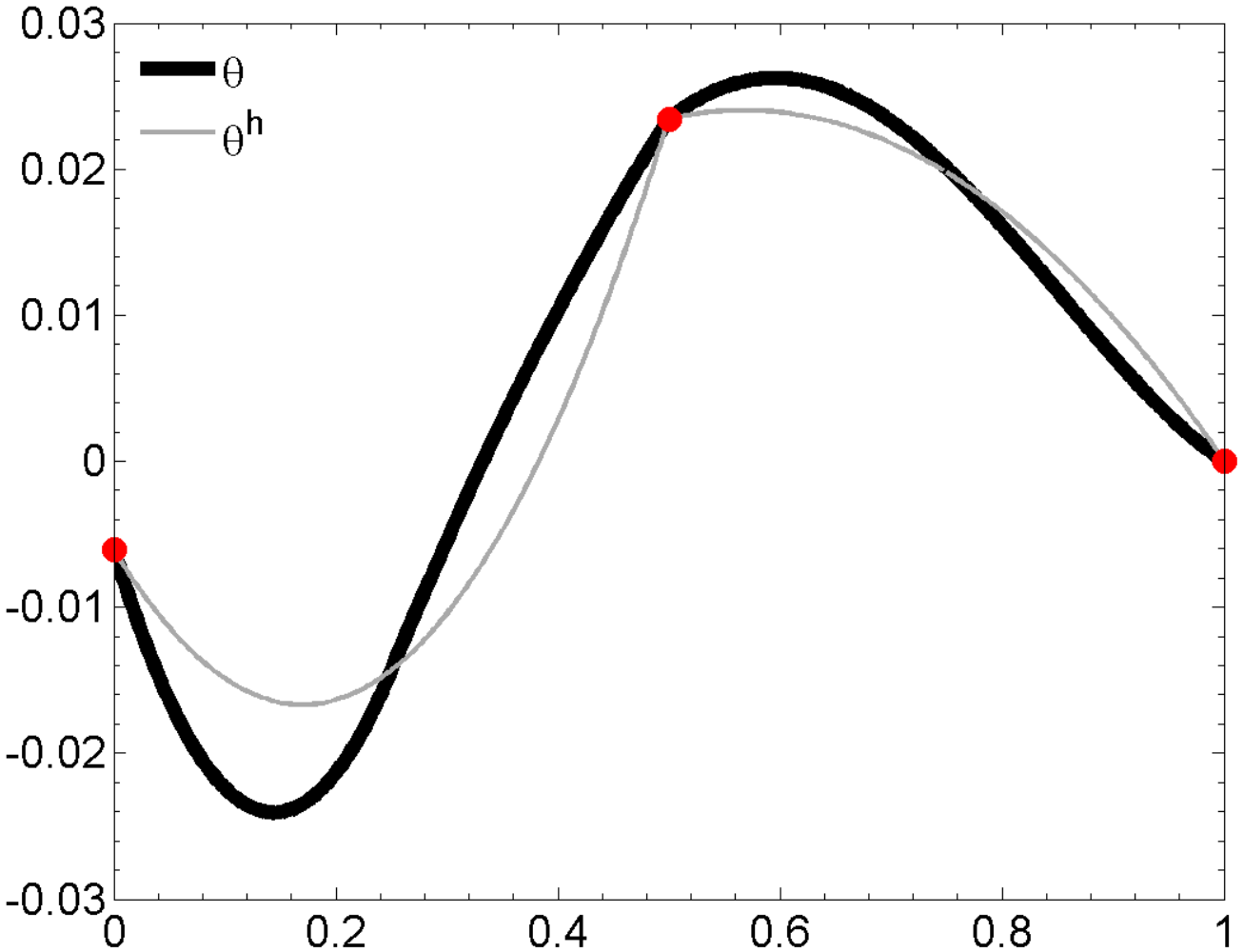
where $\langle(.)\rangle_+$ and $\mathrm{sign}\left((.)\right)$ are Macaulay bracket (positive part) and sign function, respectively.

Beam Example: FEM vs. Exact solution: y



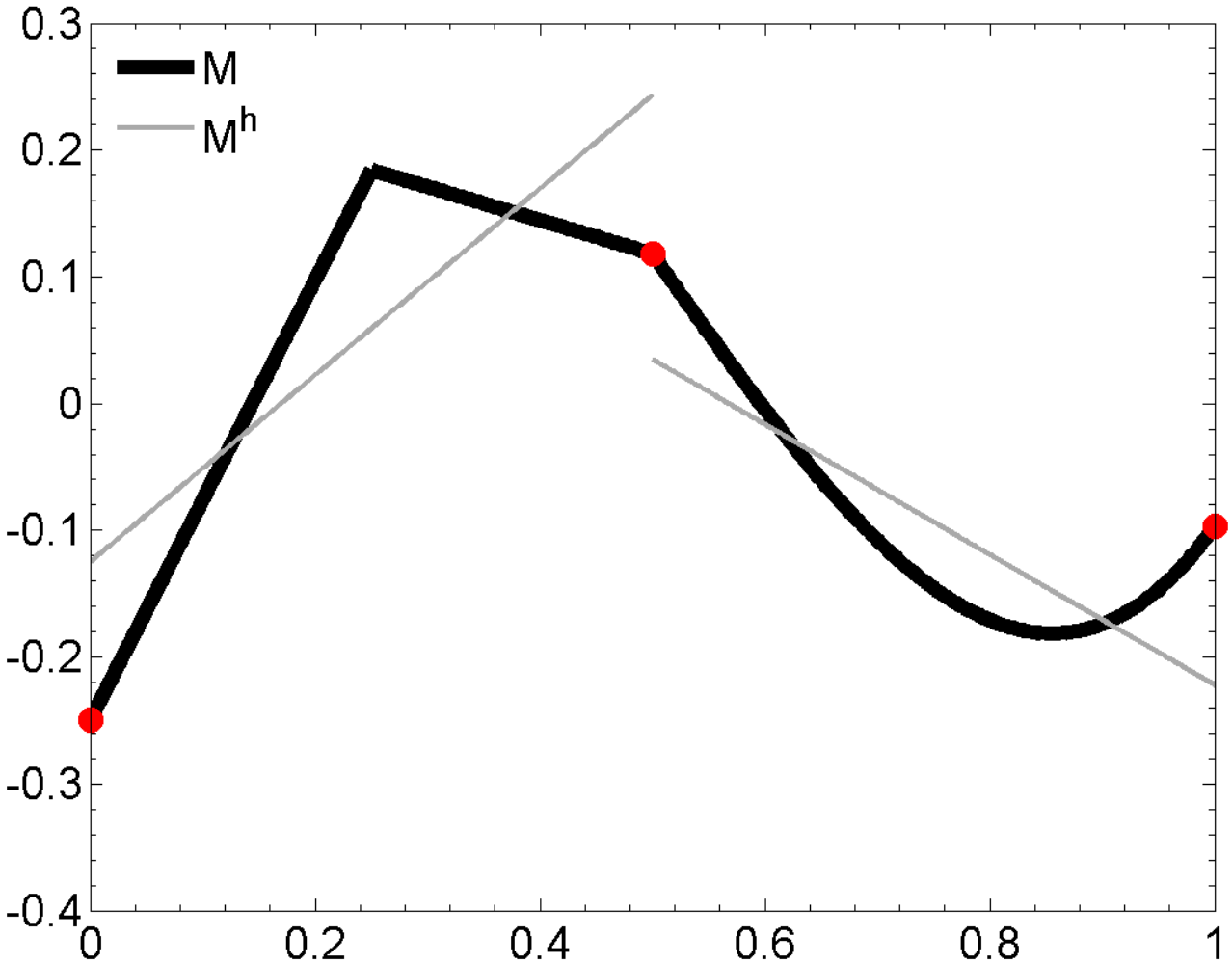
- Similar to bar element FEM and exact solutions match at nodes.
- This behavior is restricted to certain problems in 1D with constant material properties along the element and does not extend to more general cases.

Beam Example: FEM vs. Exact solution: θ



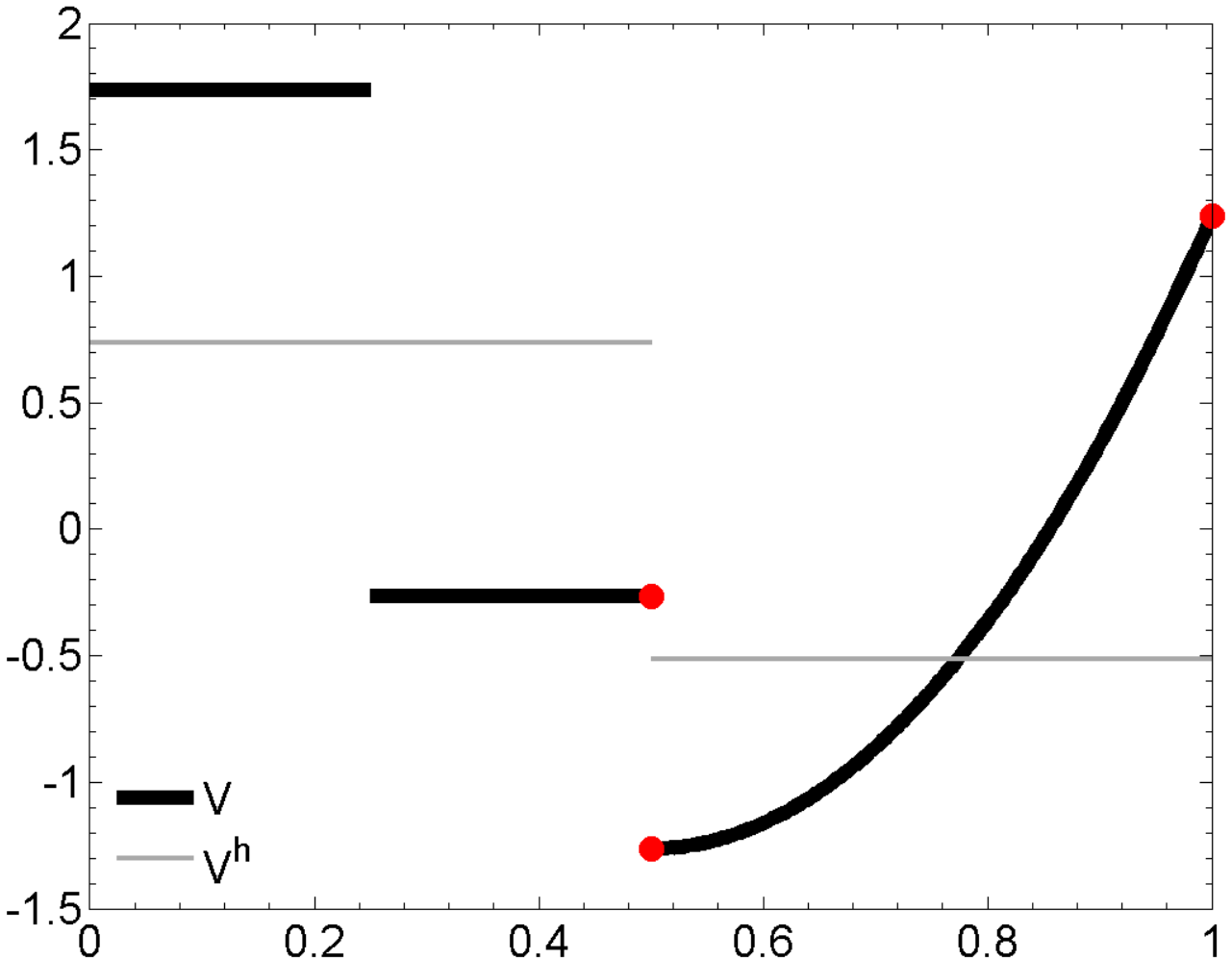
- Rotations at nodes (rotational dofs) match those from exact solution.
- Again we emphasize that while this behavior is shared for certain types of 1D problems, it does not extend to more general cases.

Beam Example: FEM vs. Exact solution: M



- It is clear that FEM solutions for M are much less accurate than those for y and θ when compared to exact solution.
- This is a general behavior where FEM accuracy decreases for solution derivatives.

Beam Example: FEM vs. Exact solution: V



Beam Example: FEM vs. Exact: Summary & Reaction forces

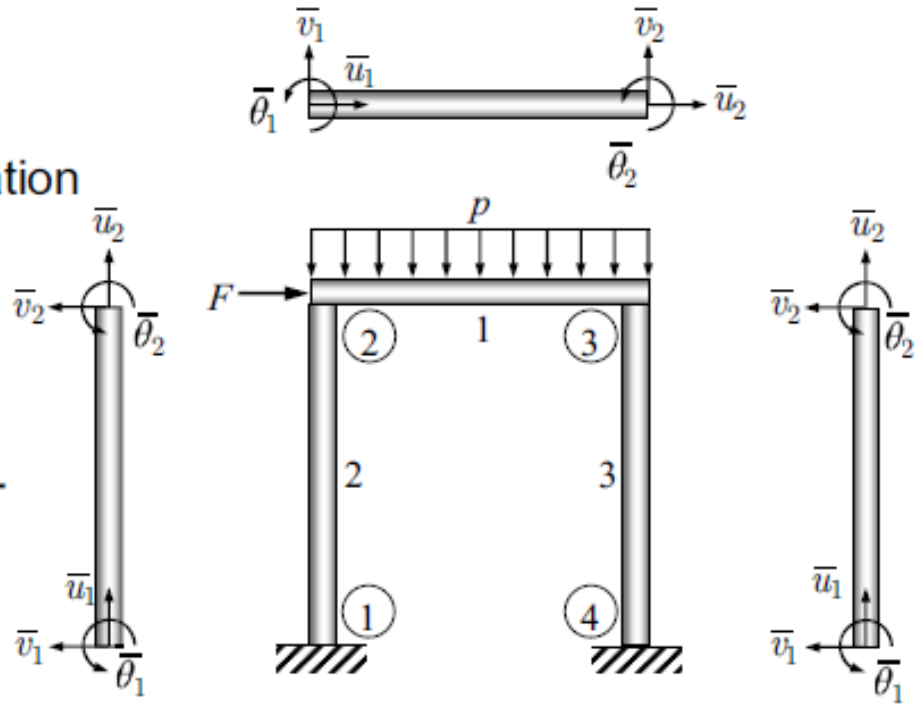
- We observed in previous two slides the shear force and moment within the elements do not match the exact solution.
- This is because the solution field in the right element is 5th order which cannot be captured by 3rd order shape functions.
- Once the contributions of source terms are added for element nodal forces and moments, *cf.* pages 372 to 373, element end point forces and moments match the exact solution.
- We observed before that y and θ also match the exact solution at element end points.
- Table below summaries y, θ, M, V at element end points. All match the exact solution.

$x = 0^+$	field	$x = \frac{1}{2}^-$	$x = \frac{1}{2}^+$	field	$x = 1^{-1}$
0	y	$-\frac{23}{6912}$	$-\frac{23}{6912}$	y	$\frac{1}{192}$
$-\frac{7}{1152}$	θ	$\frac{3}{128}$	$\frac{3}{128}$	θ	0
$-\frac{1}{4}$	M	$\frac{17}{144}$	$\frac{17}{144}$	M	$-\frac{7}{72}$
$\frac{125}{72}$	V	$-\frac{19}{72}$	$-\frac{91}{72}$	V	$\frac{89}{72}$

Frames: 2D frame elements

- Beam
  - Vertical deflection and slope. No axial deformation
- Frame structure
  - Can carry axial force, transverse shear force, and bending moment (Beam + Truss)
- Assumption
  - Axial and bending effects are uncoupled
  - Reasonable when deformation is small
- 3 DOFs per node

  $\{u_i, v_i, \theta_i\}$
- Need coordinate transformation like plane truss



source: Nam-Ho Kim, Raphael T. Haftka; http://www2.mae.ufl.edu/nkim/eml5526/ sec. 5

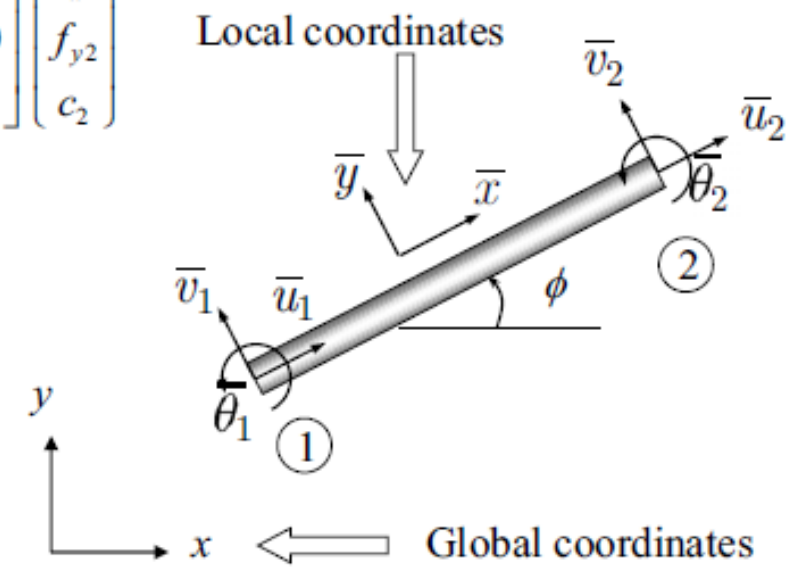
2D frame element: coordinate transformation

- Element-fixed local coordinates $\bar{x} - \bar{y}$
- Local DOFs $\{\bar{u}, \bar{v}, \bar{\theta}\}$ Local forces $\{f_{\bar{x}}, f_{\bar{y}}, \bar{c}\}$
- Transformation between local and global coord.

$$\begin{Bmatrix} f_{\bar{x}1} \\ f_{\bar{y}1} \\ \bar{c}_1 \\ f_{\bar{x}2} \\ f_{\bar{y}2} \\ \bar{c}_2 \end{Bmatrix} = \begin{bmatrix} \cos\phi & \sin\phi & 0 & 0 & 0 & 0 \\ -\sin\phi & \cos\phi & 0 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & \cos\phi & \sin\phi & 0 \\ 0 & 0 & 0 & -\sin\phi & \cos\phi & 0 \\ 0 & 0 & 0 & 0 & 0 & 1 \end{bmatrix} \begin{Bmatrix} f_{x1} \\ f_{y1} \\ c_1 \\ f_{x2} \\ f_{y2} \\ c_2 \end{Bmatrix}$$

$$\{\bar{\mathbf{f}}\} = [\mathbf{T}]\{\mathbf{f}\}$$

$$\{\bar{\mathbf{q}}\} = [\mathbf{T}]\{\mathbf{q}\}$$



source: Nam-Ho Kim, Raphael T. Haftka; http://www2.mae.ufl.edu/nkim/eml5526/ sec. 5

2D frame element: axial and bending stiffness matrices

- Axial deformation (in local coord.)

$$\frac{EA}{L}\begin{bmatrix} 1 & -1 \\ -1 & 1 \end{bmatrix}\begin{Bmatrix} \overline{u}_1 \\ \overline{u}_2 \end{Bmatrix} = \begin{Bmatrix} f_{\overline{x}1} \\ f_{\overline{x}2} \end{Bmatrix}$$

- Beam bending

$$\frac{EI}{L^3}\begin{bmatrix} 12 & 6L & -12 & 6L \\ 6L & 4L^2 & -6L & 2L^2 \\ -12 & -6L & 12 & -6L \\ 6L & 2L^2 & -6L & 4L^2 \end{bmatrix}\begin{Bmatrix} \overline{v}_1 \\ \overline{\theta}_1 \\ \overline{v}_2 \\ \overline{\theta}_2 \end{Bmatrix} = \begin{Bmatrix} f_{\overline{y}1} \\ \overline{c}_1 \\ f_{\overline{y}2} \\ \overline{c}_2 \end{Bmatrix}$$

- Basically, it is equivalent to overlapping a beam with a bar
- A frame element has 6 DOFs

source: Nam-Ho Kim, Raphael T. Haftka; http://www2.mae.ufl.edu/nkim/eml5526/ sec. 5

2D frame: local & global coordinate stiffness matrices

- Element matrix equation (local coord.)

$$\begin{bmatrix} a_1 & 0 & 0 & -a_1 & 0 & 0 \\ 0 & 12a_2 & 6La_2 & 0 & -12a_2 & 6La_2 \\ 0 & 6La_2 & 4L^2a_2 & 0 & -6La_2 & 2L^2a_2 \\ -a_1 & 0 & 0 & a_1 & 0 & 0 \\ 0 & -12a_2 & -6La_2 & 0 & 12a_2 & -6La_2 \\ 0 & 6La_2 & 2L^2a_2 & 0 & -6La_2 & 4L^2a_2 \end{bmatrix} \begin{Bmatrix} \bar{u}_1 \\ \bar{v}_1 \\ \bar{\theta}_1 \\ \bar{u}_2 \\ \bar{v}_2 \\ \bar{\theta}_2 \end{Bmatrix} = \begin{Bmatrix} \bar{f}_{x1} \\ \bar{f}_{y1} \\ \bar{c}_1 \\ \bar{f}_{x2} \\ \bar{f}_{y2} \\ \bar{c}_2 \end{Bmatrix} \qquad a_1 = \frac{EA}{L} \qquad a_2 = \frac{EI}{L^3}$$

$$[\bar{\mathbf{k}}]\{\bar{\mathbf{q}}\} = \{\bar{\mathbf{f}}\}$$

- Element matrix equation (global coord.)

