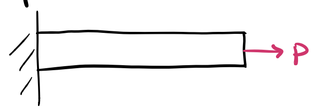
Finite Elements: Putting it All Together

① Start with differential equation ("strong form")

→ equilibrium equation: $\frac{d}{dx}(A\sigma) + bA = 0$

area, stress, body force, area

→ problem statement + boundary conditions:



② Derive weak form: mathematically equivalent to strong form

... for this particular problem, the weak form looks something like:

$$P\,\delta u(L) - \int_0^L \frac{d}{dx}(\delta u) AE \frac{du}{dx}\,dx + \int_0^L \delta u\, A L = 0$$

point of deriving weak form?

weak form only has first derivatives of u, no second derivatives

③ Make approximation with FE to weak form

$$u(x) = \sum_{I=1}^{NN} N_I d_I\,, \quad \delta u = \sum_{I=1}^{NN} N_I(x)\,\delta u_I$$

key idea: use FEM to control error/degree of freedom of approximation through shape functions $N_I(x)$, i.e. linear, quadratic, etc.

④ Obtain governing FEM equations: $Kd = F^{ext}$

$$\underbrace{\left[\int_0^L B^T AE B\,dx\right]}_{K} d = \underbrace{N^T(L)P + \int_0^L N^T bA\,dx}_{F^{ext}}$$

⑤ Convert element integrals to parent domain ξ ξ: −1 —— 1

$$f^{body} = \int_0^L N^T bA\,dx = \int_{-1}^{1} N^T(\xi)\, bA \frac{dx}{d\xi}\,d\xi$$

main benefit of this: ① no need to generate new shape functions for each element
② domain of integration is fixed

⑦ If keep domain of integration fixed, evaluate these integrals via numerical integration → Gaussian Quadrature

$$\int_{-1}^{1} N^T(\xi)\, bA \frac{dx}{d\xi}\,d\xi \approx \sum_{i=1}^{NP} N^T(\xi_i)\, bA \frac{dx}{d\xi} W_i$$

key idea: can integrate any order polynomial (integrands) exactly if choose right # of integration points

Quadratic Elements in 1-D

(first MATLAB hw will be implementing linear + quadratic elements)

General Formula for deriving Lagrange Interpolants (or shape functions)

$$N_J(x) = \prod_{\substack{1 \le I \le n+1 \\ I \ne J}} \frac{(x - x_I)}{(x_J - x_I)} \qquad n = \text{order of polynomial}$$

→ have shape function N_J that trying to find, that's function of x
→ way you get it is not a summation, but a multiplication (what π symbol is)

Example: Linear Shape Function

$$N_1(x) = \frac{(x - x_2)}{(x_1 - x_2)} \quad \text{so } J = 1, \quad N_2(x) = \frac{(x - x_1)}{(x_2 - x_1)}$$

... use ξ instead of x:

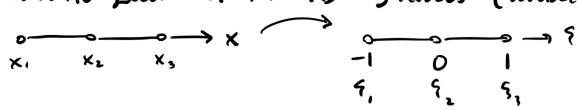
$$N_1(\xi) = \frac{\xi - \xi_2}{\xi_1 - \xi_2}, \quad N_2(\xi) = \frac{\xi - \xi_1}{\xi_2 - \xi_1}$$

... now more specifically, let's say $\xi = -1$ and $\xi_2 = 1$ because those are boundaries of parent domain

$$\rightarrow N_1(\xi) = \frac{1 - \xi}{2}, \quad N_2(\xi) = \frac{1 + \xi}{2}$$

same as before!

Quadratic Element in 1D: 3 nodes (while linear in 1D has 2 nodes)



$$N_1(\xi) = \frac{(\xi - \xi_2)(\xi - \xi_3)}{(\xi_1 - \xi_2)(\xi_1 - \xi_3)} \rightarrow \text{substitute in } \xi\text{'s} \Rightarrow \frac{(\xi - 0)(\xi - 1)}{(-1)(-2)} = \frac{1}{2}\xi(\xi - 1)$$

can immediately see that shape function is quadratic in ξ

$$N_2(\xi) = \frac{(\xi - \xi_1)(\xi - \xi_3)}{(\xi_2 - \xi_1)(\xi_2 - \xi_3)} \Rightarrow \frac{(\xi + 1)(\xi - 1)}{(-1)(1)} = 1 - \xi^2$$