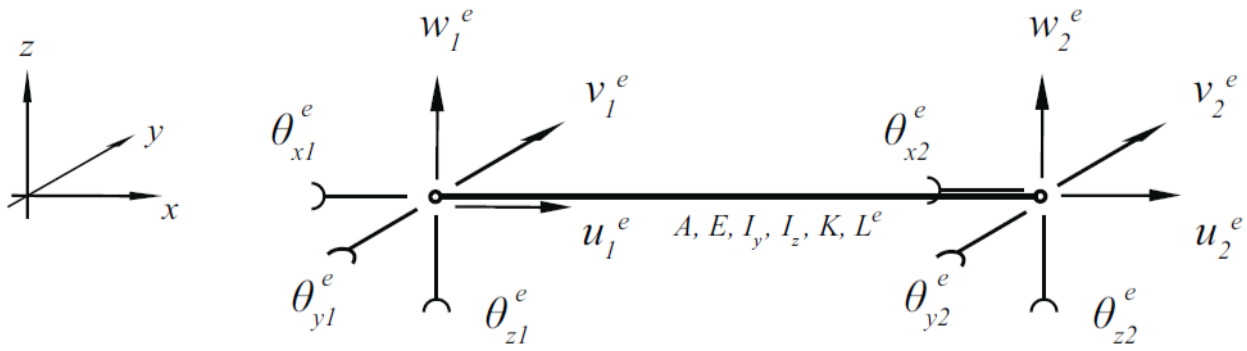
$$[\bar{\mathbf{k}}][\mathbf{T}]\{\mathbf{q}\} = [\mathbf{T}]\{\mathbf{f}\} \Longrightarrow [\mathbf{T}]^T[\bar{\mathbf{k}}][\mathbf{T}]\{\mathbf{q}\} = \{\mathbf{f}\} \Longrightarrow [\mathbf{k}]\{\mathbf{q}\} = \{\mathbf{f}\}$$

$$[\mathbf{k}] = [\mathbf{T}]^T[\bar{\mathbf{k}}][\mathbf{T}]$$

- Same procedure for assembly and applying BC

source: Nam-Ho Kim, Raphael T. Haftka; http://www2.mae.ufl.edu/nkim/eml5526/ sec. 5

3D frame element: local system



A 3D 12-freedom beam element defined in a local system

source: B. Torstenfelt; http://www.solid.iei.liu.se/Education/TMHL02/Book-Bars%20and%20Beams.pdf

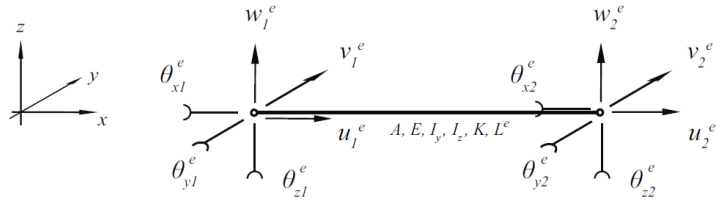
- 12 local dofs are:

$$\mathbf{a}^{e\,\mathrm{T}} = \{u_1^e\ v_1^e\ w_1^e\ \theta_{x1}^e\ \theta_{y1}^e\ \theta_{z1}^e\ u_2^e\ v_2^e\ w_2^e\ \theta_{x2}^e\ \theta_{y2}^e\ \theta_{z2}^e\} \tag{445}$$

- They correspond to axial and bending displacements, and torsional and bending rotations:

$u_1^e\ v_1^e\ w_1^e\ \theta_{x1}^e\ \theta_{y1}^e\ \theta_{z1}^e$	dofs of node n_1	(446a)
$u_2^e\ v_2^e\ w_2^e\ \theta_{x2}^e\ \theta_{y2}^e\ \theta_{z2}^e$	dofs of node n_2	(446b)
$\{u_1^e\ u_2^e\}$	axial displacements	(446c)
$\{v_1^e\ w_1^e\ v_2^e\ w_2^e\}$	vertical displacements	(446d)
$\{\theta_{x1}^e\ \theta_{x2}^e\}$	axial (torsional) rotation	(446e)
$\{\theta_{y1}^e\ \theta_{z1}^2\ \theta_{y2}^e\ \theta_{z2}^2\}$	bending rotations	(446f)

3D frame element: local system stiffness matrix



A 3D 12-freedom beam element defined in a local system

$$\boldsymbol{K}^e = \begin{bmatrix} a_1 & 0 & 0 & 0 & 0 & 0 & -a_1 & 0 & 0 & 0 & 0 & 0 \\ & b_1 & 0 & 0 & 0 & b_2 & 0 & -b_1 & 0 & 0 & 0 & b_2 \\ & & c_1 & 0 & -c_2 & 0 & 0 & 0 & -c_1 & 0 & -c_2 & 0 \\ & & & d_1 & 0 & 0 & 0 & 0 & 0 & -d_1 & 0 & 0 \\ & & & & c_3 & 0 & 0 & 0 & c_2 & 0 & c_4 & 0 \\ & & & & & b_3 & 0 & -b_2 & 0 & 0 & 0 & b_4 \\ & & & & & & a_1 & 0 & 0 & 0 & 0 & 0 \\ & & & & & & & b_1 & 0 & 0 & 0 & -b_2 \\ & & & & & & & & c_1 & 0 & c_2 & 0 \\ & & & & & & & & & d_1 & 0 & 0 \\ & & s & y & m. & & & & & & c_3 & 0 \\ & & & & & & & & & & & b_3 \end{bmatrix}$$

where

$$a_1 = EA/L^e \qquad d_1 = GK/L^e$$

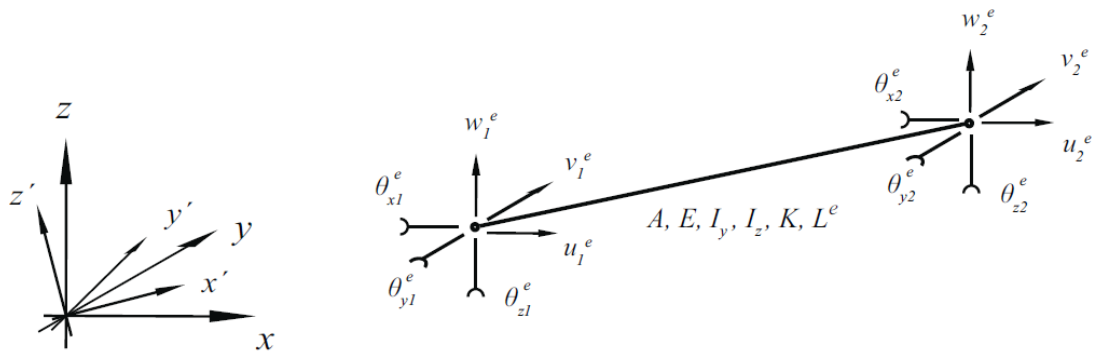
$$b_1 = 12EI_z/L^{e^3} \qquad b_2 = 6EI_z/L^{e^2} \qquad b_3 = 4EI_z/L^e \qquad b_4 = 2EI_z/L^e$$

$$c_1 = 12EI_y/L^{e^3} \qquad c_2 = 6EI_y/L^{e^2} \qquad c_3 = 4EI_y/L^e \qquad c_4 = 2EI_y/L^e$$

source: B. Torstenfelt; http://www.solid.iei.liu.se/Education/TMHL02/Book-Bars%20and%20Beams.pdf

1. EA contributes to axial displacement.
2. EI_y and EI_z contributes to bending vertical displacement and rotations.
3. GK contributes to axial (torsional) rotation. G is the torsional modulus and K is torsional proportional constant for the cross section.

3D frame: local to global element stiffness matrix



A 3D **global** 12-freedom beam element

source: B. Torstenfelt; http://www.solid.iei.liu.se/Education/TMHL02/Book-Bars%20and%20Beams.pdf

$$\mathbf{u}' = \Lambda \mathbf{u} \qquad \Lambda \ : 3 \times 3 \text{direction cosines of the two axis systems} \qquad (447a)$$

$$\mathbf{a'}^e = \begin{bmatrix} \Lambda & 0 & 0 & 0 \\ 0 & \Lambda & 0 & 0 \\ 0 & 0 & \Lambda & 0 \\ 0 & 0 & 0 & \Lambda \end{bmatrix} = \mathbf{T}\mathbf{a}^e \qquad \mathbf{T} \ : 12 \times 12 \text{ coordinate transformation matrix} \qquad (447b)$$

$$\mathbf{k}^e = \mathbf{T}^{\mathrm{T}} \mathbf{k'}^e \mathbf{T} \qquad \text{stiffness matrix in rotated system} \qquad (447c)$$

- Λ: 3×3 matrix containing direction cosines between X (global) and X' (local) coordinate systems.
- $\mathbf{T}$: 12×12 dof transformation matrix from global to local coordinate systems for the element.
- $\mathbf{k'}^e$: 12×12 element stiffness matrix in local X' coordinate system.
- $\mathbf{k}^e$: 12×12 element stiffness matrix in global X coordinate system.

Bar vs Beam element

	bar	beam
Differential equation	$\frac{\mathrm{d}}{\mathrm{d}x}\left(EA\frac{\mathrm{d}u}{\mathrm{d}x}\right)+q=0$	$\frac{\mathrm{d}^2}{\mathrm{d}x^2}\left(EI\frac{\mathrm{d}^2y}{\mathrm{d}x^2}\right)-q=0$
M	2	4
continuity requirement	C^1	C^2
$L_m(.)$	$\frac{\mathrm{d}}{\mathrm{d}x}$	$\frac{\mathrm{d}^2}{\mathrm{d}x^2}$
$\mathbf{B}^e$	$\frac{\mathrm{d}\mathbf{N}^e}{\mathrm{d}x}$	$\frac{\mathrm{d}^2\mathbf{N}^e}{\mathrm{d}x^2}$
dof/node (n_{dpn})	1	2
dofs @ node	u	y, $\frac{\mathrm{d}y}{\mathrm{d}x}(\theta)$
dof/element (n_{f}^e)	2	4
order of $\mathbf{N}^e$	1	3
$\mathbf{D}$	EA	EI

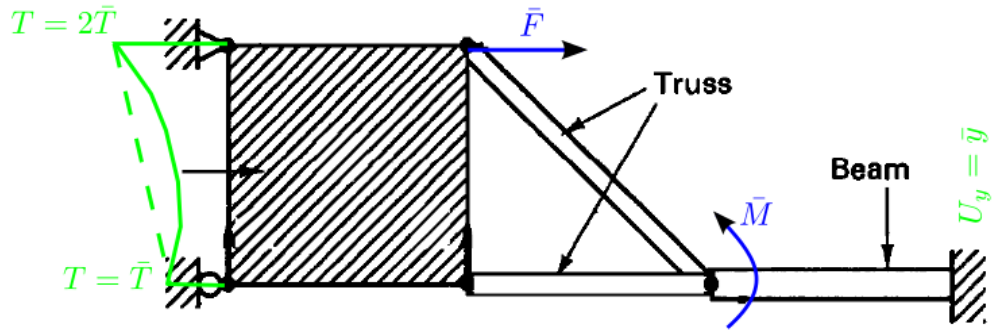
FEM stages

Finite element analysis is comprised of three main components:

- Preprocessor:
  1. Input: Geometry representation of the distinct domains, boundary conditions, loads, are specified plus other data not representable in geometric form.
  2. Meshing: Geometry representation is discretized to elements. Loads transferred to discrete model (*e.g.*, nodes, facets, *etc.*). Element types and their material is assigned based on the geometry model.
- Solver:
  1. Assembly: Discrete model equations (*e.g.*, stiffness matrix and force vector) are formed.
  2. System solve: Equations are solved (*e.g.*, $\mathbf{a} = \mathbf{K}^{-1}\mathbf{F}$).
  3. Element solutions: Transfer (nodal) solutions to elements.
- Postprocessor:
  1. Recovery process: Improving FEM raw solution using *e.g.*, superconvergence property (if needed).
  2. Error analysis (if needed).
  3. Output: in text and graphical forms.

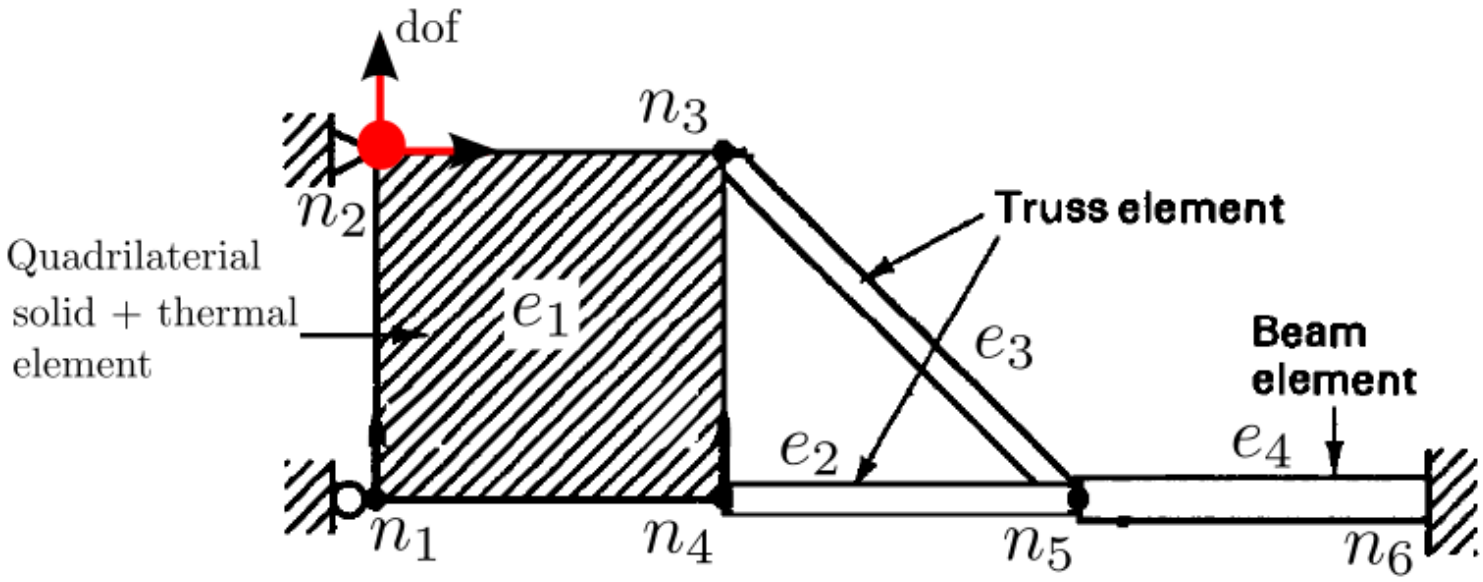
For adaptive, nonlinear, and/or time-dependent problems all of some parts of each stage are looped.

FEM Objects in preprocessor



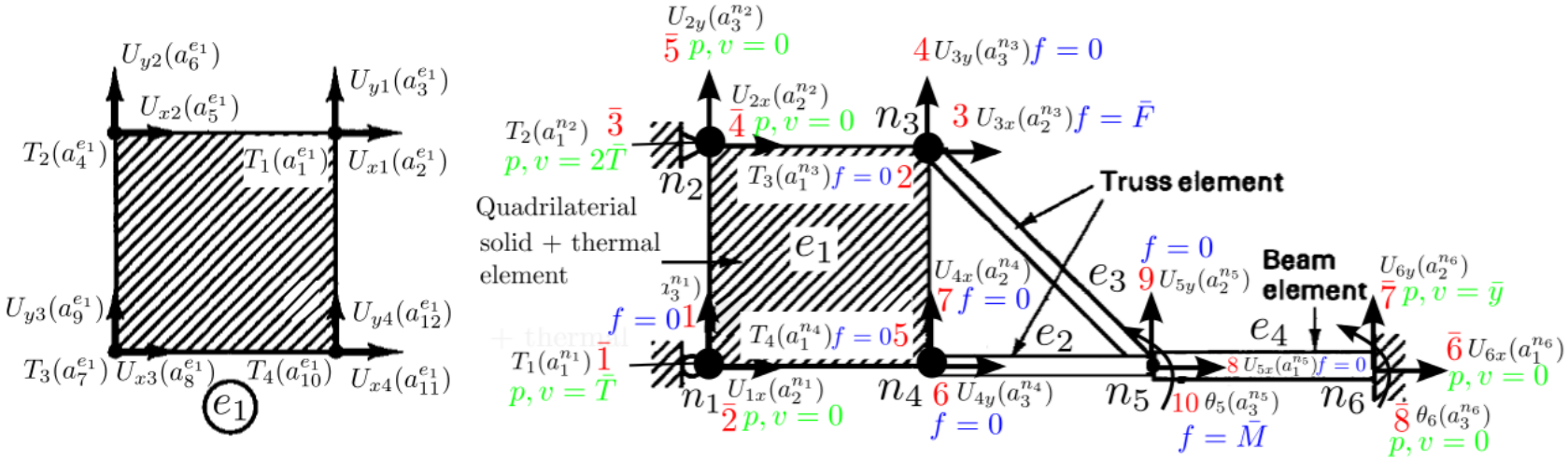
- Geometry Objects:
  - These objects define the exact geometry of the model, loads, boundary conditions and so forth.
  - Geometric objects are different than their possibly approximate representation in discretized version. For example, the discretization of a 2D domain with curved boundaries, using triangular elements introduces a geometric error.
  - In the figure above, a domain with simple geometry (a square solid region with 3 linear bar and elements) is shown:
  - As will be shown next, for one linear square solid element the actual representation of the prescribed temperature (in green) will be approximated by a linear function (dash-line). That is, geometry, loads, BCs, *etc.* should have an independent representation than their FEM counterparts.
  - In this course, we do not elaborate on the implementation of pre- and post-process stages and our main focus will be on the solver stage.

FEM Objects in solver



- For the remainder of the section, the focus is only on those used in solver stage.
- Some examples are shown in the figure: elements, nodes, dofs. These will be further discussed next.
- In the following, $\{a\}$ refers to a set, vector, tensor, or loosely speaking a collection of objects a (for example set of elements, nodes, *etc.*).

FEM Solver Objects: 1. Element: Data



- id: Clearly, id of e_1 is 1.
- neNodes (n_{n}^e): Number of element nodes (*e.g.*, for element 1 $n_{\mathrm{n}}^e = 4$).
- eNodes (LEM): Indices of element nodes in global system; *e.g.*, for e_1:

$$\mathrm{LEM}^{e_1} = \begin{bmatrix} 3 & 2 & 1 & 4 \end{bmatrix}$$

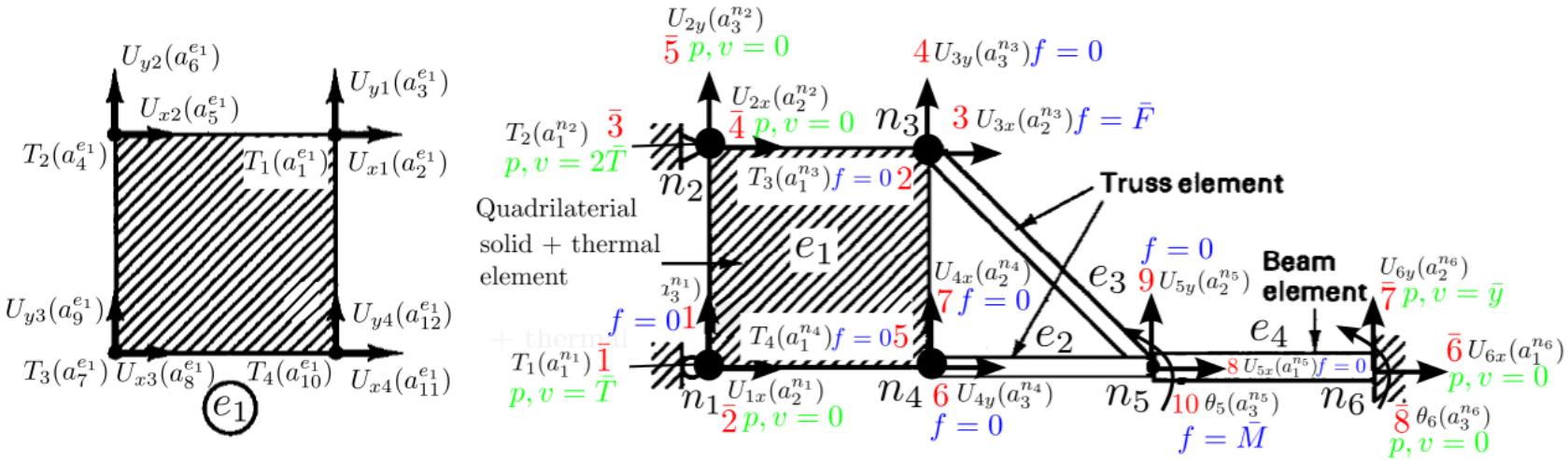
- number of element dof (nedof: n_{f}^e): can be different than n_{n}^e; for e_1 : $n_{\mathrm{f}}^e = 12$.
- edofs ($\mathbf{a}^e$): n_{f}^e vector of dofs; *e.g.*, for e_1:

$$a_1^e = \begin{bmatrix} a_1^{e_1} & a_2^{e_1} & \cdots & a_{n_{\mathrm{f}}^e}^{e_1} \end{bmatrix}$$
$$= \begin{bmatrix} T_1 & U_{x1} & U_{y1} & T_2 & U_{x2} & U_{y2} & T_3 & U_{x3} & U_{y3} & T_4 & U_{x4} & U_{y4} \end{bmatrix}$$

- dofMap ($\mathbf{M}_t^e$): map from element to global dofs; *e.g.*, for e_i we will observe:

$$\mathbf{M}^{e_1}{}_t = \begin{bmatrix} 2 & 3 & 4 & \bar{3} & \bar{4} & \bar{5} & \bar{1} & \bar{2} & 1 & 5 & 6 & 7 \end{bmatrix}$$

FEM Solver Objects: 1. Element: Data



- stiffness matrix (ke: $\mathbf{k}^e$): $n_{\mathrm{f}}^e \times n_{\mathrm{f}}^e$ local stiffness (conductivity, *etc.*) matrix.
- force vectors (fde, foe, fee: $\mathbf{f}_D^e, \mathbf{f}_o^e, \mathbf{f}_e^e$): A variety of element n_{f}^e load vectors such as essential BC ($\mathbf{f}_D^e$), sum of other forces ($\mathbf{f}_o^e = \mathbf{f}_r^e + \mathbf{f}_N^e + \cdots$, *etc.*), and element total $\mathbf{f}_e^e = \mathbf{f}_D^e + \mathbf{f}_o^e$.
- {physics}: Physics represented by the element; *e.g.*, for e_1 physics are solid and thermal.
- eType: one or more than one object that identify the element type. Examples are shape and order of element (linear triangle, *etc.*).