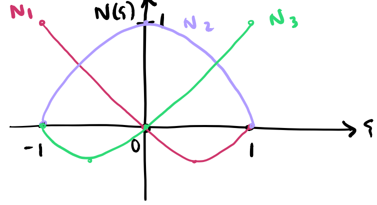
... also quadratic with ξ

$$N_3(\xi) = \frac{(\xi - \xi_1)(\xi - \xi_2)}{(\xi_3 - \xi_2)(\xi_3 - \xi_1)} \Rightarrow \frac{(\xi + 1)(\xi)}{(1)(2)} = \frac{1}{2}\xi(\xi + 1)$$

Check on shape functions:

$N_1 + N_2 + N_3 = 1$ (same property we had for linear shape functions)

Plot shape functions:



$N_1 = 1$ at $\xi = -1$, $= 0$ at $\xi = 0$, $= 0$ at $\xi = 1$ and quadratic in between

same as before: N is 1 at node, 0 at other nodes, quadratically varying in between nodes

Jacobian Mapping: multiply shape functions by coordinates

$$x(\xi) = N(\xi)x = \left[\tfrac{1}{2}\xi(\xi-1) \quad 1-\xi^2 \quad \tfrac{1}{2}\xi(\xi+1)\right]\begin{bmatrix} x_1 \\ x_2 \\ x_3 \end{bmatrix}$$

$$\underset{\text{Jacobian}}{J} = \frac{dx}{d\xi} = \frac{dN}{d\xi}x$$

$$= \left[\xi - \tfrac{1}{2} \quad -2\xi \quad \xi + \tfrac{1}{2}\right]\begin{bmatrix} x_1 \\ x_2 \\ x_3 \end{bmatrix}$$

$$= x_1\left(\xi - \tfrac{1}{2}\right) + x_2(-2\xi) + x_3\left(\xi + \tfrac{1}{2}\right)$$

$$J = \xi\underbrace{(x_1 - 2x_2 + x_3)}_{=0\ \ldots\ \text{if the nodes are evenly spaced}} + \tfrac{1}{2}\underbrace{(x_3 - x_1)}_{L^e\ \ldots\ \text{length of the element}}$$ ← assume nodes are spaced evenly

$\boxed{J = \frac{L^e}{2}}$ → same as linear element assuming nodes are regularly spaced

---

Numerical Integration of terms using quadratic $N(\xi)$

$$f^{body} = \int_0^L N^T(x)\,bA\,dx = \int_{-1}^{1} \underbrace{N^T(\xi)}_{\text{shape function maps from } x \text{ to } \xi} \underset{\text{body force}}{b}\,\underset{\text{area}}{A}\,\underset{\text{Jacobian}}{\frac{dx}{d\xi}}\,\underset{\text{integrating over } \xi}{d\xi}$$

using 1-point quadrature → is it sufficient?

$$f^{body} = \sum_{I=1}^{1} \frac{bAL}{2}\begin{bmatrix} \frac{1}{2}\xi(\xi-1) \\ 1-\xi^2 \\ \frac{1}{2}\xi(\xi+1) \end{bmatrix}\Bigg|_{\xi=0} \underset{\text{weight}}{(2)} = bAL\begin{bmatrix} 0 \\ 1 \\ 0 \end{bmatrix}$$

sum: bAL

problem: can integrate linear function exactly
→ can't integrate exactly with 1-point because integrand is quadratic

we call this "under integrated"

using 2-point integration $\xi = \pm\frac{1}{\sqrt{3}}$, $w = 1$

$$f^{body} = \frac{bAL}{2}\left(\begin{bmatrix} \frac{1}{2}\left(-\frac{1}{\sqrt{3}}\right)^2 - \frac{1}{2}\left(-\frac{1}{\sqrt{3}}\right) \\ 1-\left(-\frac{1}{\sqrt{3}}\right)^2 \\ \frac{1}{2}\left(-\frac{1}{\sqrt{3}}\right)^2 + \frac{1}{2}\left(-\frac{1}{\sqrt{3}}\right) \end{bmatrix}(1) + (1)\begin{bmatrix} \frac{1}{2}\left(+\frac{1}{\sqrt{3}}\right)^2 - \frac{1}{2}\left(+\frac{1}{\sqrt{3}}\right) \\ 1-\left(+\frac{1}{\sqrt{3}}\right)^2 \\ \frac{1}{2}\left(+\frac{1}{\sqrt{3}}\right)^2 + \frac{1}{2}\left(+\frac{1}{\sqrt{3}}\right) \end{bmatrix}\right)$$