FEM Solver Objects: 1. Element: Function

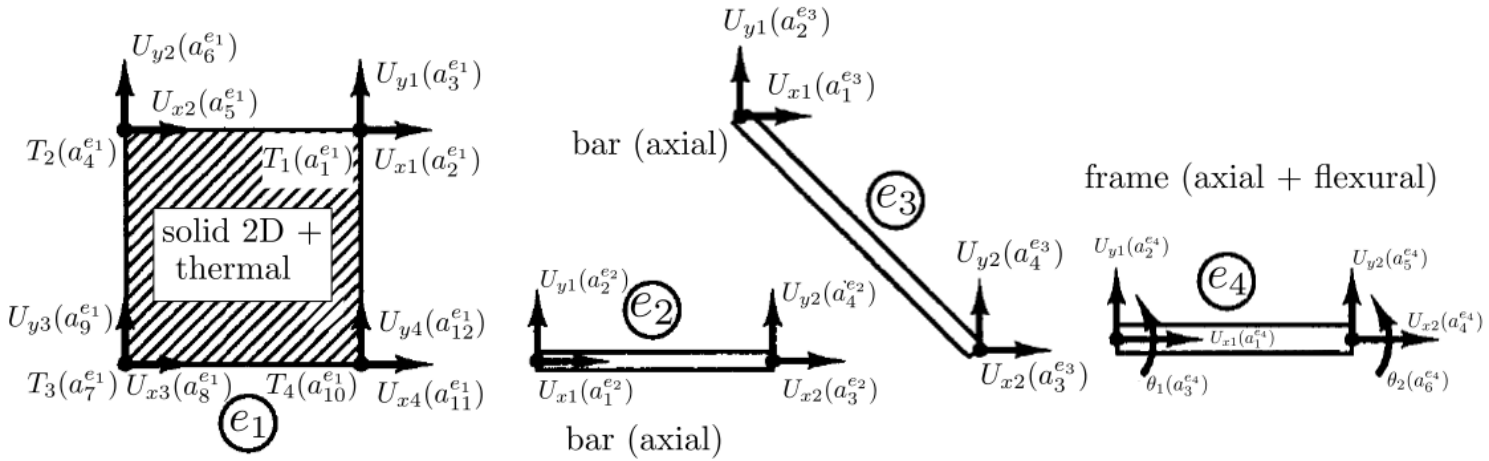
Some sample element functions are:

- Calculate $\mathbf{k}^e$ (virtual)
- Calculate $\mathbf{f}_o^e$: sum of all forces, but $\mathbf{f}_D^e$ (virtual)
- Compute_Output_Element: computes and outputs element; *e.g.*, axial forces for truss: (virtual)
- Calculate $\mathbf{f}_D^e = \mathbf{k}^e \mathbf{a}^e$
- Assemble $\mathbf{k}^e$ and $\mathbf{f}_e^e$ into global $\mathbf{K}$ and $\mathbf{F}$

Virtual

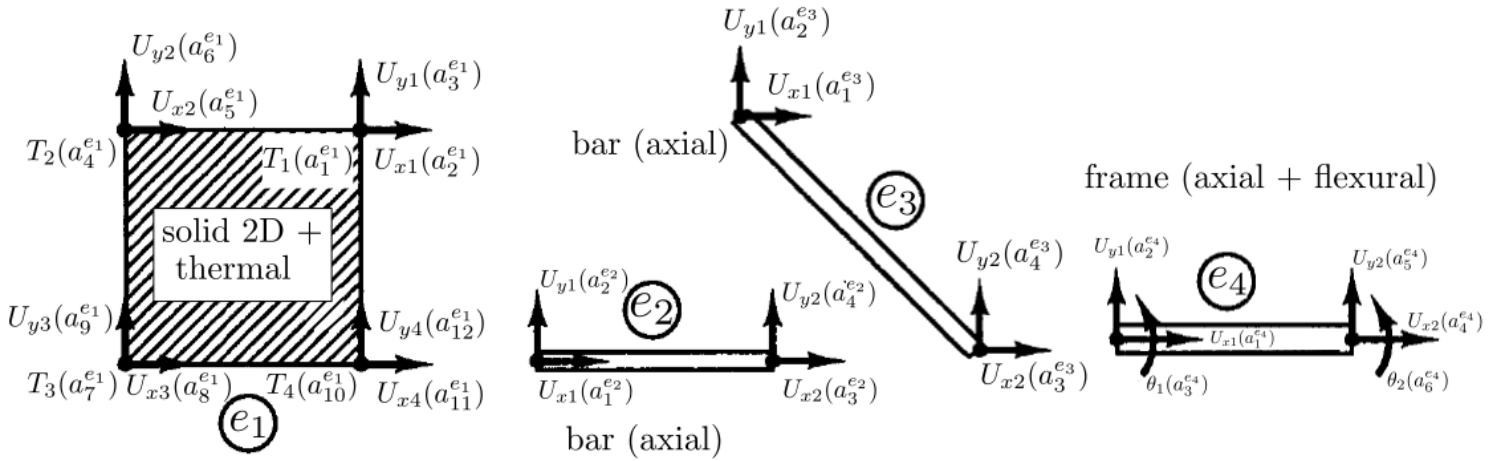
Virtual is an attribute of some functions in object oriented programming. Without going to details, virtual functions are black box functions where any different type of object (*e.g.*, solid physics element, thermal physics element, *etc.*) performs its independent routine to achieve the objective of the function (*e.g.*, compute $\mathbf{k}^e$ and $\mathbf{f}_o^e$ i the first two functions; compute area and surface for shapes, *etc.*).
This is opposed to the last two functions in this example (assembly, and $\mathbf{f}_D^e = \mathbf{k}^e \mathbf{a}^e$) where the same exact routine is performed for all types of objects.

FEM Solver Objects: 2. Physics



- Physics refers to physics that an element represents. For example for e_1 the element includes two physics of plane solid and thermal.
- Elements e_2 and e_3 have one physics of solid truss.
- Depending on interpretation, e_4 can be considered as one physics of solid frame or two physics of truss + beam.
- Physics includes data that specify how physics is integrated into the element.
- Some of physics data are similar to element data in terms of dofs, dof maps, *etc.*.
- As an example remember frame element where the integration of two physics of axial (bar) and flexural(beam) into the element resembled assembly of local element to global system.
- Due to complexity and flexibility of description of physics object (*cf.* refer to the choice of interpretation above), as well as the fact that our focus will be on single physics problems we skip details of physics class.
- One important member of physics is {field}: that is a physics includes a set of tensor (or parts of tensor) fields. Examples are (U_x, U_y) for truss, and (U_x, U_y) and θ for frame.

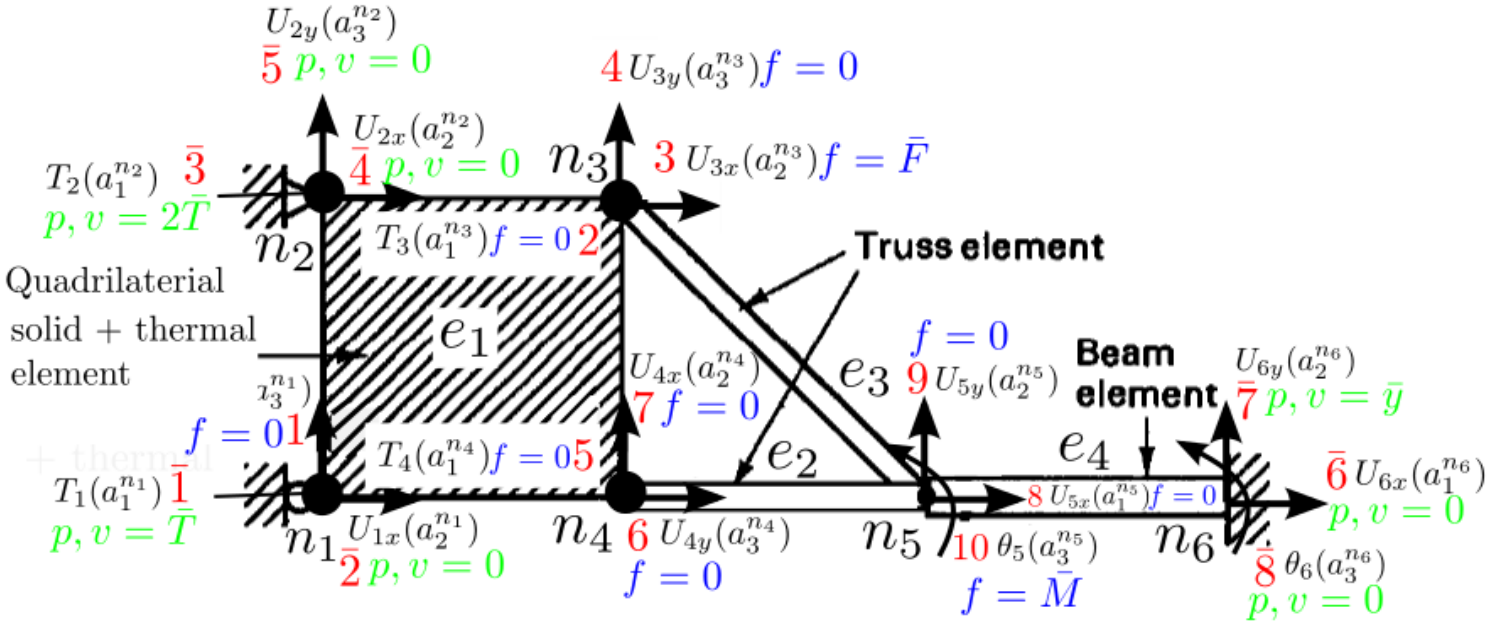
FEM Solver Objects: 3. Field



- Field is one complete or partial tensor field represented by the element
- For example, if we consider a 2D frame as one physics (as opposed to two physics of axial bar + flexural bending) this physics has the two fields: 1. Displacement: (U_x, U_y) and 2. rotation: (θ).
- In this example, (U_x, U_y) is partial since it misses U_z and $\theta := \theta_z$ is only one of the three rotations.
- Data: One important data of field is component. U_x is a component of the field, while the set (U_x, U_y) is the actual field.
- In summary:

Element data: {Physics}
Physics data: {Field}
Field data: {Component}

FEM Solver Objects: 4. Node: Data



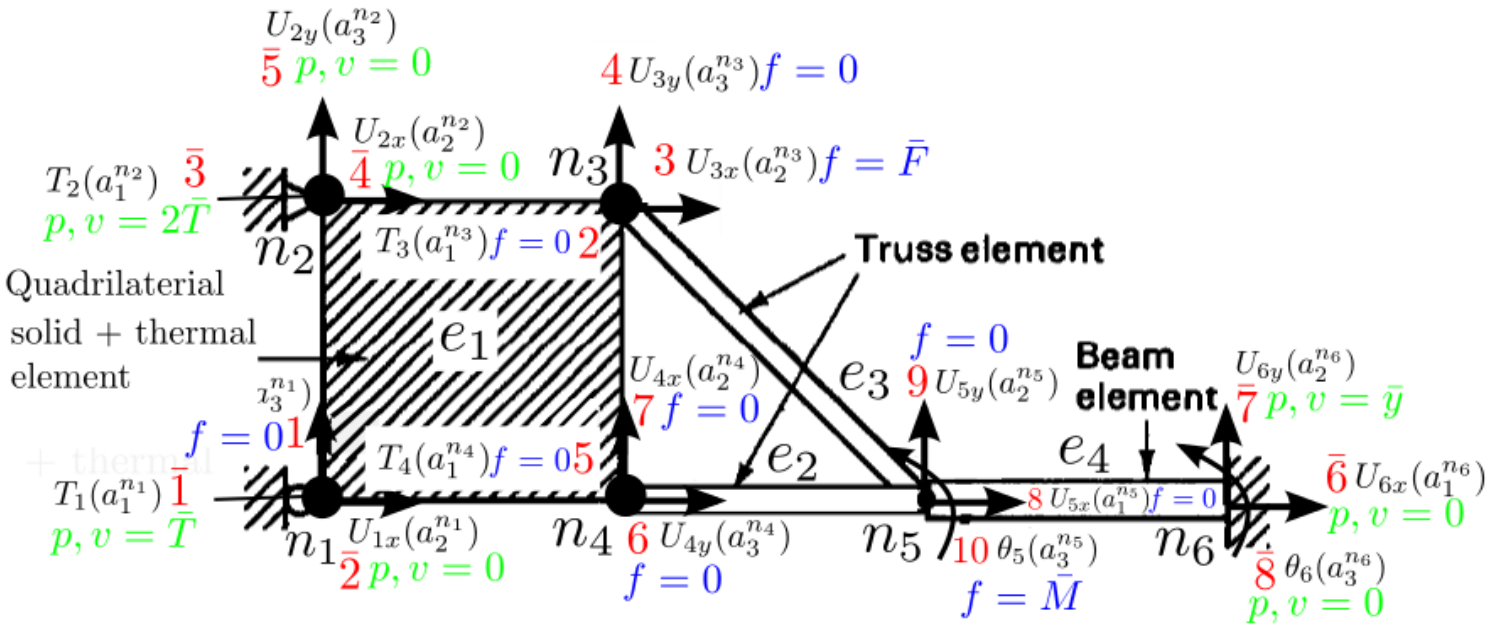
- id: Clearly, id of n_1 is 1.
- coordinate: *e.g.*, for n_1 the coordinate in the figure can be:

$$\mathrm{crd}_{XY}(n_1) = \begin{bmatrix} 0 & 0 \end{bmatrix}$$

  coordinate components are alway represented with respect to a coordinate system (another geometry object we will not further discuss herein).
- {ndof}: *i.e.*, a "set" of dofs. Dof is a class being described next. It includes data such as being free or prescribed, position in global free or prescribed dofs, value (*e.g.*, displacement), and force.
- nndof: Number of dof for the given node.

We will not discuss functions for the node object.

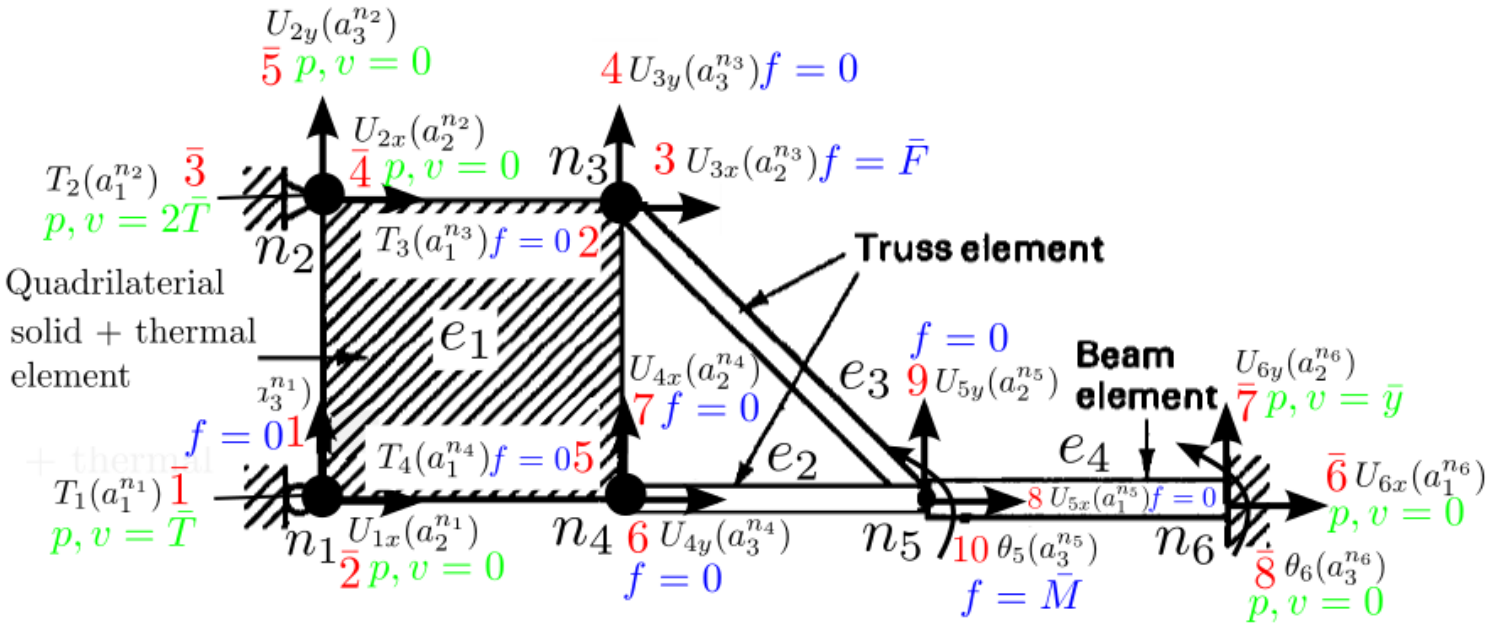
FEM Solver Objects: 5. Dof: Data



Dof refers to a degree of freedom for the system considered. It refers to both free and prescribed dofs. For continuous Galerkin FEM dofs are nodal dofs.Data for such dofs are:

- prescribed (p) A boolean that specified whether dof is free f (it's value is unknown) or prescribed p. DEFAULT: is $p = 0$, that is ONLY p dofs should be listed in input files.
- position (pos) - or alternatively called id: is the loc of dof in its corresponding group of f or p.
- value (v): is the value of dof. This value is unknown for f dofs and prescribed for p dofs. After solution the value will be known for all dofs.
- force (f): is the corresponding force for the dof. Force is known for f and unknown for p dofs (opposite to value). Again, all forces will be known after solution.
- Field (F): Referring to physical field; *e.g.*, Displacement $\mathbf{U}$, temperature T, *etc.*.
- index (i): The (tensorial) index for the field.

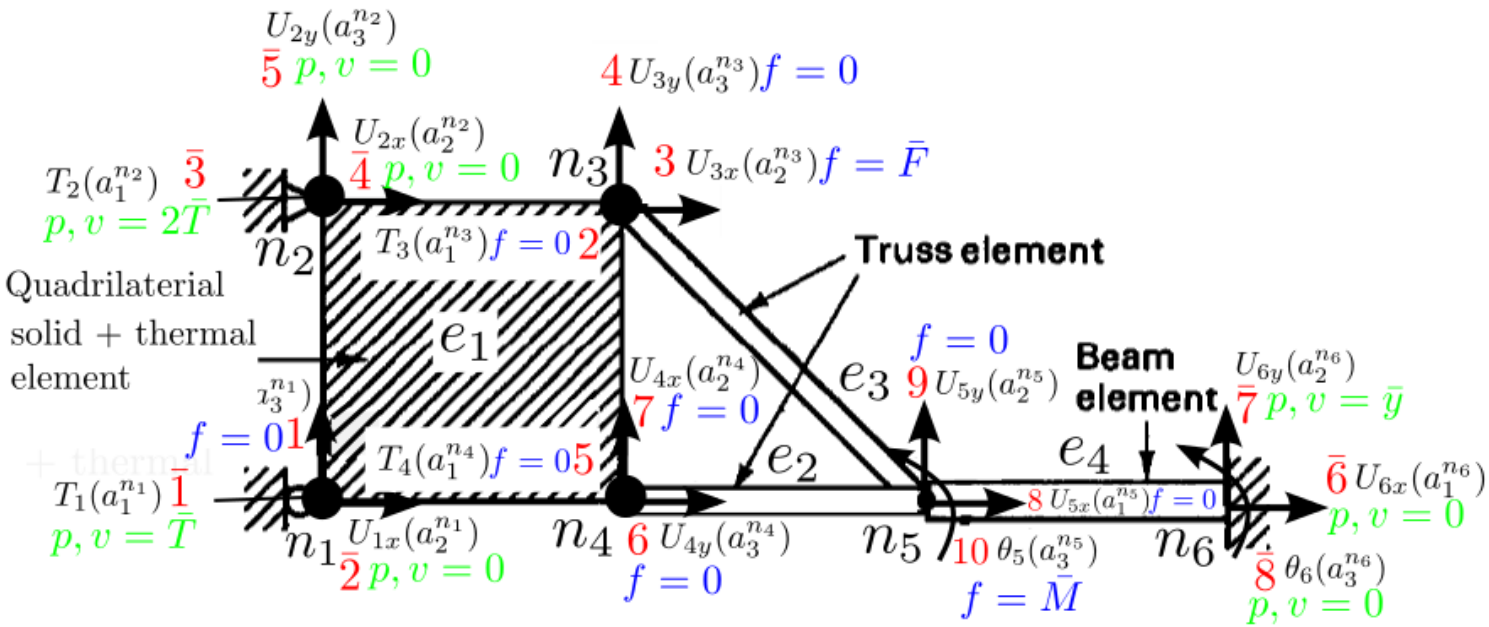
FEM Solver Objects: 5. Dof



Examples of dof for the structure shown are:

dof	p	pos	v	f	field	index
1 of n_1	true	$\bar{1}$	$\bar{T}$	unknown	T	-
3 of n_1	false	1	unknown	0	$\mathbf{U}$	2
3 of n_5	false	10	unknown	$\bar{M}$	θ	- (a vector in 3D)
2 of n_6	true	$\bar{7}$	$\bar{y}$	unknown	$\mathbf{U}$	2

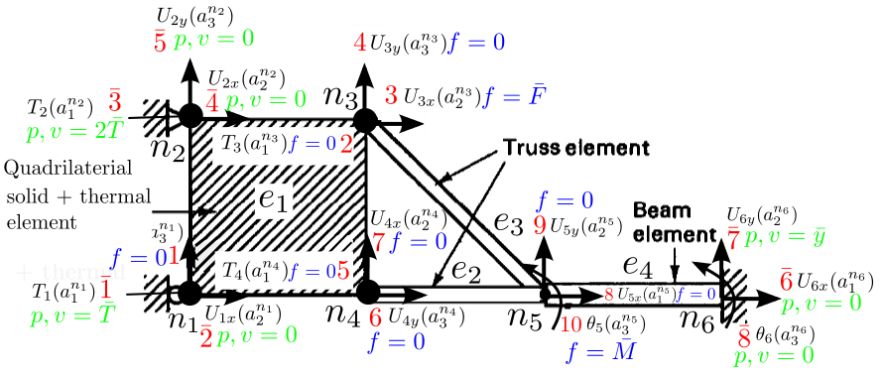
FEM Solver Objects: 6. Other solver related objects



Some other objects in solver stage are:

- Material Property: This object stores material properties, such as elastic modulus E, conductivity κ. Some other properties such as A, and I may be included in this object or in a separate object.
- Objects for I/O: Several objects or temporary storage members are used for input (*e.g.*, p and f dofs) and output (nodal and element solutions).

FEM Solver Objects: 7. FEM solver



There is not a clear object for FEM solver. However, we can think of FEM solver as an object that is responsible for 1) reading in discretization data; 2) storing all node and element data; 3) solution of global system. Basically FEM solver is the driver for FEM solution.

- dim, n_{dim} spatial dimension for the problem (1D, 2D, and 3D)
- number of nodes (nNodes, n_{n}) in the domain; *e.g.*, $n_{\mathrm{n}} = 6$.
- nodes {node}: vector of nodes in the domain; *e.g.*, $n_1, n_2, n_3, n_4, n_5, n_6$.
- number of elements (ne, n_{e}) in the the domain; *e.g.*, $n_{\mathrm{e}} = 4$.
- elements {element}: vector of elements in the domain; *e.g.*, e_1, e_2, e_3, e_4.
- free dofs (dofs: $\mathbf{a}$): vector of global free dofs.
- number of free dof (nf, n_{f}); *e.g.*, $n_{\mathrm{f}} = 10$.
- number of prescribed dof (np, n_{p}); *e.g.*, $n_{\mathrm{p}} = 8$.
- number of dof (ndof, $n_{\mathrm{dof}} = n_{\mathrm{f}} + n_{\mathrm{p}}$); *e.g.*, $n_{\mathrm{dof}} = 18$.
- stiffness matrix ($\mathbf{K} = \mathbf{K}_{ff}$); $n_{\mathrm{f}} \times n_{\mathrm{f}}$ matrix.
- force vector ($\mathbf{F} = \mathbf{F}_f$); $n_{\mathrm{f}}(\times 1)$ vector.
- prescribed force vector ($\mathbf{F}_p$) (Optional); $n_{\mathrm{p}}(\times 1)$ vector. May be used for more streamlined computation of prescribed dof forces.
- number of materials: nmats.
- material database {mats}: Parameters and Values for all material in the model.

Solution steps

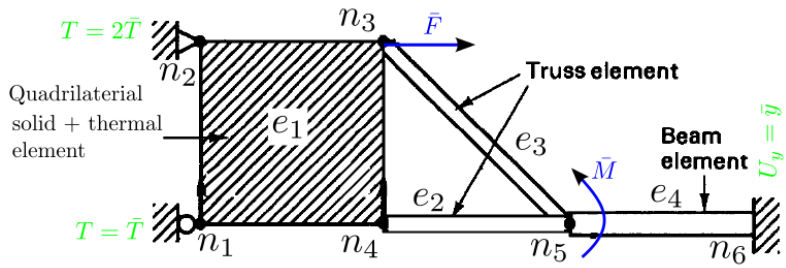
The steps for FEM solution are:

1. Set Element nodal dofs.
2. Set global dofs using element dofs.
3. Compute n_{f} from n_{dof} and n_{p} and resize and zero stiffness matrix and force vector.
4. Set global prescribed dofs.
5. Set global free dofs.
6. Set dof (free + prescribed) positions.
7. Set $\mathbf{F}(\mathbf{F}_f)$.
8. Set element dof maps $\mathbf{M}_t^e$.
9. Set element (prescribed) dofs.
10. Compute element stiffness matrix and force vectors.
11. Assemble element stiffnesses and forces to global system.
12. Solve for (free) dofs $\mathbf{a}$ from $\mathbf{K}\mathbf{a} = \mathbf{F}$.
13. Assign $\mathbf{a}$ to nodes and elements.
14. Compute prescribed dof forces: $\mathbf{F}_p$ (if needed).
15. Compute (if needed) output nodes and elements.

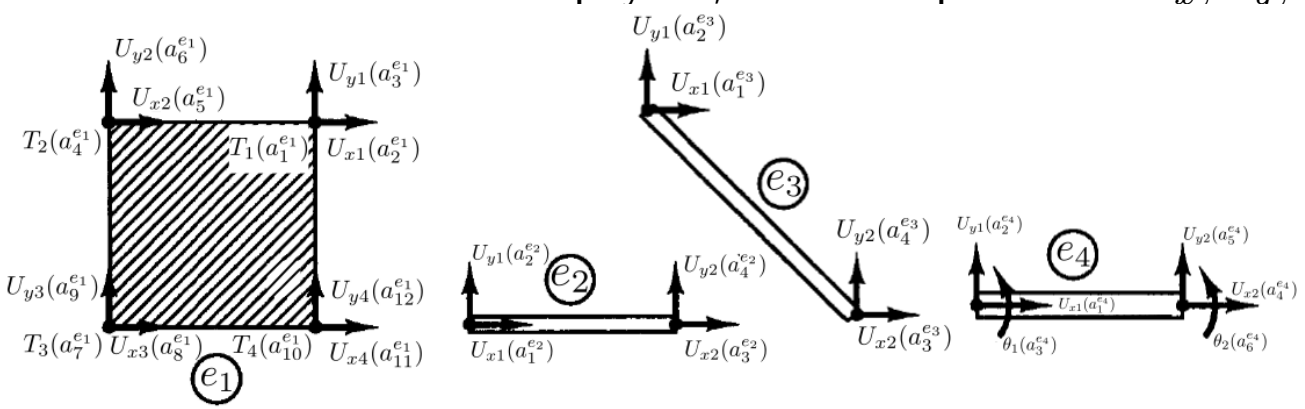
Solution steps

- We are assuming that 1) domain is discretized (meshing) 2) all element and node raw data is read 3) p dofs and f dofs with nonzero force are specified 4) material properties and element types are specified.
- Several steps can be done at once as they involve the same loop over nodes or elements, but for clarify they are separated in the list.
- Some steps (*e.g.*, computation of $\mathbf{F}_p$ and element values) are optional.
- Some steps can be done in parallel or the order can be changed (*e.g.*, computation of $\mathbf{F}_p$ and element values).

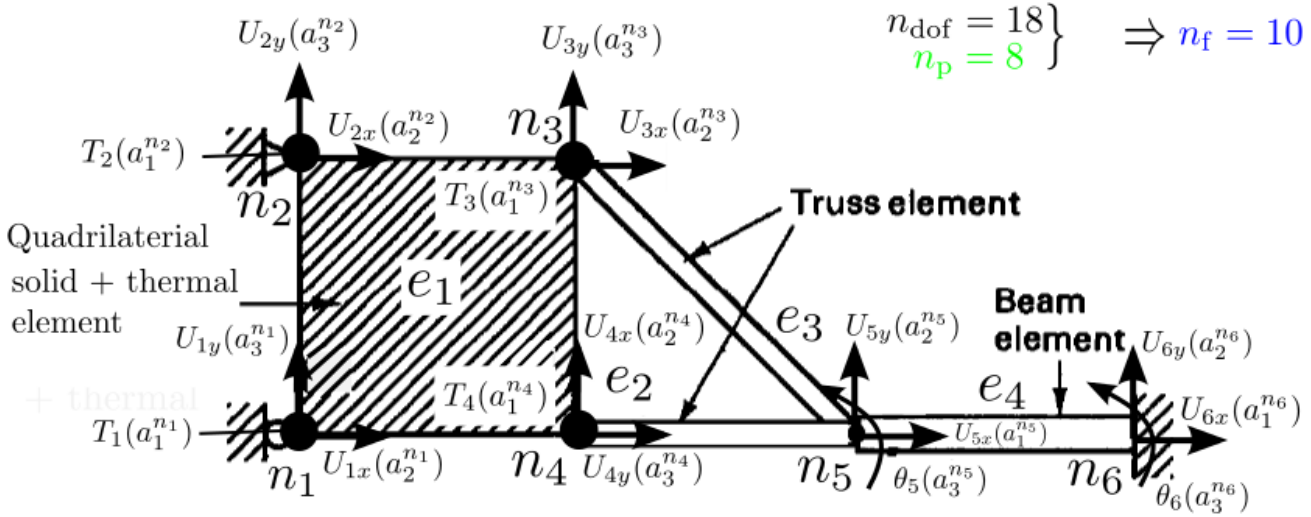
Step 1: Set element nodal dofs



- We loop over elements, physics, and fields to update each node of elements dof:
  for e = 1:ne
    for p = 1:elements(e).np np: number of physics
      for f = 1:elements(e).physicsp.nfields
        update elements(e) nodal dof for nodes having field f
      end
    end
  end
- e_1 for two physics of solid and thermal each node has 3 dofs: T, U_x, U_y.
- e_2 and e_3 have the two dof of U_x, U_y at their nodes.
- e_4, whether considered as one or two physics, has 3 dof per node: U_x, U_u, θ.



Step 2: Set global dofs using element dofs.

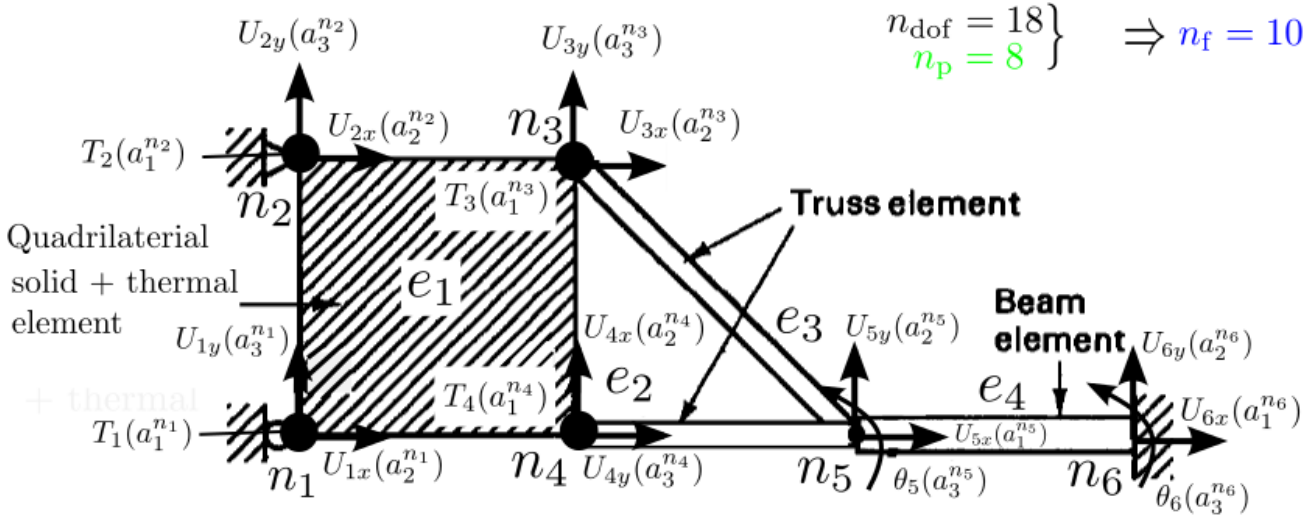


- Once we have each elements dofs at their nodes, we can form the entire set of dofs for each node in the domain.
- For each node, we loop over elements attached to the node and complete the node's set of dof:

```
for n=1:nNodes
   for e:1:number element attached to node n
      update node n dofs
   end
end
```

- However, we often do not have such information; We know which nodes are attached to elements, but not other way around.

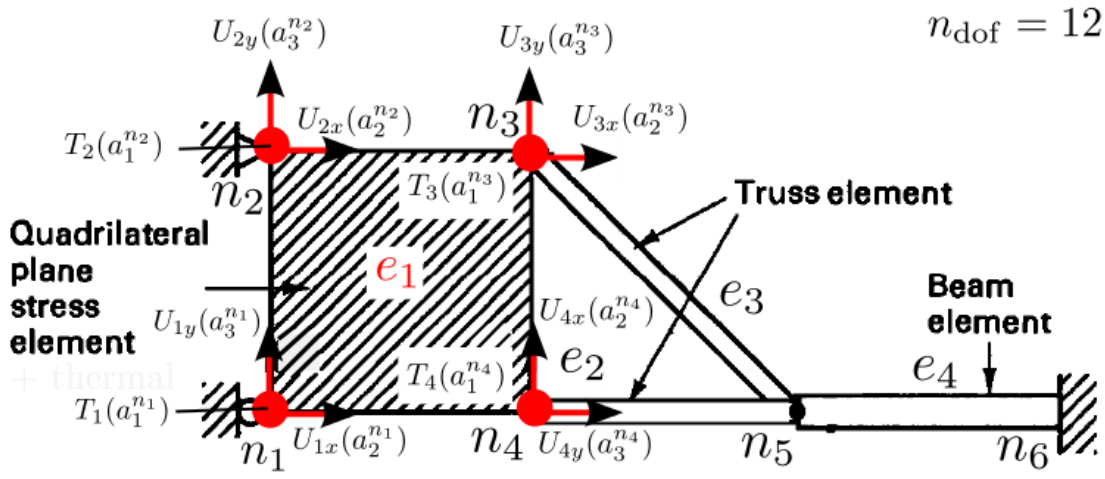
Step 2: Set global dofs using element dofs.



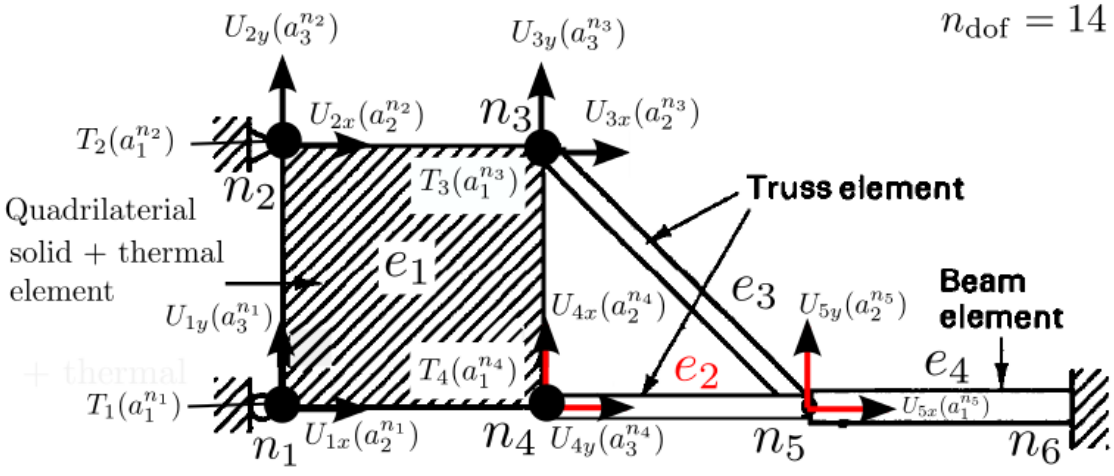
- We need to change the order of double nested for loop above:
  ndof = 0 number of total dof
  for e=1:ne
    for en=1:neNodes(e) number of nodes of element e
      g_nn = element(e) (LEM(en)) global node number for element node en
      nnew = update node g_nn's dof from element e node en ndofs
        nnew is number of newly added dofs
      ndof = ndof + nnew keep the ndof updated in this process.
    end
  end
- Each element may add some, all or none to the set of dofs of nodes attached to it, as shown in next slides.
- During this process we also compute ndof (n_{dof})

Step 2: Set global dofs using element dofs.

- e_1: 3 dofs are added to nodes n_1, n_2, n_3, n_4.

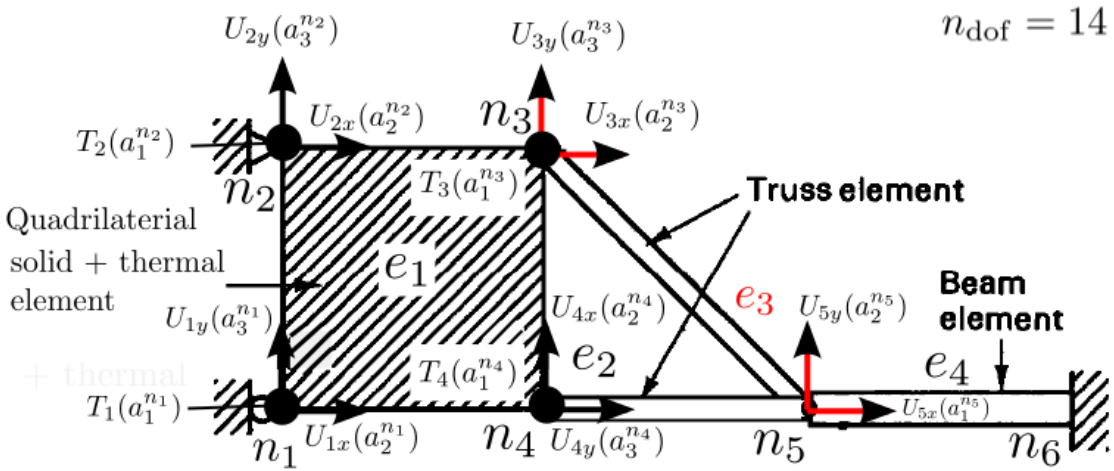


- e_2: U_x, U_y dofs are already added at n_4.
- We keep track of which global dofs correspond to elements nodal dof. For example, in e_2, node 1 dofs U_x, U_y correspond to dof 2 and 3 of global n_4 (first dof is T).
- Only two dofs U_x, U_y are added at n_5 $\Rightarrow$ $n_{\text{dof}} = 12 + 2 = 14$.

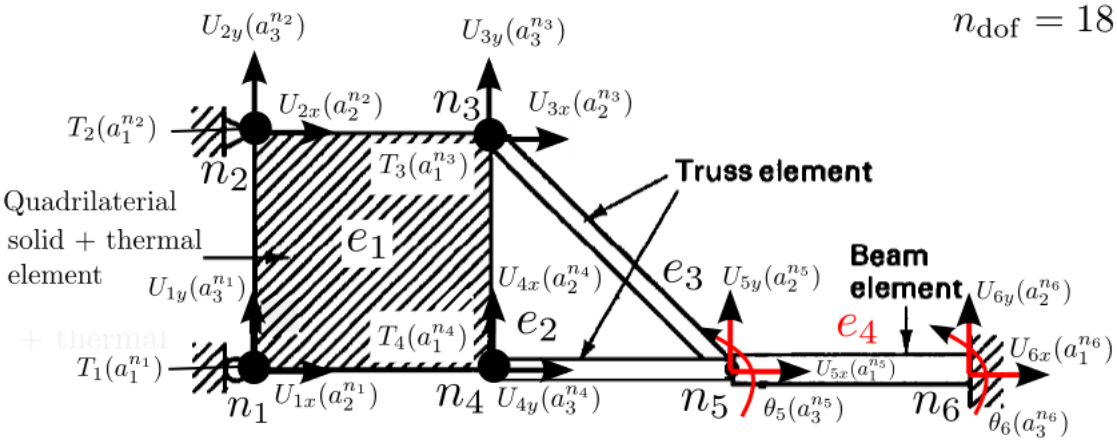


Step 2: Set global dofs using element dofs.

- e_3: no dof is added as n_3 and n_5 already have U_x, U_y.



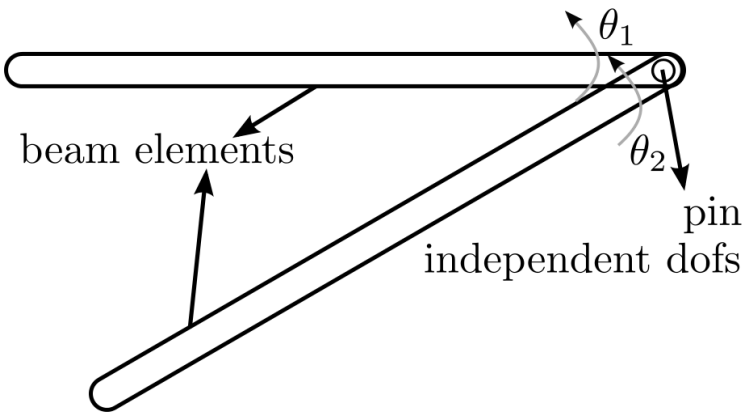
- e_4: θ is added to n_5 and U_x, U_y, θ added to n_6 $\Rightarrow$ $n_{\text{dof}} = 14 + 4 = 18$.



Steps 1 & 2: Difficulties

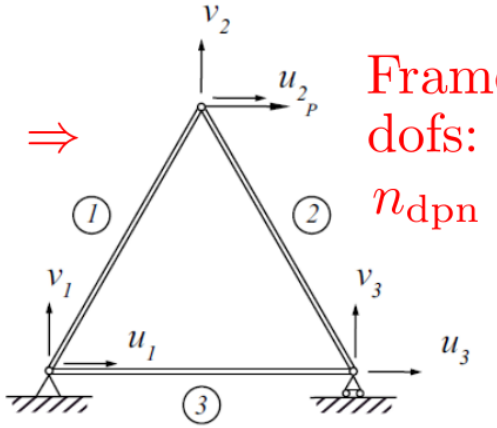
The two first steps are complicated for the following reasons:

1. Step 1: set-up of element dofs can be a very tedious process in general case since it needs a nested assembly of dofs at element level from components $\rightarrow$ fields $\rightarrow$ physics $\rightarrow$ elements.
2. Data structures for labeling physics, fields, components, and their bookkeeping can be complicated.
3. We need to keep track of element nodal dof map to their corresponding global dof map (example being in e_2 left nodes 1st and 2nd dof correspond to global n_4 2nd and 3rd dof (U_x, U_y)).
4. Number of dof per global node and its composition can change from node to node (*e.g.*, T, U_x, U_y vs U_x, U_y, θ in this example).
5. There are many different connections that make general handling of this process difficult; *e.g.*, consider the hinge connection for two beams. Each beam has a rotational dof, but that is not shared between the two.