$$= \frac{bAL}{2}\begin{bmatrix} 1/3 \\ 4/3 \\ 1/3 \end{bmatrix}$$

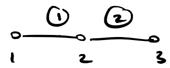
sum: bAL

→ middle node gets 4x body force as other 2 nodes
(while linear elements, body force distributed equally between 2 nodes)

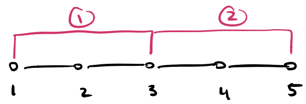
Now how do we assemble these elements together?

Assembly of Quadratic Elements

→ review: linear :



→ quadratic :



→ each have 3 nodes, share a common node at node 3

body force assembly:

$$f^{body} = \frac{bAL}{2}\begin{bmatrix} 1/3 \\ 4/3 \\ 1/3 \end{bmatrix} + \frac{bAL}{2}\begin{bmatrix} 1/3 \\ 4/3 \\ 1/3 \end{bmatrix}$$

element 1

element 2

coupled at node 3

$$= \frac{bAL}{2}\begin{bmatrix} 1/3 \\ 4/3 \\ 2/3 \\ 4/3 \\ 1/3 \end{bmatrix}$$

①

⇒ 2 quadratic elements gives 5-degrees of freedom (in 1-D)

⇒ 2 linear elements gives 3-degrees of freedom (in 1-D)

implication: quadratic k is 3x3, linear k is 2x2 (1 element)

3 nodes   2 nodes

②

⇒ quadratic element requires 2 integration points per element

⇒ linear element requires 1 integration point per element (to integrate body force)

implication: calculating k and f^{body} takes twice as long for 1 quadratic element as compared to 1 linear element (loop over twice)

⇒ so total computational cost assessment: (1 element in 1D)

→ getting k takes 2x as long

→ solving $kd = F$ takes at least 1.5x as long

} adds up to at least 3x as long for same # of elements ... and this is just in 1D!

Sources of Error in FEM Simulations

① Boundary error / domain approximation

→ basically saying FEM approximation not exactly geometry of part (not as much of an issue now a days with CAD, other software)

② Quadratic / numerical precision error

→ small error though

③ Error due to solution approximation (that we introduce as user) i.e choice of FEM shape function

⊛ BIGGEST SOURCE OF THE 3!

⇒ Ways to measure/quantify error ③ :
① for solid problems: measure error in displacement (L2 norm)
② " " " measure error in strains (H1 norm)

$$\text{L2 norm: } \| u_{exact} - u_{FEM} \| = \left[\int_a^b |u_{exact} - u_{FEM}|^2 dx \right]^{1/2}$$

integrate displacement field over domain, square, take sqroot

$$\text{H1 norm: } \left\| \frac{du_{exact}}{dx} - \frac{du_{FEM}}{dx} \right\| = \left[\int_a^b \left| \frac{du_{exact}}{dx} - \frac{du_{FEM}}{dx} \right|^2 dx \right]^{1/2}$$

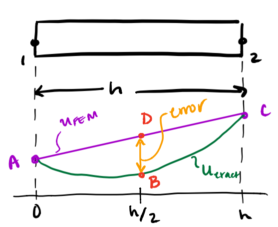
→ evaluate error at integration points
→ typically use more integration points to calculate error than are needed to integrate weak form exactly
→ examine element performance via convergence study
idea: as you use more elements to solve same problem over and over, error should decrease
typically:
mesh 1 size: h, then resolve
mesh 2 size: h/2, then resolve
mesh 3 size: h/4, then resolve
} ① error should decrease w/smaller mesh size
② how fast will error decrease?