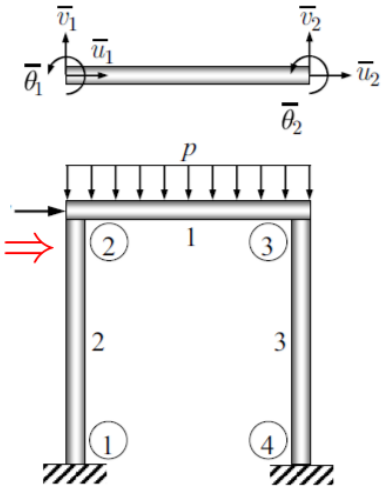


Steps 1 & 2: Simplified limited case

Truss
dofs: u, v $\Rightarrow$
$n_{\text{dpn}} = 2$



Frame
dofs: u, v, θ $\Rightarrow$
$n_{\text{dpn}} = 3$

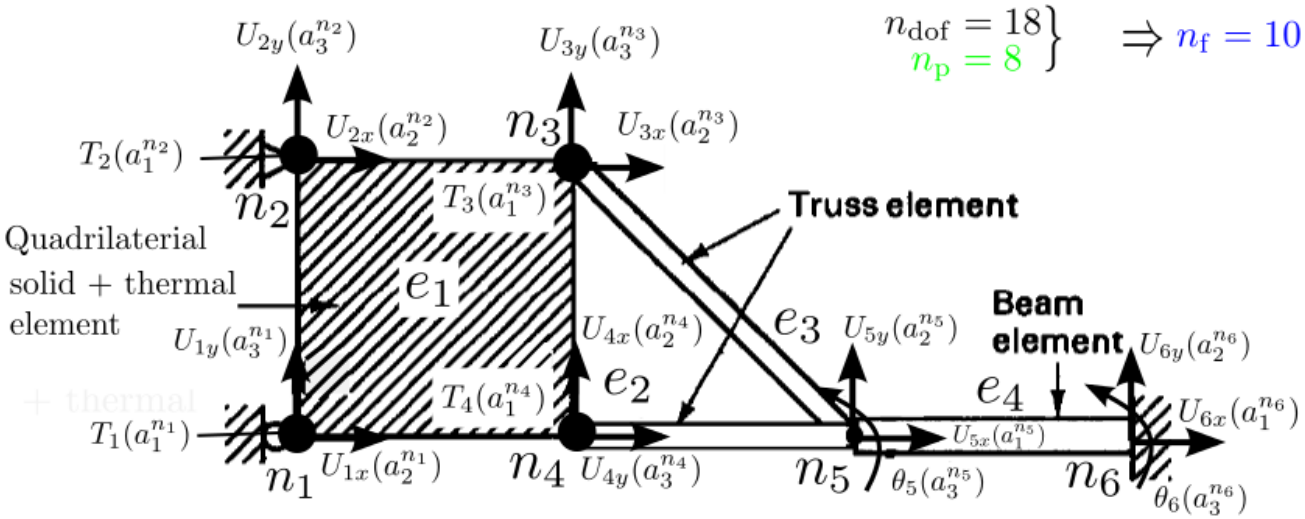


- FEM implementation become considerably simpler for problems where all elements are of the same type (regardless of number of physics per element).
- In this case, we define:

$$\boxed{n_{\text{dpn}} := \text{Number of dof. per node denoted by ndofpn}} \tag{448}$$

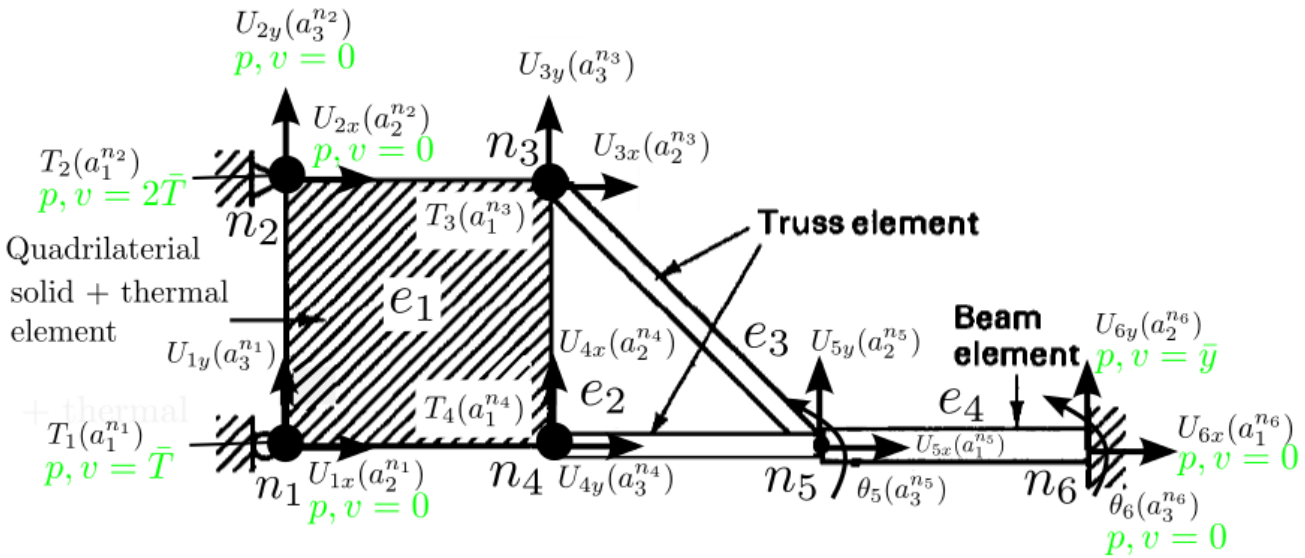
- There would be identity map between element nodal dof and global nodal dofs. That is, there is the same set of dofs used for both.
- Figure above shows two of such examples:

Step 3: Set global number of dofs, stiffness, and force.



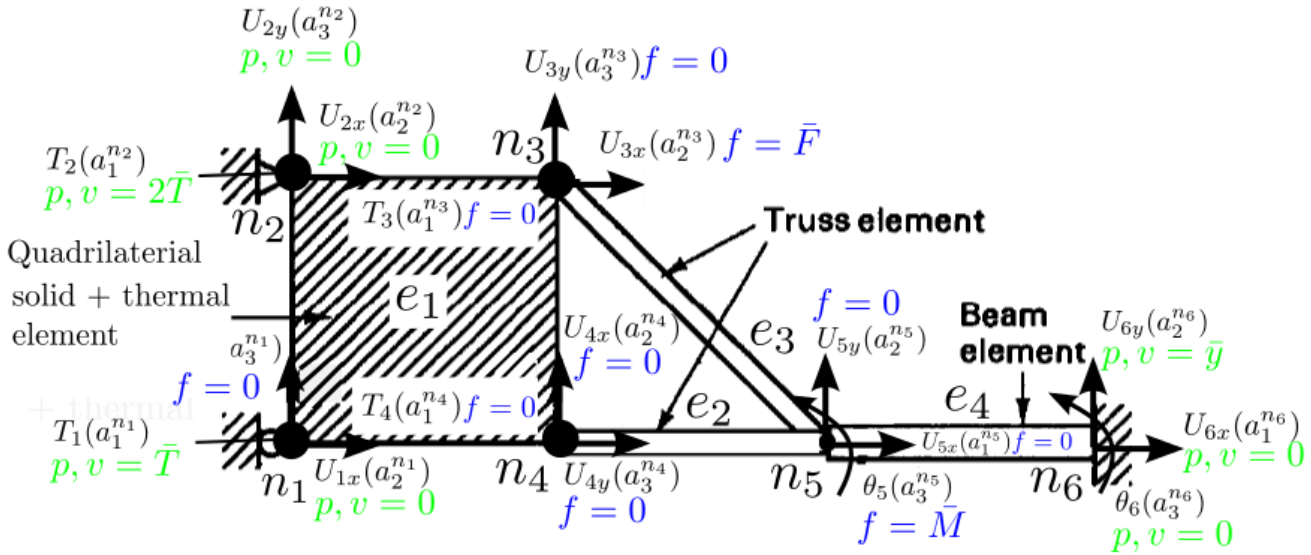
- n_{p} (number of prescribed dofs) is an input to FEM analysis.
- n_{dof} is computed in step 2.
- $n_{\text{f}} = n_{\text{dof}} - n_{\text{p}}$ is the number of free dofs.
- We size and zero $\mathbf{K}(\mathbf{K}_{ff}) : n_{\text{f}} \times n_{\text{f}}$ and $\mathbf{F}(\mathbf{F}_f) : n_{\text{f}}$ (member of FEM Solver).
- While not necessary, for simplicity we also size and zero $\mathbf{F}_p : n_{\text{p}}$ (member of FEM Solver).

Step 4: Set global prescribed nodal dof



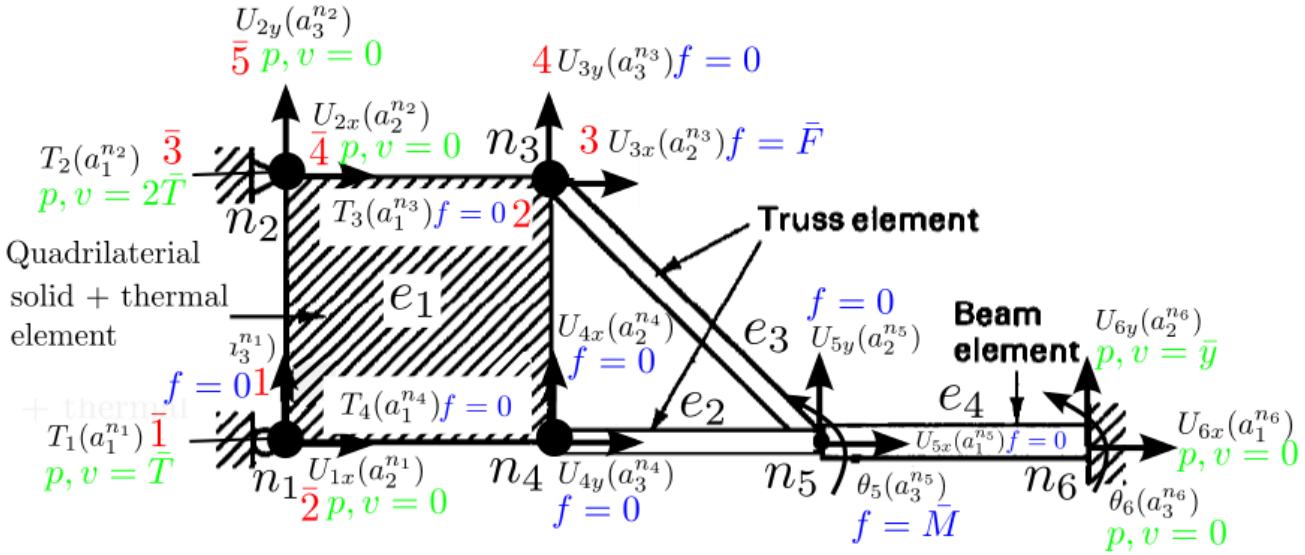
- In next few steps we set all members of nodal dofs.
- First, from the list of prescibed dofs with their corresponding values we set:
  1. p boolean flag is set to true.
  2. Value v for such dofs.
- Figure above shows all dofs after p ones are set (with their value).
- 8 dofs (shown in green) are prescribed dof.

Step 5: Set global free nodal dof



- All the nodal dofs that are not specified to be prescribed are by default free.
- We only input free dofs that have nonzero force.
- As a result, default value for force f of dofs is zero.
- We loop over input list of nonzero free dofs and set their values in nodal dof.
- In the figure only two dofs have nonzero free dofs and only those are input to solver.

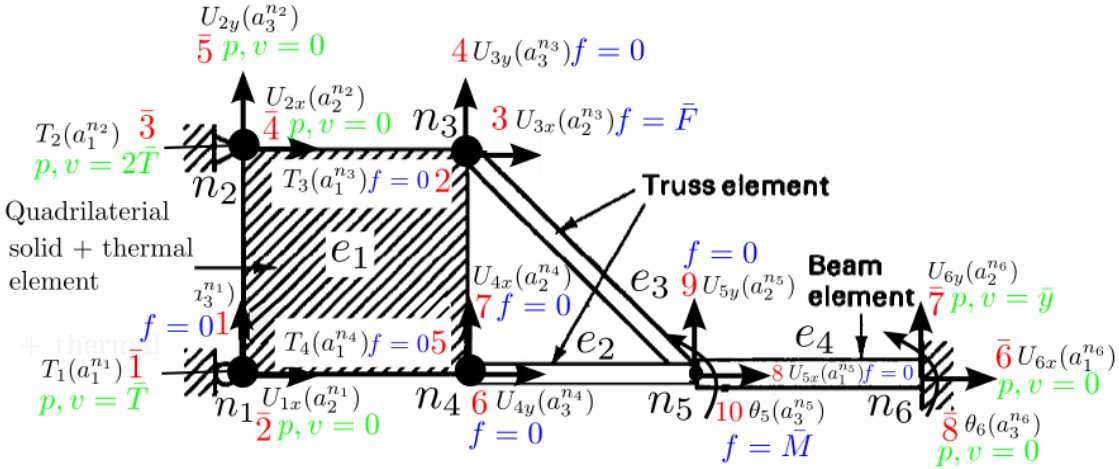
Step 6: dof positions; Step 7: Set $\mathbf{F}(\mathbf{F}_f)$



- Step 6, dof positions: We loop over nodes and dofs of nodes, increment their positions in global dofs.
- Dofs with $p =$ true have a separate position than free dofs (the former can take negative values denoted by $\bar{1}, \bar{2}, \ldots, \bar{n_{\mathrm{p}}}$.
- Step 7, set $\mathbf{F}$: While we do the double loop on nodal dofs, we can form the force vector as well. This can also be done subsequent to Step 6 altogether.
- Figure above shown the numbering after looping over 3 nodes.
- The current load vector is:

$$\mathbf{F} = \begin{bmatrix} 0 & 0 & 0 & \bar{F} & 0 & 0 & 0 & 0 & 0 & 0 \end{bmatrix}$$

Step 6: dof positions; Step 7: Set $\mathbf{F}(\mathbf{F}_f)$

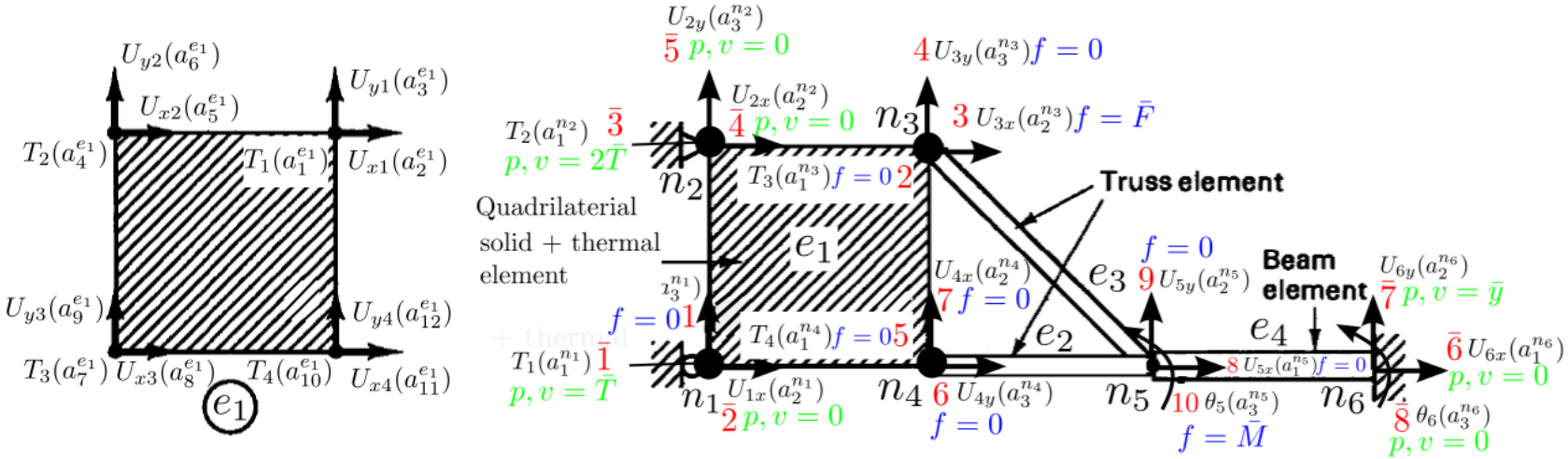


- After looping over all dofs the final values in load vector are:

$$\mathbf{F} = \begin{bmatrix} 0 & 0 & 0 & \bar{F} & 0 & 0 & 0 & 0 & 0 & \bar{M} \end{bmatrix}$$

```
posf = 0, posp = 0
for n = 1:nNodes
   for dofi = 1: node(n).nndof num dof for node (n)
      if node(n).ndof(dofi).p == true prescribed dof
         posp = posp - 1;
         node(n).ndof(dofi).pos = posp;
      else free dof
         posf = posf + 1;
         node(n).ndof(dofi).pos = posf;
         F(posf) = node(n).ndof(dofi).f
      end
   end

end
```

Step 8: Element dof maps $\mathbf{M}_t^e$



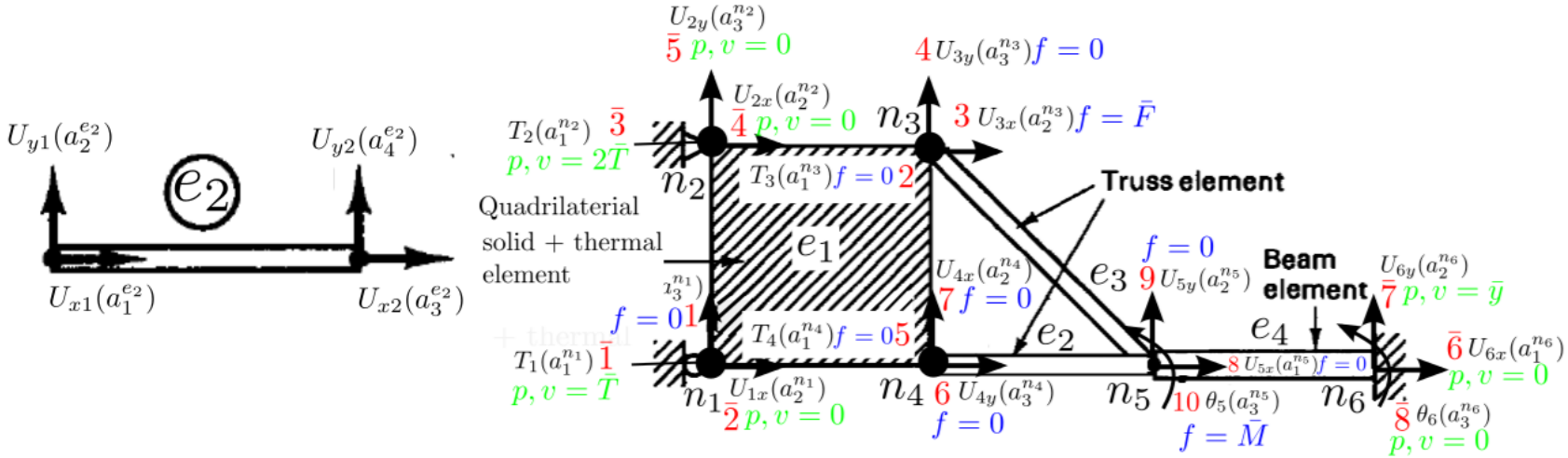
- As mentioned, $\mathbf{M}_t^e$ is a vector of size n_{dof}^e that maps elements dofs to global positions.
- For element 1, dofs are ordered as (loop over nodes, then loop over dofs for the node):

$$
\begin{aligned}
a_1^e &= \begin{bmatrix} a_1^{e1} & a_2^{e1} & \cdots & a_{12}^{e1} \end{bmatrix} \\
&= \left[\begin{array}{ccc|ccc|ccc|ccc} T_1 & U_{x1} & U_{y1} & T_2 & U_{x2} & U_{y2} & T_3 & U_{x3} & U_{y3} & T_4 & U_{x4} & U_{y4} \end{array}\right]
\end{aligned}
$$

- We need to map these dofs to global dofs and have their position in $\mathbf{M}_t^e$ vector. For example, 1st dof of node 1 ($a_1^{e1} = T_1$) is mapped to first dof of n_3 which has position 2.
- 2nd dof of node 3 ($a_8^{e1} = U_{y2}$) is mapped to 2nd dof of n_1 which has position $\bar{2}(-2)$.
- The map for element e_1 is:

$$
{\mathbf{M}^{e_1}}_t = \begin{bmatrix} 2 & 3 & 4 & \bar{3} & \bar{4} & \bar{5} & \bar{1} & \bar{2} & 1 & 5 & 6 & 7 \end{bmatrix}
$$

Step 8: Element dof maps $\mathbf{M}_t^e$



- For element 1, dofs are ordered as:

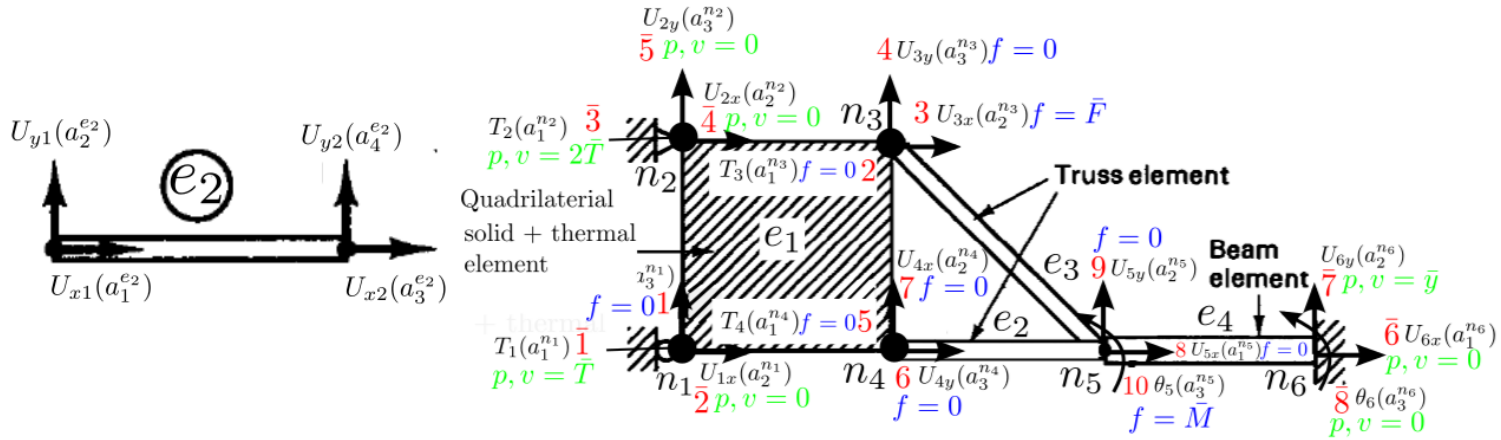
$$a_2^e = \begin{bmatrix} a_1^{e2} & a_2^{e2} & a_3^{e2} & a_4^{e2} \end{bmatrix}$$
$$= \left[\begin{array}{cc|cc} U_{x1} & U_{y1} & U_{x2} & U_{y2} \end{array}\right]$$

- For example, 1st dof of node 1 ($a_1^{e2} = U_{x1}$) is mapped to 2nd dof of n_4 which has position 6.
- Note that 1st dof of n_4 is temperature T.
- That is ith dof of node in element is not necessarily mapped to ith dof of corresponding global node.
- The map for element e_2 is:

$$\mathbf{M}^{e_2}{}_t = \begin{bmatrix} 6 & 7 & 8 & 9 \end{bmatrix}$$

- We also note that node 2 of element (global n_5 has 3 dofs from which the rotation θ_5 is not used in e_2 and does not contribute to $\mathbf{M}^{e_2}{}_t$

Step 8: Element dof maps $\mathbf{M}_t^e$



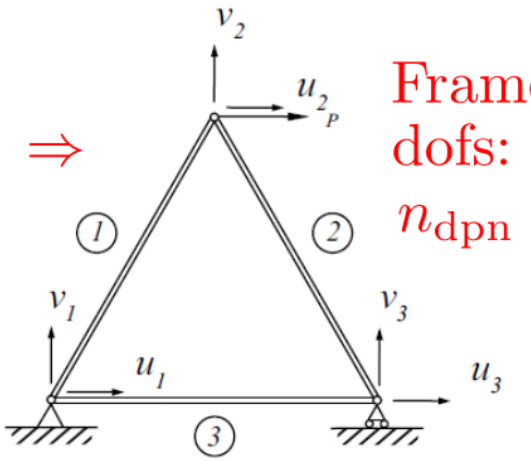
In general case, to form $\mathbf{M}_t^e$ we do the following:

ecdof = 1 dof counter for element
for en = 1: neNodes number of element nodes
  gn = LEM(en) global node number for element node en, *e.g.*, gn = 5 for en = 2 in e_2
  for endof = 1:nndof(en) number of dofs of element node en *e.g.*, nndof(en) = 2 for en = 2 in e_2
    gndof = e2globalDofMap(en, endof)
      gndof, corresponding dof in gn, *e.g.*, gndof = 2 for endof = 1, en = 2 in e_2
    dofMap(ecdof) = node(gn).dof(gndof).pos
      pos of dof #gndof of global node gn, *e.g.*, 6 for endof = 1, en = 2 in e_2
    ecdof = ecdof + 1 increment counter
  end
end

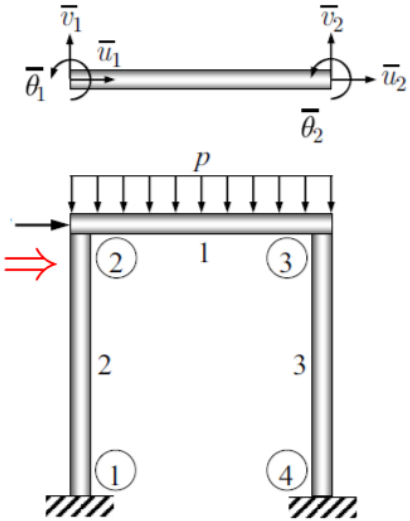
- The bookkeeping for this process (*e.g.*, formation of e2globalDofMap) can be tedious.
- The complexity of the formation of dofMap ($\mathbf{M}_t^e$) is: ith dof of node in element is not necessarily mapped to ith dof of its corresponding global node.

Step 8: Element dof maps $\mathbf{M}_t^e$: Simplified limited case

Truss
dofs: u, v $\Rightarrow$
$n_{\text{dpn}} = 2$



Frame
dofs: u, v, θ $\Rightarrow$
$n_{\text{dpn}} = 3$



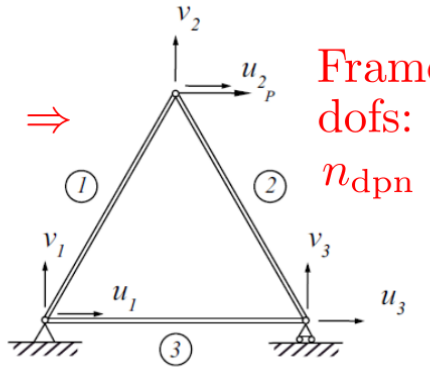
- in Steps 1 and 2 we mentioned that FEM implementation becomes considerably simpler if we assume all nodes share exactly the same set of dofs.
- Examples are bars, beams, trusses, and frames.
- In (448) we defined n_{dpn} (ndofpn) as,

$$n_{\text{dpn}} := \text{Number of dof. per node}$$

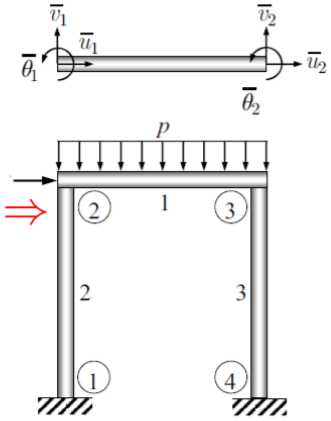
- In this limited scenario ith dof of node in element is mapped to ith dof of its corresponding global node.

Step 8: Element dof maps $\mathbf{M}_t^e$: Simplified limited case

Truss
dofs: u, v $\Rightarrow$
$n_{\text{dpn}} = 2$



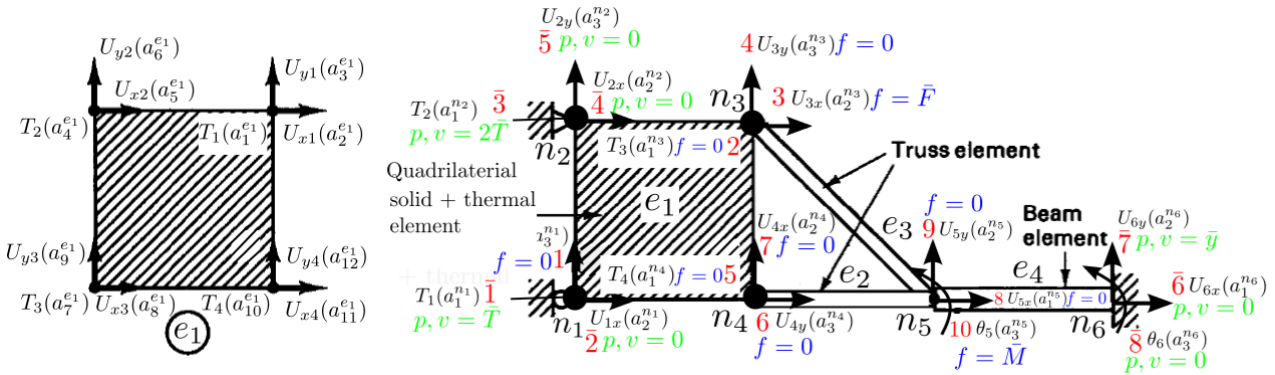
Frame
dofs: u, v, θ $\Rightarrow$
$n_{\text{dpn}} = 3$



A simplified pseudo code looks like: ecdof = 1 dof counter for element
for en = 1: neNodes number of element nodes
  gn = LEM(en) global node number for element node en
  for endof = 1: ndofpn This number is fixed now, *e.g.*, 2 for 2D trusses
    dofMap(ecdof) = node(gn).dof(endof).pos
      gndof = endof, we bypass some steps here
    ecdof = ecdof + 1 increment counter
  end
end

- Two major simplifications are:
  - number of dof per node is fixed for all nodes.
  - gndof = endof: That is dof number endof of node en of element is the same dof in its corresponding global node. For example U_1 in trusses is always 1st dof for element nodes and 1st for global nodes as well.

Step 9: Set element dofs $\mathbf{a}^e$

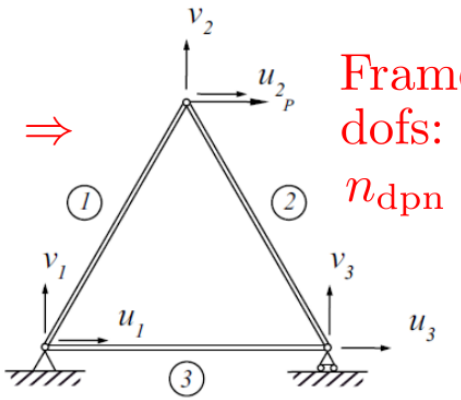


- To calculate Essential BC force for element $\mathbf{f}_D^e = \mathbf{k}^e\mathbf{a}^e$, we need to set $\mathbf{a}^e$.
- This step as many other cases can be combined with other steps in one loop over elements, but for clarity it is described separately.
- We can merge this step with Step 8: Element dof maps $\mathbf{M}_t^e$ by adding lines in red:

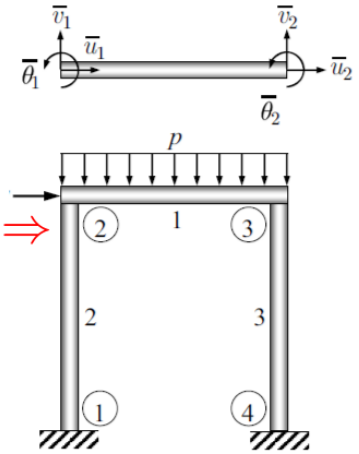
edofs = zeros(nedof) element dofs (edof) resized to number of element dofs and zerod
ecdof = 1 dof counter for element
for en = 1: neNodes # element nodes
  gn = LEM(en) global node number for element node en, *e.g.*, gn = 5 for en = 2 in e_2
  for endof = 1:nndof(en) number of dofs of element node en *e.g.*, nndof(en) = 2 for en = 2 in e_2
    gndof = e2globalDofMap(en, endof)
      gndof, corresponding dof in gn, *e.g.*, gndof = 2 for endof = 1, en = 2 in e_2
    if (node(gn).dof(gndof).p == true)
      dofs(ecdof) = node(gn).dof(gndof).value; e dof val = corresponding global val
    end
    dofMap(ecdof) = node(gn).dof(gndof).pos
      pos of dof #gndof of global node gn, *e.g.*,6 for endof = 1, en = 2 in e_2
    ecdof = ecdof + 1 increment counter
  end

Step 9: Set element dofs $\mathbf{a}^e$: Simplified limited case

Truss
dofs: u, v $\Rightarrow$
$n_{\text{dpn}} = 2$



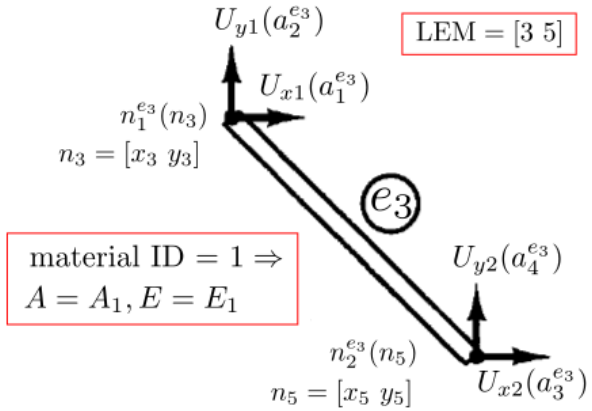
Frame
dofs: u, v, θ $\Rightarrow$
$n_{\text{dpn}} = 3$

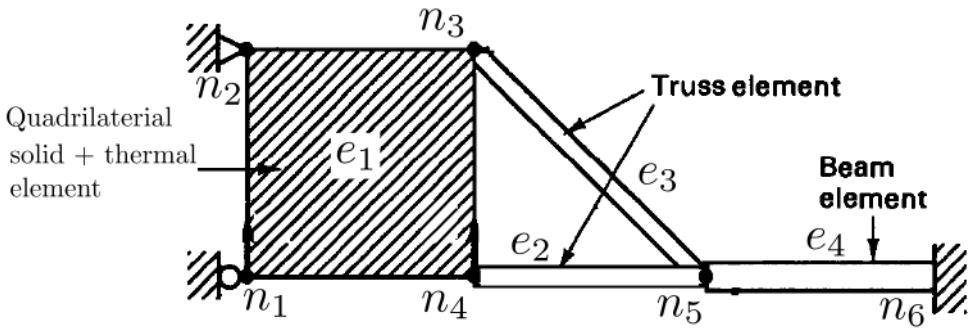


- Similar to steps 1, 2, and 8, step 9 can be greatly simplified if we assume all nodes share exactly the same set of dofs.
- Noting n_{dpn} (ndofpn) = Number of dof. per node, simplified merged steps 8 & 9 are:

```
dofs = zeros(nedof) element dofs (edof) resized to number of element dofs and zeroed
ecdof = 1 dof counter for element
for en = 1: neNodes number of element nodes
   gn = LEM(en) global node number for element node en
   for endof = 1: ndofpn This number is fixed now, e.g., 2 for 2D trusses
      if (node(gn).dof(endof).p == true)    gndof = endof, we bypass some steps here
         dofs(ecdof) = node(gn).dof(endof).value; e dof val = corresponding global val
      end
      dofMap(ecdof) = node(gn).dof(endof).pos
      ecdof = ecdof + 1    increment counter
   end
end
```

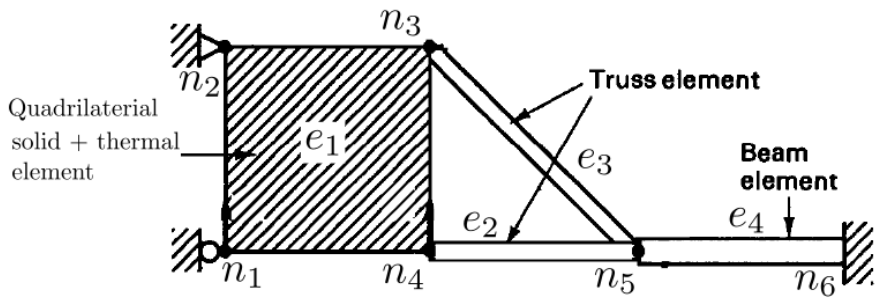
Step 10: Compute element stiffness/force





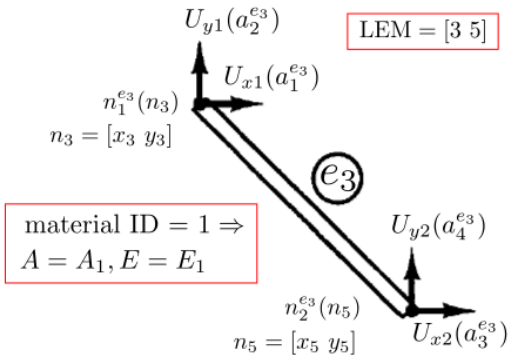
- Almost all assembly of element stiffness/forces are done in a "black box".
- That is, as far as assembly of local to global stiffnesses and forces are considered, there is no need to know how the local stiffness matrix is assembled.
- This is if an object oriented language is used, most of these functions are virtual (*i.e.*, different elements have different implementations for these computations under the same function name).
- For multiphysics problems the assembly of physics into the element is very much similar to assembly of elements into the global system: Each physics assembles its own part into element system which subsequently will be mapped to global system.
- Types of information needed before element stiffness/force computations:
  1. Element type: *e.g.*, truss or bar element; 1st, 2nd order, *etc.*.
  2. Geometry: Geometry of element (*e.g.*, L^e for trusses).
  3. Material properties: Properties needed such as A and E for trusses.

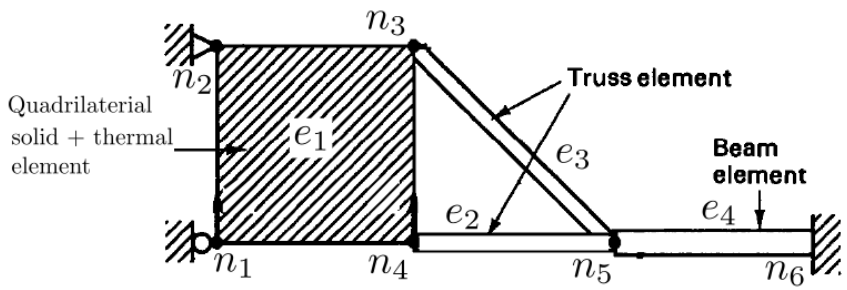
Step 10: Compute element stiffness/force: 1. Element type



- Element type refers to general data needed to clearly specify the element type.
- For example, e_1 is a 2D element of degree one (Q4); elements 2 and 3 are trusses and element 4 is a frame element.
- In general, various specifiers may be used to clearly specify the element type. However, for our purposes we may simply specify element types by one number (*e.g.*, 1: truss, 2: beam, *etc.*).

Step 10: Compute element stiffness/force: 2. Geometry



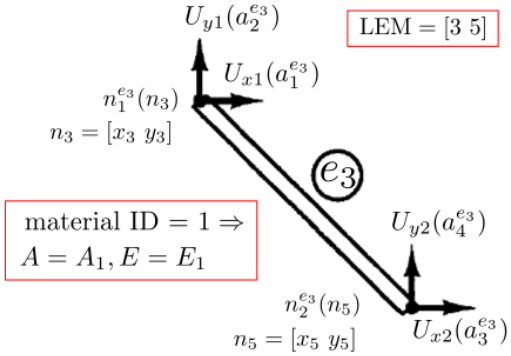


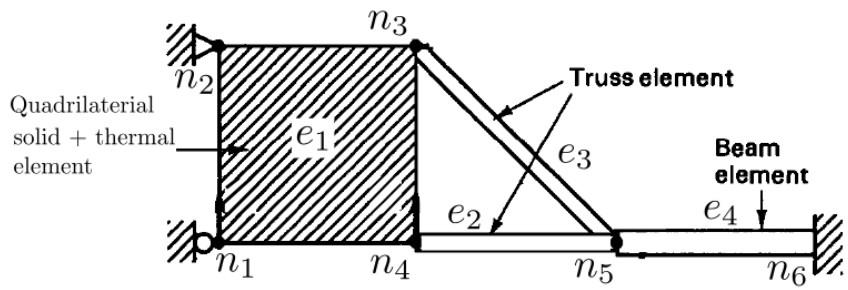
- Geometry of each element is fully described by its nodal coordinates.
- We do not enter nodal coordinates for each elements as a part of its data. This not only considerably enlarges data sets but can easily result in inconsistencies due to repetition of the same nodal coordinate.
- Nodal coordinates are only entered at the global level.
- Through element LEM map (element to global nodal map) we would have element nodal coordinates.
- In the figure for the truss element e_3:

$$\text{LEM}(e_3) = [3\ 5] \Rightarrow \left.\begin{array}{ll} \mathbf{X}_1 & = [x_3\ y_3] \\ \mathbf{X}_2 & = [x_5\ y_5] \end{array}\right\} \Rightarrow \left.\begin{array}{ll} L_x & = x_5 - x_3 \\ L_y & = y_5 - y_3 \end{array}\right\} \Rightarrow$$

$$\boxed{\begin{array}{ll} L^{e_3} & = \sqrt{L_x^2 + L_y^2} \\ c^{e_3} & = \frac{L_x}{L^{e_3}} \\ s^{e_3} & = \frac{L_y}{L^{e_3}} \end{array}} \tag{449}$$

Step 10: Compute element stiffness/force: 3. Material

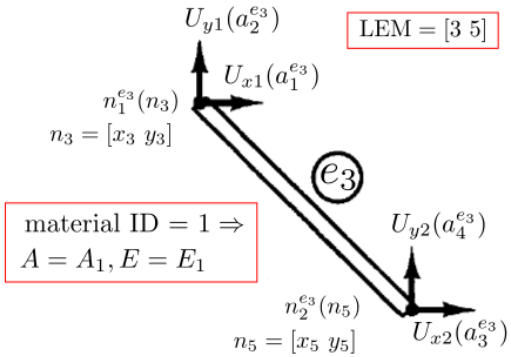


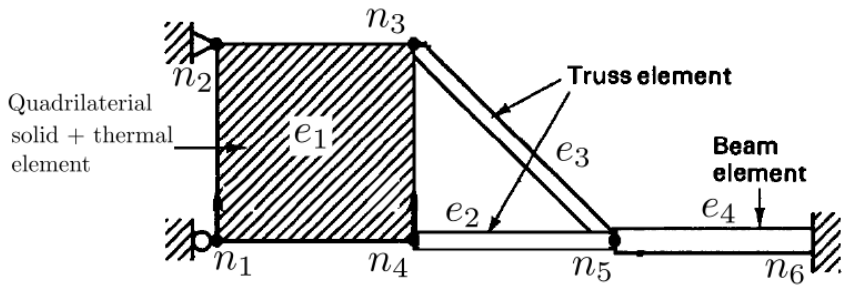


- Element material refers to parameters/flags needed to specify element parameters.
- *e.g.*, for bar/truss elements these are A, E and for beam element are A and I.
- Some are geometric in nature for 1D and 2D element (area section A, thickness t, *etc.*).
- Others are parameters of constitutive equation type; *e.g.*, elastic modulus E.
- Similar to element nodal data, we do not directly enter material parameters for elements.
- Rather, for an FEM simulation we list materials at a global level *cf.* ({mats} on slide 401); *e.g.*,
  numMat 3
  id numPara paras
  1 2 10 10000
  2 3 0.01 100 0.3
- In the sample data input, material refers to truss element material parameters $A = 10, E = 10000$ while material 2 is for 2D solid with $t(\text{thickness}) = 0.01, E = 100, \nu(\text{Poisson ratio}) = 0.3$.
- At element level a matID refers to global {mats}; *e.g.*,

$$\text{For } e_3, \text{matID} = 1, \quad \Rightarrow A = \{\text{mats}\}(1)(1) = 10, E = \{\text{mats}\}(1)(2) = 10000 \qquad (450)$$

Step 10: Compute element stiffness/force: Evaluation





- The calculation of element stiffness and force is specific to the type of the element.
- There are two general approaches to compute stiffness/force:
  1. Closed form (analytical) integration: In some instances, particularly for 1D elements, the integrations in element stiffness and force can be done analytically. For example for bar element we observed:

$$\mathbf{k}^e = \frac{AE}{L^e}\begin{bmatrix} 1 & -1 \\ -1 & 1 \end{bmatrix} \quad \text{(constant } A \text{ and } E\text{)} \quad (cf.\ (374))$$

$$\mathbf{f}_r^e \approx \mathbf{r}^e \begin{bmatrix} q_1 \\ q_2 \end{bmatrix}, \quad \mathbf{r}^e = \frac{L^e}{6}\begin{bmatrix} 2 & 1 \\ 1 & 2 \end{bmatrix} \quad (cf.(378))$$

also for the truss example e_3 (390):

$$\mathbf{k}^e = \frac{AE}{L}\left[\begin{array}{c|c} \mathbf{k}_b & -\mathbf{k}_b \\ \hline -\mathbf{k}_b & \mathbf{k}_b \end{array}\right], \mathbf{k}_b = \begin{bmatrix} c^2 & cs \\ cs & s^2 \end{bmatrix} \text{ for } A = 10, E = 10000, \ \& \ c, s \text{ from (449}$$

Step 10: Compute element stiffness/force: Evaluation

2. Numerical integration (quadrature):
   - Most often, especially for 2D and 3D elements, the analytical evaluation of integrals is either very cumbersome or impossible.
   - This is particularly true for high order iso- and super-parametric elements that will be covered later in the course.
   - Under such circumstances, a numerical integration scheme, *i.e.*, a quadrature, is used to numerically evaluate all stiffness matrix and force vector integrals.

Step 10: Compute element stiffness/force: Forces

- While there is only one stiffness matrix we compute, there are several types of forces that are computed at element level *cf.* slide 295:
  1. Essential BC force $\mathbf{f}_D^e = k^e \mathbf{a}^e$ (*cf.* (363d)).
  2. Natural BC force $\mathbf{f}_N^e = \int_{\partial e_f} \mathbf{N}^{e\mathrm{T}} \bar{\mathbf{F}} . \, \mathrm{d}\mathbf{s}$ (*cf.* (363c)).
  3. Source term force: $\mathbf{f}_r^e = \int_e \mathbf{N}^{e\mathrm{T}} . \mathbf{r} \, \mathrm{dv}$ (*cf.* (363b)).
  4. Other force: Some examples that we did not cover in this course are forces dues to thermal strain in solids. These forces are very specific to the type of physics being solved.
- The only force that takes exactly the same form regardless of physics is $\mathbf{f}_D^e$.
- As we observed, *e.g.*, beam element compared to bar element, $\mathbf{f}_N^e$ does not take exactly the same form and should have a separate implementation for every physics.
- For $\mathbf{f}_N^e$ and $\mathbf{f}_r^e$ we can generic implementations corresponding to definitions above. If for a given element the definitions of these forces matches general expression, we use them otherwise the element should provide its specific implementation.