⇒ can prove convergence analytically for linear problems
... so you can predict how much error will decrease when reduce your mesh size

---

Convergence of FEM Solutions (error)
↳ how fast does your error decrease?
→ metric as a function of # of elements
→ assumes knowledge of analytic solution

Example: linear finite element ... assume one linear element w/ 2 nodes



→ because linear element, displacement field also linear
⇒ linear FEM displacement u_{FEM}

→ assume exact displacement field ABC is quadratic

u_{exact} = quadratic = $a + bx + cx^2$ ⇒ $u_{exact}(0) = a$, $u_{exact}(h) = a + bh + ch^2$
$u_{FEM} = a + (b + ch)x$ ⇒ $u_{FEM}(0) = a$, $u_{FEM}(h) = a + bh + ch^2$

* u_{exact} and u_{FEM} match at the nodes and the error is largest at the midpoint D

$$\text{error} = e = u_D - u_B = u_{FEM}(h/2) - u_{exact}(h/2)$$

$$= \frac{u_A + u_C}{2} - u_B = \frac{a + a + bh + ch^2}{2} - \left(a + \frac{bh}{2} + \frac{ch^2}{4}\right)$$

$$e = a + \frac{bh}{2} + \frac{ch^2}{2} - \left(a + \frac{bh}{2} + \frac{ch^2}{4}\right) \Rightarrow e = \frac{ch^2}{4}$$

note that $\frac{d^2u}{dx^2} = 2c$ or $c = \frac{u''}{2}$, $u'' = \frac{d^2u}{dx^2}$ ← 2nd derivative of displacement

... substitute this in for c :

$$\boxed{e = \frac{h^2}{8}u''}$$

→ ① error proportional to h^2
② error also proportional to $u'' = \frac{d^2u}{dx^2}$

error in strain : largest at A and C

so final error at $A = e'_A = \frac{u_c - u_A}{h} - b \Rightarrow e'_A = \frac{(a + bh + ch^2 - a)}{h} - b \Rightarrow \boxed{e'_A = ch = \frac{h}{2}u''}$

for x @ a

"e prime" = error in strain

because $\frac{du}{dx}(x=0) = b + 2cx$
$= b$

① error proportional to h
② error also proportional to $\frac{d^2u}{dx^2} = u''$

What is this all saying?

Displacement :

$u_{exact} = a_0 + a_1 x + a_2 x^2$ and remember $u_{FEM} = a_0 + a_1 x$ $+ a_2 x^2$ missing

FE error $\sim \frac{h^2}{8}u''$

Strain :

$\frac{du_{exact}}{dx} = a_1 + 2a_2 x$

FE error $\sim \frac{h}{2}u''$ and remember $\frac{du_{FEM}}{dx} = a_1$ $+ 2a_2 x$ missing

→ looking at these expressions, what can we conclude about the FE error as a result of the displacement field?

- error is proportional to 2nd derivative of u

⇒ FEM error is proportional to first higher order term not contained in FEM approx.
... which in this case is 2nd order quadratic term

→ now what about for strain?

- same as above

if exact was linear and FE soln linear, then error would be zero. the idea is that for example, if have cubic but use quadratic FE approx, then error proportional to the missing term

Let's say you halve element length, i.e. $h \to h/2$
or I change from 1 element of length h to 2 elements of length $h/2$.
How much does displacement error change?

remember: $e \sim \frac{h^2}{8}u''$

so then if $h \to h/2$ then $e \to e/4$ and we therefore quartered our error

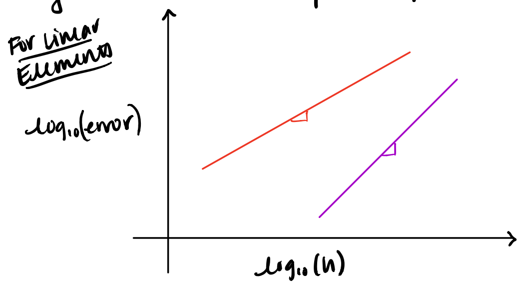
How much does strain error change?

remember: $e' \sim \frac{h}{2}u''$

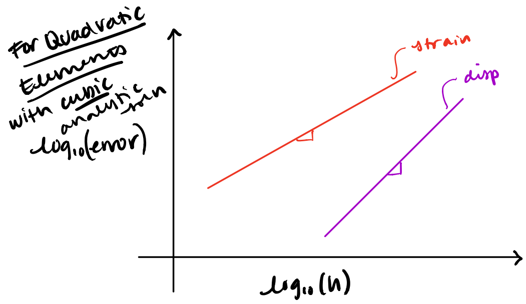
so then if $h \to h/2$ then $e' \to e/2$ and we therefore halve our error

For doubling of # of elements, error goes down by 75% !