Step 10: Compute element stiffness/force: Forces

- Motivated by the fact that $\mathbf{f}_D^e$ is the only force that always takes the same form, we defined three force members for elements on slide 393:

$$\mathbf{f}_D^e = \text{Essential BC force}$$
$$\mathbf{f}_o^e = \text{Sum of } \mathbf{f}_N^e, \mathbf{f}_r^e, \text{ and possibly all other loads}$$
$$\mathbf{f}_e^e = \mathbf{f}_D^e + \mathbf{f}_o^e = \text{Total element force}$$

- The calculation of $\mathbf{f}_o^e$ is specific to a given element (and its physics) and should have its own particular implementation.
- The calculation of $\mathbf{f}_D^e$ and $\mathbf{f}_e^e$ (summation of two forces) is generic on the other hand.
- This is the motivation for the definition of these functions for element class on slide 394:
  1. Calculate $\mathbf{f}_o^e$: sum of all forces, but $\mathbf{f}_D^e$ (virtual)
  2. Calculate $\mathbf{f}_D^e = \mathbf{k}^e \mathbf{a}^e$
  3. Calculate $\mathbf{f}_e^e = \mathbf{f}_D^e + \mathbf{f}_o^e$
- Again, the keyword virtual means having the same function of all instances of the object with their own distinct implementation; *cf.* slide 394.

Step 11: Assembly from local to global system

- $\mathbf{K}$ and $\mathbf{F}$ (global stiffness and force) are already sized and set to zero.
- Element level (local) stiffness and force is calculated (Step 10).
- Element local to global dof $\mathbf{M}_t^e$ is already set (Step 8).
- Using dof map, we assemble local values to global values.
- Clearly, only free dofs are added to stiffness matrix and force vector.
- Element dof values (dofs: $\mathbf{a}^e$ is also set (Step 9).
- $\mathbf{f}_D^e = \mathbf{k}^e\mathbf{a}^e$ may not need to be formed and can be directly added to $\mathbf{f}_e^e$.

```
for e = 1:ne loop over elements
  fee = feo element total force = element all forces except essential force
  for i = 1:nedof loop over rows of ke; nedof = element # dof
    I = dofMap(i) local to global dof map M_t^e
    if (I > 0) I corresponds to a free dof, we skip prescribed dofs
      for j = 1:nedof loop over columns of ke
        J = dofMap(j) global dof corresponding to j
        if (J > 0) now both I and J are free and can add ke(i,j) to global K
          K(I, J) = K(I, J) + ke(i, j)
        else J < 0, prescribed dof j; add contributions of f_D^e = k^e a^e to f_e^e
          fee(i) = fee(i) - ke(i, j) * edofs(j) edofs: element dofs = a^e
        end
      end
      F(I) = F(I) + fee(i)element's total force fee component i'th is computed→added to F(I)
    end
  end
end
```

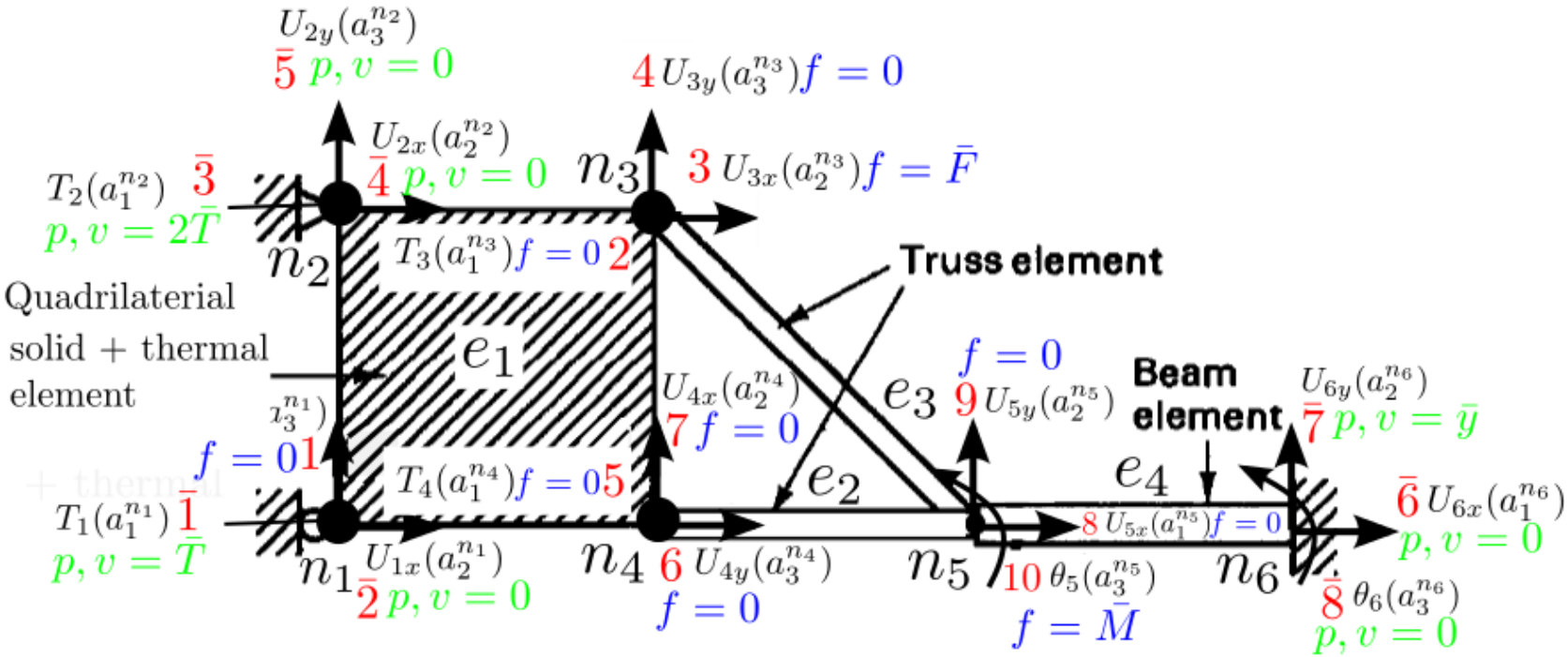
Step 11: Assembly from local to global system

- **Rows**: In loop over edof (element dofs) we skip prescribed dofs i.
- **Columns**: free columns j: Stiffness is assembled $K(I, J) = K(I, J) + ke(i, j)$.
- **Columns**: prescribed columns j: $\mathbf{f}_D^e$ is added to total element force $\mathbf{f}_e^e$: fee(i) = fee(i) - ke(i, j) * edofs(j).
- **Force**: Once fee is updated with fde (essential BD force), it is assembled to global F: F(I) = F(I) + fee(i).
- **Both K and k are symmetric**:
  1. K can be stored into a symmetric matrix → almost half the storage.
  2. ke symmetric: can be stored in symmetric matrix → almost half the storage.
  3. ke symmetric: half the loop computation: instead of looping (i: 1 → nedof; j: 1 → nedof) we can loop **(i: 1 → nedof; j: i → nedof)**. How?

Step 12: Solve global (free) dof $\mathbf{a}$ from $\mathbf{Ka} = \mathbf{F}$

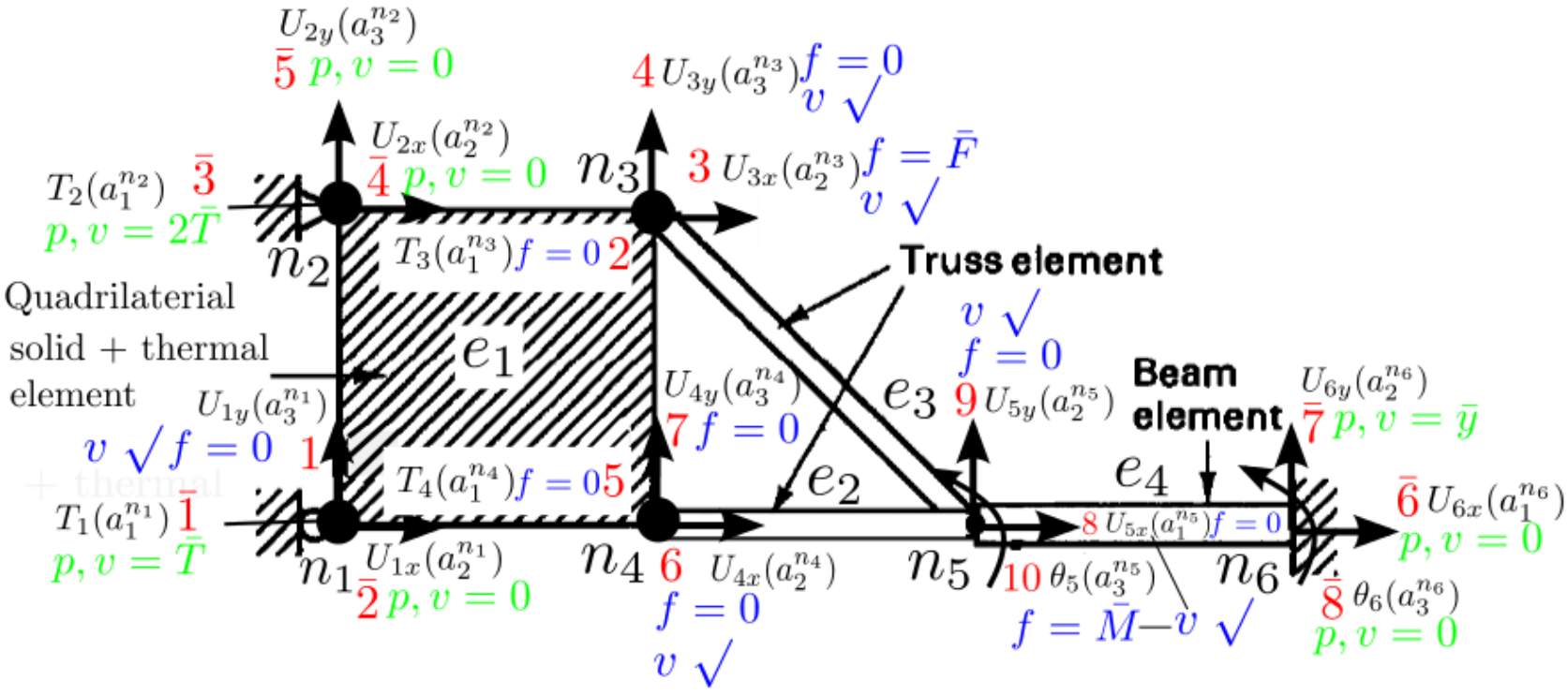
- Two major computational costs during FEM solve are:
  1. Assembly: Refers to: all node, element, and dof set up; computation of local ke and fee; assembly of those to global system. This step scales linearly versus n_{e} (ne)
  2. Linear algebra solution: $\mathbf{Ka} = \mathbf{F}$: We solve for unknown $\mathbf{a}$. Although conceptually simple, this step is a major source of computational cost. It scales higher than linear versus n_{e} $\Rightarrow$ As the problem size increases this term becomes more dominant.
- Solution of $\mathbf{Ka} = \mathbf{F}$:
  - WE DO NOT OBTAIN $\mathbf{a}$ from $\mathbf{a} = \mathbf{K}^{-1}\mathbf{F}$: We do not invert $\mathbf{K}$.
  - We only solve the problem for the specific RHS of $\mathbf{F}$.
  - In Comparison $\mathbf{K}^{-1}$ corresponds to the solution of $\mathbf{Ka} = \mathbf{F}$ for n_{f} RHS of $\mathbf{F} = e_i,\ i = 1, \ldots, n_{\mathrm{f}}$ where n_{f} is the number of rows (and columns) of $\mathbf{K}$.
  - We employ methods such as LU factorization that computationally only solve the problem for the given RHS $\mathbf{F}$.
  - We take advantage of the structure of stiffness matrix: symmetry, bandedness, sparsity in choosing the right solution technique.
  - order of free dofs affects band of the matrix $\rightarrow$ various algorithms reorder free dofs such that the matrix band get smaller and the solution cost is optimized.
  - In your term projects you can simply employ simply compute

Step 13: Assign $\mathbf{a}$ to nodes and elements



- Once we obtain global free dofs ($\mathbf{a}$ dofs) we assign the values of free nodal dof and elements' free dofs:
  1. Nodes: loop over nodes (outer loop) and dofs (inner loop) and for free dofs we set their values equal to corresponding position in $\mathbf{a}$.
  2. elements: dofMap ($\mathbf{M}_t^e$) provides a direct map to global $\mathbf{a}$ and we set element free dof values that were not known before the solution.

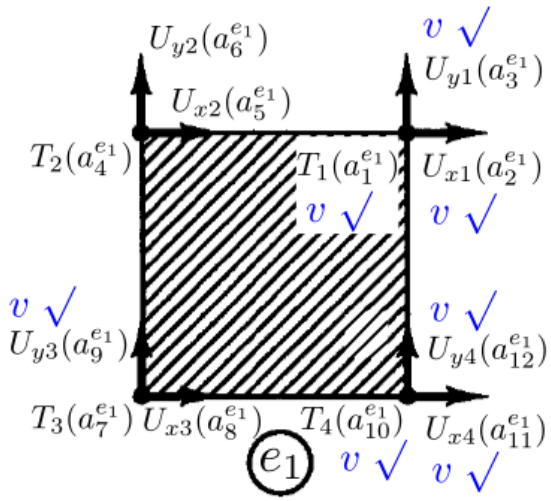
Step 13: Assign $\mathbf{a}$ to nodes and elements: a) nodes



- Loop over nodes (outer loop), loop over nodes' dofs (inner loop), for free dofs set their values equal to their corresponding values in dofs ($\mathbf{a}$).

```
for n = 1:nNodes
   for dofi = 1: node(n).nndof num dof for node (n)
      if node(n).ndof(dofi).p == false free dof
         posn = node(n).ndof(dofi).pos position of dof in global free F
         node(n).ndof(dofi).v = dofs(posn) set free dof val to corresponding val in global dofs (a)
      end
   end
end
```

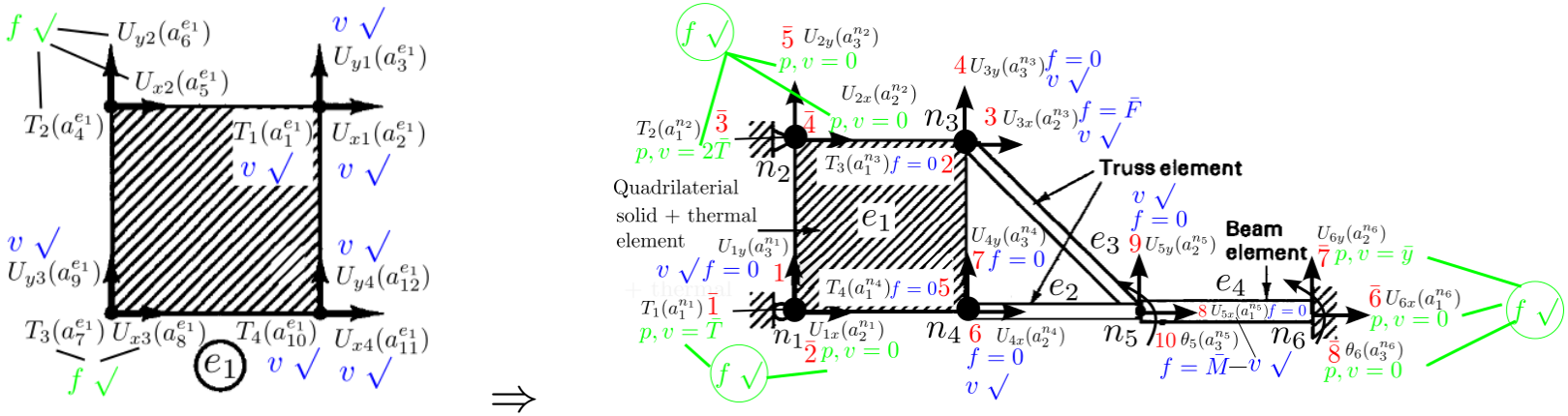
Step 13: Assign $\mathbf{a}$ to nodes and elements: b) elements



- Loop over element (outer loop), dofs of element (inner loop), from the map (dofMap: $\mathbf{M}_t^e$) directly get their corresponding global dof; if a free dof set it equal to corresponding values in dofs ($\mathbf{a}$).

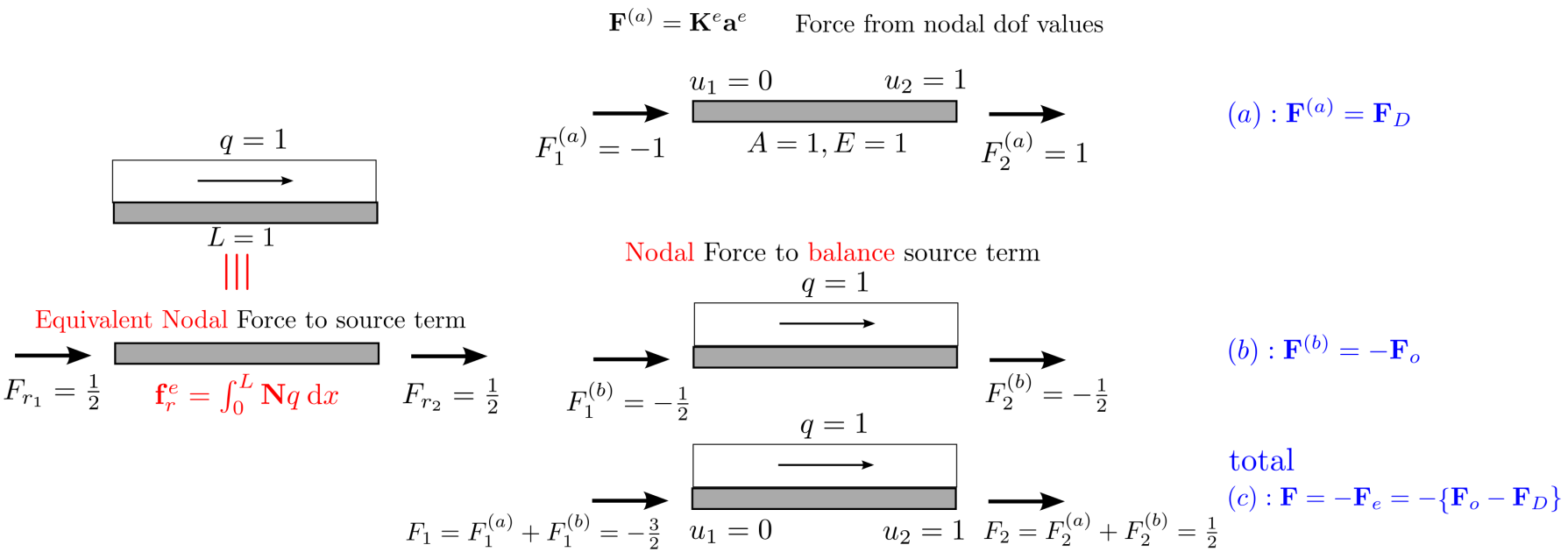
```
for e = 1:ne loop over elements
  for i = element(e).nedof loop over element dofs; nedof = # dof (n^e_dof)
    posn = element(e).dofMap(i) corresponding global position using dofMat (M^e_t)
    if (posn > 0) free dof
      element(e).edofs(i) = dofs(posn)
        set free element dof a^e to corresponding val in global dofs (a)
    end
  end

end
```

Step 14: Compute prescribed dof forces



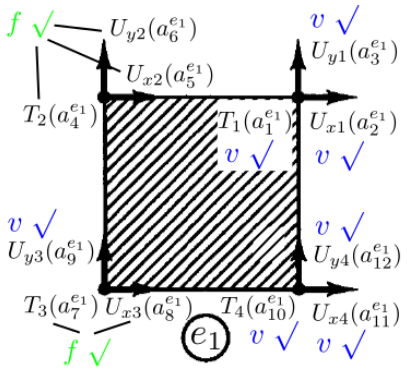
- At global nodes, the only unknowns at this stage are prescribed forces (reaction forces for solid mechanics).
- To set global nodal prescribed forces we need to:
  1. **Compute all element forces**: $\mathbf{f}_o^e$ (all forces but $\mathbf{f}_D^e$ is calculated), we add this to contributions from nodal dof values (previously known prescribed and newly solved free dofs).
  2. **Add element nodal forces to get global nodal forces**: The contribution (sum) of all element forces at a given prescribed dof is equal to the prescribed force (reaction force) at that dof. For convenient we store these values in $\mathbf{F}_p$ (n_p vector).
  3. **Assign prescribed global nodal dof forces** from their corresponding values in $\mathbf{F}_p$.
- **loop over elements → steps 1 & 2; loop over nodes → step 3**.

Step 14: Compute prescribed dof forces: a) elements



- Total forces at element nodes have two sources:
  1. (a) Nodal dof values $\mathbf{f}_D^e$: For example in solid mechanics nodal displacements generate forces. This is the force $\mathbf{f}_D^e = \mathbf{k}^e\mathbf{a}^e$. During solution only **prescribed dofs** in $\mathbf{a}^e$ were known and only their contribution was considered.
  2. (b) All other forces $\mathbf{f}_o^e$: All other forces are **transferred to equivalent nodal forces**. This is shown in figure (b) for sample source term forces. Equivalency means that we can replace all forces but those caused by nodal dof values with $\mathbf{f}_o^e \Rightarrow$ To balance all other contributions with nodal forces, negative of $\mathbf{f}_o^e$ is exerted at nodes.
- The total nodal forces at nodes are $\mathbf{f}_D^e - \mathbf{f}_o^e$ which is $-\mathbf{f}_e^e$ (the total element **equivalent** force used during assembly).

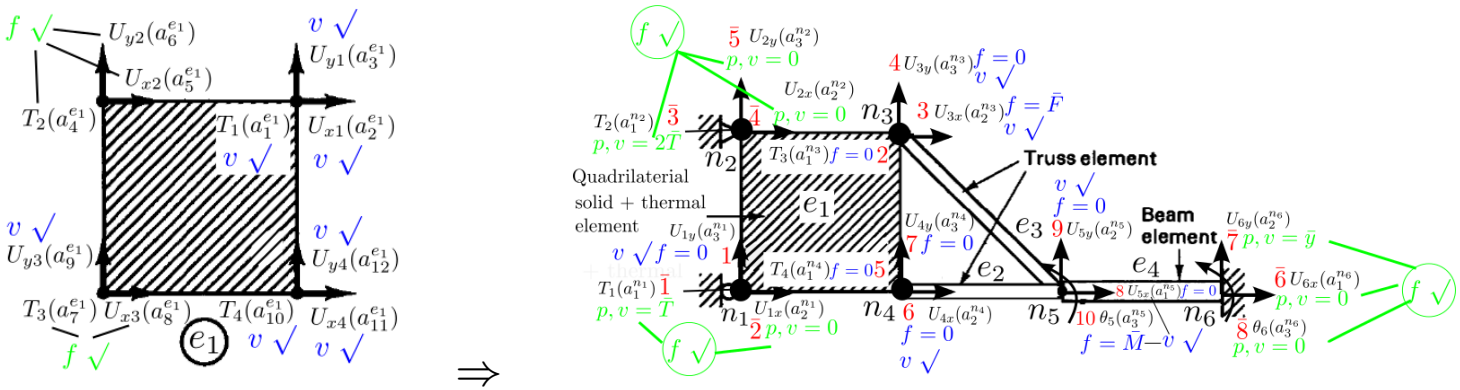
Step 14: Compute prescribed dof forces: a) elements



- Steps for calculation of $\mathbf{F}_p$ from elements: 1. calculation of element nodal forces; 2. Assembling those to correct position in $\mathbf{F}_p$. Corresponding pseudocode:

```
for e = 1:ne loop over elements
  fee = feo element total force = element all forces except essential force
  for i = 1:nedof loop over rows of ke; nedof = element # dof
    I = dofMap(i) local to global dof map M_t^e
    if (I < 0) I corresponds to a prescribed dof, we skip free dofs
      for j = 1:nedof loop over columns of ke. ALL columns (dofs) of p and f used
        fee(i) = fee(i) - ke(i, j) * edofs(j) edofs: element dofs = a^e
      end
      Fp(-I) = F(-I) - fee(i)
        1. element's total force fee component i'th is computed→added to Fp(-I)
        2. -I used because I < 0: prescribed dof
        3. fee is subtracted
    end
  end
end
```

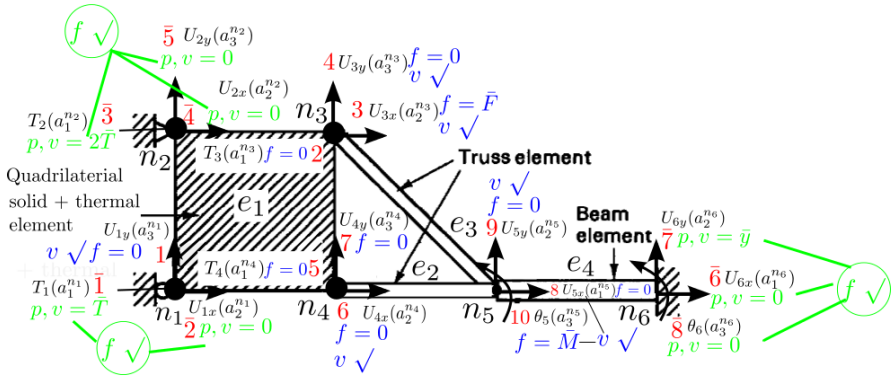
Step 14: Compute prescribed dof forces: b) nodes



- Now that $\mathbf{F}_p$ is computed, we need to loop over nodes, and their prescribed dofs to set them to corresponding forces in $\mathbf{F}_p$. Corresponding pseudocode:

```
for n = 1:nNodes
   for dofi = 1: node(n).nndof num dof for node (n)
      if node(n).ndof(dofi).p == true prescribed dof
         posn = node(n).ndof(dofi).pos position of dof in global prescribed force F_p
         node(n).ndof(dofi).f = Fp(-posn)
            1. set prescribed dof force to corresponding force in global Fp (F_p)
            2. posn < 0; prescribed dof
      end
   end

end
```

Step 15: Compute/output nodes & elements: a) nodes

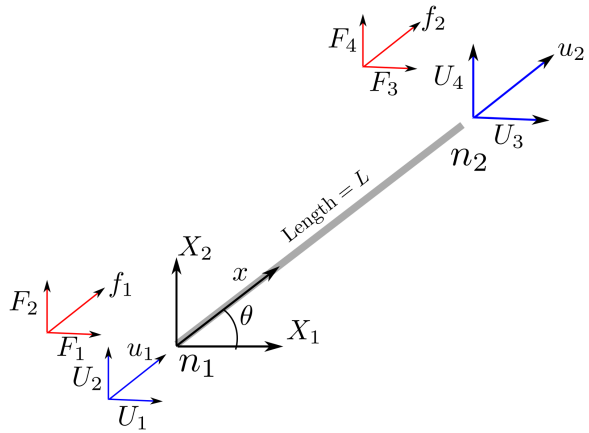


- After free dof values and prescribed dof forces are calculated, we can output the results.
- Node output is general, while element output is specific to its type. Sample pseudocode:

```
for n = 1:nNodes
   print: node (n) from (nNodes) nodes
   coord: (node(n).coordinate)
   for dofi = 1: node(n).nndof num dof for node (n)
      print dof:
         dof (dofi) from (node(n).nndof) dofs
            Field: (node(n).ndof(dofi).Field)
            Index: (node(n).ndof(dofi).index)
            Value: (node(n).ndof(dofi).value)
            Force: (node(n).ndof(dofi).force)
            prescribed: (node(n).ndof(dofi).p) may only be output for debugging purposes
            position: (node(n).ndof(dofi).position) may only be output for debugging purposes
   end
end
```

- Refer to slides 398 and 397 for other data members in node and dof objects.

Step 15: Compute/output nodes & elements: b) elements



- The computation and output of an element is again a black box function.
- Each element has its own specific compute/output function.
- For example, for truss elements we only need to compute axial force (figure above, also *cf.* 315 and (387)):

$$T = AE\{c(U_3 - U_1) + s(U_4 - U_2)\}$$

- For a beam element, moment M and shear force V change along the beam and they need a completely different process for computation (*cf.* page 374).
- These functions will be virtual in an object-oriented language. Sample pseudocode:
- For example, the virtual function's implementation for a truss element computes and outputs T from equation above/

```
for e = 1:ne loop over elements
   elements(e).Compute_Output_Element() virtual function on page 394
end
```

Simplified implementation in Matlab

- After covering more general FEM implementation, we focus on the implementation of a more simplified (and limited version) in Matlab.
- The simplifications are:
  1. Fixed set and number of dof per node (*cf.* slide 410 and (448)):

$$n_{\mathrm{dpn}} := \text{ndofpn is a fixed number}$$

  2. No physics, fields, components hierarchy: As ndofpn (fixed number of dof per node) exists for the problem, we do not need to maintain the objects physics, fields, and components in elements.
  3. Limited sources of force: Natural and Essential boundary conditions and concentrated forces are the only forces.
- Also, we very briefly discuss the options to implement the FEM solver without actually resorting to object-oriented approaches.

Simplified objects: 1. Element

Given the simplifications in this section, the element object described from slide 392 to 394 can be fully described by:

- neNodes (n_{n}^e): Number of element nodes (*e.g.*, for element 1 $n_{\mathrm{n}}^e = 4$).
- eNodes (LEM): Indices of element nodes in global system; *e.g.*, for e_1:
- number of element dof (nedof: n_{f}^e): In fact, (nedof $=$ neNodes $\times$ ndofpn).
- edofs ($\mathbf{a}^e$): n_{f}^e vector of dofs.
- dofMap ($\mathbf{M}_t^e$): map from element to global dofs.
- stiffness matrix (ke: $\mathbf{k}^e$): $n_{\mathrm{f}}^e \times n_{\mathrm{f}}^e$ local stiffness (conductivity, *etc.*) matrix.
- force vectors (fde, foe, fee: $\mathbf{f}_D^e, \mathbf{f}_o^e, \mathbf{f}_e^e$): Essential BC, forces other force, and total force vectors. For term project, $\mathbf{f}_o^e = 0$, so $\mathbf{f}_D^e$ and $\mathbf{f}_e^e$ or even one of them are sufficient.
- eType: One Digit encapsulates element type. Let's adapt the following convention:

$$\text{eType } = 1 : \text{bar element} \qquad \text{eType } = 2 : \text{beam element} \tag{451}$$

$$\text{eType } = 3 : \text{truss element} \qquad \text{eType } = 4 : \text{frame element} \tag{452}$$

$$\text{eType } = 5 : \text{2D thermal element} \qquad \text{eType } = 6 : \text{2D solid element} \tag{453}$$

Simplified objects: 2. (Combined) Node + Dof

Again from our simplifications, the node and dof objects on slides 397 and 398 can be combined into,

- coordinate: a vector of size dimension of node coordinates.
- prescribed (dof_p) vector of size ndofpn having dof booleans of being prescribed dof.
- position (dof_pos) vector of size ndofpn having dof positions.
- value (dof_v) vector of size ndofpn having dof values.
- force (dof_f) vector of size ndofpn having dof forces.

Notable simplifications are:

1. All members starting with "dof_" are all that was used to be inside the set ({dof}) as a member of node object (*cf.* to {ndof} on slide 397).
2. Node's member {nndof} is no longer required as it is always equal to ndofpn.

Simplified objects: 3. FEM Solver

Following the same trend, FEM Solver on slide 401 is simplified to :

- dim, n_{dim} spatial dimension for the problem (1D, 2D, and 3D)
- number of nodes (nNodes, n_{n}).
- nodes {node}: vector of nodes in the domain.
- number of elements (ne, n_{e}) in the the domain.
- elements {element}: vector of elements in the domain.
- free dofs (dofs: $\mathbf{a}$): vector of global free dofs.
- number of free dof (nf, n_{f}).
- number of prescribed dof (np, n_{p}).
- number of dof (ndof, $n_{\mathrm{dof}} = n_{\mathrm{f}} + n_{\mathrm{p}}$).
- stiffness matrix ($\mathbf{K} = \mathbf{K}_{ff}$); $n_{\mathrm{f}} \times n_{\mathrm{f}}$ matrix.
- force vector ($\mathbf{F} = \mathbf{F}_{f}$); $n_{\mathrm{f}}(\times 1)$ vector.
- prescribed force vector ($\mathbf{F}_{p}$); $n_{\mathrm{p}}(\times 1)$ vector.
- number of materials: nmats.
- material database {mats}: a nmat size cell(Matlab) of vectors.

Object-oriented language

- Among many advantages of objected-oriented language we observe that objects encapsulate data.
- For example, for ne element, we only need a set (cell in Matlab) of elements. A different approach is not to use classes and represent "elements" by several vectors all of size ne.
- If we do not want to use objects (classes) we can do the following in Matlab:
  1. FEMSolver: A Function (*e.g.*, FEMSolver function) has all members listed on slide 446 as its internal parameters.
  2. nodes: Alternative to {node}, where "node" refers to node object on slide 445:
     - node_coordinate: Matrix (nNodes $\times$ dim) of node coordinates.
     - prescribed (node_dof_p): Matrix (nNodes $\times$ ndofpn) of node prescribed dof flags.
     - position (node_dof_pos): Matrix (nNodes $\times$ ndofpn) of node dof positions.
     - value (node_dof_v): Matrix (nNodes $\times$ ndofpn) of node dof values.
     - force (node_dof_f): Matrix (nNodes $\times$ ndofpn) of node dof forces.

Alternative to object-oriented Programming: Solver, Nodes

- Among many advantages of objected-oriented language we observe that objects encapsulate data.
- For example, for ne element, we only need a set (cell in Matlab) of elements. A different approach is not to use classes and represent "elements" by several vectors all of size ne.
- If we do not want to use classes(objects) we can do the following in Matlab:
  1. FEMSolver: A Function (*e.g.*, FEMSolver function) has all members listed on slide 446 as its internal parameters.
  2. nodes: Alternative to {node} ("node" refers to node object on slide 445):
     - node_coordinate: Matrix (nNodes $\times$ dim) of node coordinates.
     - prescribed (node_dof_p): Matrix (nNodes $\times$ ndofpn) of node prescribed dof flags.
     - position (node_dof_pos): Matrix (nNodes $\times$ ndofpn) of node dof positions.
     - value (node_dof_v): Matrix (nNodes $\times$ ndofpn) of node dof values.
     - force (node_dof_f): Matrix (nNodes $\times$ ndofpn) of node dof forces.

Alternative to object-oriented Programming: Elements

3. elements: Alternative to {element} (set of element object; *cf.* slide 444):
   - element_neNodes (n_{n}^e): ne vector; element_neNodes(en) = number of nodes for element en.
   - element_eNodes (LEM): ne cell; element_eNodes{en} = vector of size element_neNodes(en) of element en node ids.
   - element_nedof (n_{f}^e): ne vector; element_nedof(en) = number of dofs (= neNodes $\times$ ndofpn) for element en.
   - element_edofs ($\mathbf{a}^e$): ne cell; element_edofs{en} = vector of size element_nedof(en) of element dofs.
   - element_dofMap ($\mathbf{M}_t^e$): ne cell; element_dofMap{en} = vector of size element_nedof(en) of element dof positions in global system (dofMap).
   - element_ke ($\mathbf{k}^e$): ne cell; element_ke{en} = element stiffness matrix of size element_nedof(en) $\times$ element_nedof(en).
   - element_fee ($\mathbf{f}_e^e$): ne cell; element_fee{en} = element total force vector of size element_nedof(en).
   - element_foe ($\mathbf{f}_o^e$): ne cell; element_foe{en} = element other forces (all but essential) vector of size element_nedof(en).
     only needed if we have "other" load types (*e.g.*, source term).
   - element_eType: : ne vector; element_eType(en) = one integer (eType) specifying the element en's type; *cf.* (451).

Alternative to object-oriented Programming: Materials

4. Material properties: Alternative to {mats} (set of material properties objects; *cf.* slide 446):
   - mat: nmats cell; nmats{en} = vector of arbitrary size that defines the material needed for corresponding elements. For the following element types the order of material parameters are given,

eType $= 1$	Bar	mat$\{en\} = [E\ A]$	(454a)
eType $= 2$	Beam	mat$\{en\} = [E\ I]$	(454b)
eType $= 3$	Truss	mat$\{en\} = [E\ A]$	(454c)
eType $= 4$	Frame	mat$\{en\} = [E\ A\ I]$	(454d)
eType $= 5$	2D thermal	mat$\{en\} = [\kappa\ t]$	(454e)
eType $= 6$	2D solid	mat$\{en\} = [E\ \nu\ t]$	(454f)

Alternative to object-oriented Programming: Summary

- In simplified FEMSolver class there were sets of three objects: {nodes}, {elements}, {mats}.
- On slides 448, 449, and 450 respectively we discussed how to replace these members as well as FEMSolver as on object with simple Matlab vectors and cells.
- It is evident in this alternative approach, we would be dealing with larger number of data at solver level and book keeping of them can be more difficult.
- Also, we lose the notion of objects in this approach.
- Difference between vectors and cells in Matlab:

	data indexed	indexed by
vector	numbers	(i)
cell	arbitrary: objects(*e.g.*, tensors) of different types & sizes	{i}

- In Matlab, we need to use cells to store variable data. For example, on slide 449 cells are used for storing possibly variable data (*e.g.*, vectors of variable size). Same can be observed on slide 450.

Alternative to virtual functions

- We have encountered several functions that are shared among all different realizations of an object, but may have different implementations. Some examples were:
  1. Compute element stiffness matrix $\mathbf{k}^e$.
  2. Compute various forces for elements: $\mathbf{f}_r^e$, $\mathbf{f}_N^e$, and in general all forces contributing to $\mathbf{f}_o^e$.
  3. Element output functions. For example for two node bar element we only need to output the axial force, whereas for beam elements a different output is needed.
- In we are not employing an object oriented language along with polymorphism (to enable virtual functions) we can compute or operate the right routines based on element type for all examples above:

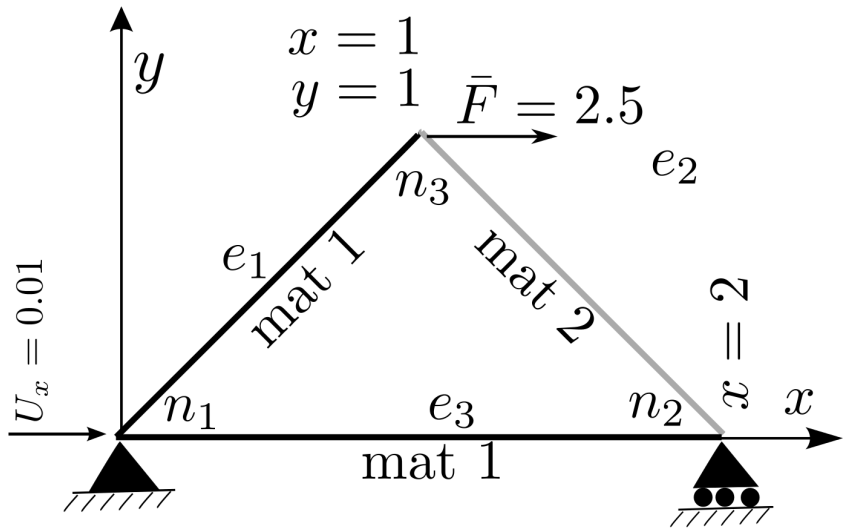
  Compute Stiffness:
  if (eType $==$ 1) % implementation to calculate ke for bar elements.
  elseif (eType $==$ 2) % ke for beam element
  $\vdots$
  end

Input file format

- Our input file has the following blocks:
  1. Header: It includes dim and ndofpn
  2. Nodes
  3. Elements
  4. Prescribed dofs
  5. Free dofs (with nonzero nodal force)
  6. Materials
- A sample input file is shown on the next page.
- We include description of all inputs (*e.g.*, id elementType matID neNodes eNodes) to make input file more readable.

Input file format

```
dim 2
ndofpn 2
Nodes
nNodes 3
id crd
1 0 0
2 2 0
3 1 1
Elements
ne 3
id elementType matID neNodes eNodes
1 3 1 2 1 3
2 3 2 2 3 2
3 3 1 2 1 2
PrescribedDOF
np 3
node node_dof_index value
1 1 0.01
1 2 0
2 2 0
FreeDOFs
nNonZeroForceFDOFs 1
node node_dof_index value
3 1 2.5
Materials
nMat 2
id numPara Paras
1 2 100 1
2 2 200 2
```



$E_1 = 100, A_1 = 1$
$E_2 = 200, A_2 = 2$

Output file format

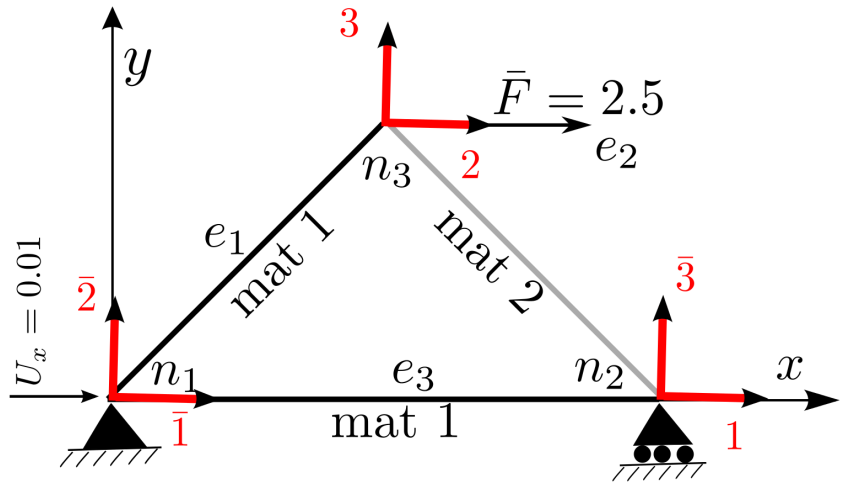
- Main FEMSolver function would be:

  function FEMSolver(runName, verboseOutput)

- runName is the name of the run (*e.g.*, TrussA).
- FEMSolver opens a file with name runName.txt to read the problem (with the format conforming to the sample on slide 456).
- This function, computes and outputs results in file runName.out.
- verboseOutput: if it is true more data will be output (explained on the next slide).
- Format of output file is
  1. Nodes: Node dof values and forces (plus dof position and prescribed boolean with verbose option).
  2. Elements: All element nodal forces (plus element specific output; *e.g.*, axial force for bars and trusses, *etc.*).

Output file format

```
Nodes
nNodes 3
id crd
values
forces
position(verbose)
prescribed_boolean(verbose)
1 0 0
a1_1 a1_2
F1_1 F1_2
-1 -2 (verbose)
1 1 (verbose)
2 2 0
a2_1 a2_2
F2_1 F2_2
1 -3 (verbose)
0 1 (verbose)
a3_1 a3_2
F3_1 F3_2
2 3 (verbose)
0 0 (verbose)
Elements
ne 3
id elementType
forces(verbose)
specific output
1 3
fee1_1 fee1_2 fee1_3 fee1_4 (verbose)
Te1
2 3
fee2_1 fee2_2 fee2_3 fee2_4 (verbose)
Te2
3 3
fee3_1 fee3_2 fee3_3 fee3_4 (verbose)
Te3
```



- lines with (verbose) are only output for verboseOutput == 1. Obviously (verbose) is not printed in either case and is only printed for clarity here.
- ai_j: is value (solution) for node i dof number j;*e.g.*, a3_1 is x displacement at node 3 ($x = 1, y = 1$).
- Fi_j: is force for node i dof number j;*e.g.*, F3_1 is x force at node 3 (which should be equal to 2.5, why?)
- feei_j: is total force (foe + fDe) for element i dof number j;*e.g.*, fee3_1 is the x force at its left node (global n_1).
- Last item of element output is specific to its type.
- For 2 node bar and truss elements Tei is the axial force in the element.

Programming hints for Matlab

- In Matlab input arguments can only be passed to functions by copy.
- That is, Matlab does not provide the option of sending arguments by reference.
- One approach to limit the number of arguments passed to functions is to have some of input arguments global in the function and from where the function is called.
- This not only reduces memory overhead by argument copy (first item) but also reduced the number of input arguments which are generally much larger if classes are not employed (*e.g.*, compare sending an element object to a function versus all its internal data through arguments on slide 449).
- As mentioned before, If classed are not employed various data (*e.g.*, those pertained to elements) should be stored in cells and Matlab vectors (or matrices) are not appropriate.
- Finally, functions that depend on the type of element can have internal "if else or switch" statements based on elementType, instead of streamlined polymorphism (virtual function) aspect offered by object oriented languages.