Say error is $O(h^2)$ for displacement, and $O(h)$ for strain, where O means order.



① What is slope of each line?
1 bc order h
2 bc order h^2
② Which line is error in displacement and strain?
red = strain
purple = displacement



① What are the slopes now?
strain $\sim O(h^2)$
displacement $\sim O(h^3)$

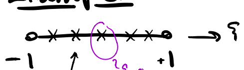
for linear elements if mesh $h \to h/2$, divide error by 4, strain error by 50%.

for quadratic, divide error by 8 instead of 4 and strain error by 75%.

How to calculate error in FEM?
→ analytical solutions are functions of physical space (x,y,z)
→ FEM solutions are functions of (x,y,z)
→ but, FEM shape functions known in isoparametric (ξ) space

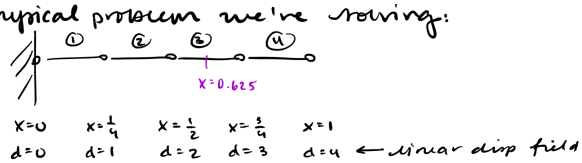
Procedure:
① map quadrature points from isoparametric (ξ) space to physical space (x)
② calculate exact solution at mapped quadrature point locations in x
③ interpolate FEM displacements from nodes to mapped quadrature point locations
④ calculate error at each mapped quadrature point in each element, sum to get element error, sum up all elements to get total error for entire structure

Example:



5 integration points to calculate the error

physical problem we're solving:



① map quadrature point $\xi = 0$ to element 3

$$x_a = \sum_{I=1}^{2 \leftarrow \#\text{ of nodes}} N_I(\xi) x_I = \begin{bmatrix} \frac{1-\xi}{2} & \frac{1+\xi}{2} \end{bmatrix} \begin{bmatrix} 0.5 \\ 0.75 \end{bmatrix}$$ ← x coord for element 3

← valid to map any ξ to element 3

for $\xi = 0 \Rightarrow \begin{bmatrix} 1/2 & 1/2 \end{bmatrix} \begin{bmatrix} 0.5 \\ 0.75 \end{bmatrix} = \frac{1}{2}(0.5) + \frac{1}{2}(0.75)$

tells us that point $\xi = 0$, maps to right in the middle of element 3 $\Rightarrow$ $x_a = 0.625$

choice of x determines the element mapping to

First element, $\xi = 0$ maps to $x_a = \frac{1}{8}$: $x_a = \begin{bmatrix} \frac{1-0}{2} & \frac{1+0}{2} \end{bmatrix} \begin{bmatrix} 0 \\ 0.25 \end{bmatrix} = \frac{1}{2}(0) + \frac{1}{2}(0.25) = \frac{1}{8}$

② Can calculate analytic solution at x_a

... let's say analytic solution (making this up) is : $u_{exact}(x) = \frac{1}{2}x^3$

so $u_{exact}(x_a) = \frac{1}{2} x_a^3$

③ Interpolate FEM nodal displacements to x_a

$$u_{FEM}(x_a) = \sum_{I=1}^{2} N_I(\xi) d_I$$

$$= \begin{bmatrix} \frac{1-\xi}{2} & \frac{1+\xi}{2} \end{bmatrix} \begin{bmatrix} 2 \\ 3 \end{bmatrix} = \begin{bmatrix} \frac{1-0}{2} & \frac{1+0}{2} \end{bmatrix} \begin{bmatrix} 2 \\ 3 \end{bmatrix} = \frac{1}{2}(2) + \frac{1}{2}(3) = 2.5$$

makes sense from before since 2.5 in between 2 and 3

④ Calculate error: (calculated as a norm)

$$\| u_{exact} - u_{FEM} \| = \left[\int_a^b | u_{exact} - u_{FEM} |^2 dx \right]^{1/2}$$

written slightly differently :

$$\left(\sum_{i=1}^{\#\text{ of elements}} \left[\sum_{j=1}^{\#\text{ of integration points}} | u_{exact}(x_j) - u_{FEM}(x_j) |^2 \frac{dx}{d\xi} w_j \right) ^{1/2}$$

x_j = mapped integration point location in x

loop over elements

loop over integration points

square so (+)

Jacobian

weight