What is FEM useful for?

1. heat xfer: conduction/convection
2. structural failure analysis
   - weakest member
   - stress analysis
3. deflection of structures
4. electromagnetism (FEM outside of mechanical engineering)
5. thermodynamics
6. fluid flow/chemical xport
7. vibrations

→ What do these all have in common?

- all are modelled by differential equations

Types of Differential Equations

1. fluids: Navier-Stokes equation
2. electromagnetism: Maxwells equations
3. heat xfer: diffusion equation

* 4. mechanics: equilibrium/momentum equation *

What is FEM?

→ approximate (numerical, not analytical) solution to differential equations

→ framework that allows continual, systematic improvement of solution

→ useful for when can't solve engineering problems analytically

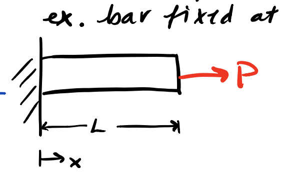
Why FEM?

1. little/no restrictions on geometry
2. allows all boundary/loading conditions
3. arbitrary material properties

DISCRETE vs. CONTINUOUS

→ continuous systems

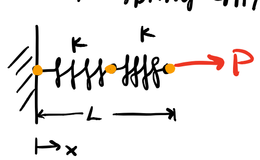
ex. bar fixed at one end, what is the deflection?



$\delta(x) = \dfrac{Px}{AE}$ → known for <u>any</u> value of x

→ discrete systems: solution for δ only known at certain, finite # of points ("nodes")

k = spring stiffness • - nodes discrete points where δ is known



*this is an approximation!!!

↳ this can be exact, but usually is not

Review of Matrix Notation

$$F_0 = \begin{bmatrix} F_{1x} \\ F_{1y} \\ F_{2x} \\ F_{2y} \end{bmatrix}$$ → vector, dimensions are 4 x 1

$$F_1 = \begin{bmatrix} F_{1x} & F_{1y} & F_{2x} & F_{2y} \end{bmatrix}$$ → vector, now is 1 x 4

$F_0^T = F_1$ (transpose)

matrix k:

$$k = \begin{bmatrix} k_{11} & k_{12} \\ k_{21} & k_{22} \end{bmatrix}$$ = matrix, 2x2

$$k^T = \begin{bmatrix} k_{11} & k_{21} \\ k_{12} & k_{22} \end{bmatrix}$$ → still 2x2

Systems of equations

$x + y = 2$
$x - y = 0$ $\Rightarrow \begin{bmatrix} 1 & 1 \\ 1 & -1 \end{bmatrix}\begin{bmatrix} x \\ y \end{bmatrix} = \begin{bmatrix} 2 \\ 0 \end{bmatrix}$

$Ax = b$

$x = A^{-1} b$

$$\overset{A}{\begin{bmatrix} 1 & 1 \\ 1 & -1 \end{bmatrix}}\overset{x}{\begin{bmatrix} x \\ y \end{bmatrix}} = \overset{b}{\begin{bmatrix} 2 \\ 0 \end{bmatrix}}$$

$$\underbrace{2 \times \underbrace{2 \quad 2}_{\text{same}} \times 1}_{\text{resultant}} \qquad 2 \times 1$$

THREE DIFFERENT WAYS TO GET THE FINITE ELEMENT EQUATION IN MECHANICS

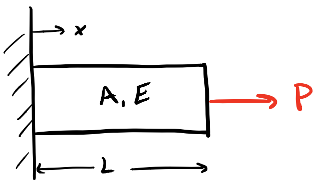
First Way : "Direct Stiffness Method"
Second Way : "Minimizing Potential Energy"
Third Way : ("Real Finite Elements") → derive differential equation, corresponding weak form

→ 3 different ways of getting FEM equations for solids

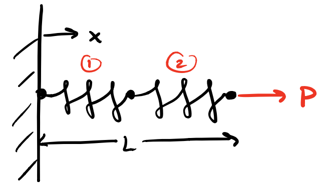
$$Kd = F^{ext}$$

K = stiffness
d = unknown nodal displacements

Continuous to discrete

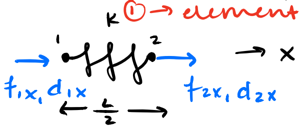


discrete approximation



goal: derive stiffness matrix k for 1D spring

1 → element #1 = spring plus 2 nodes

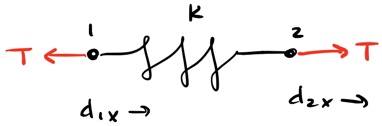


f_{1x} = force on node 1 $\quad$ d_{1x} = displacement of node 1

f_{2x} = force on node 2 $\quad$ d_{2x} = displacement of node 2

2 nodes ("degrees of freedom")
↳ where displacements of element are known
→ this element has 2 degrees of freedom: 2 nodes × 1 DOF/node
→ how many DOFs for each node in 2D?
  3 DOF: d_x, d_y, rotation

Step 1: select element type
→ easiest/cheapest: linear spring



T = tensile external force

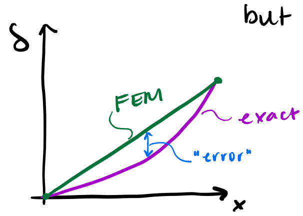
Step 2: select a displacement function
→ pick ahead of time mathematical function to represent the deformed shape
→ generally use low order polynomials (linear / quadratic)
→ 1D Bar: $\delta(x) = \frac{Px}{AE}$

↳ $\delta(x)$ is linear with respect to x

↳ now let's say we approximate with finite elements

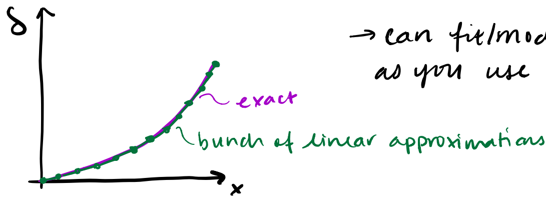
linear spring → displacement must be linear, so FEM is exact

How about if $\delta(x) = \frac{Px^2}{AE}$ (just making this up)



but FEM uses linear spring?
→ get exact answer at two points but everywhere else is off

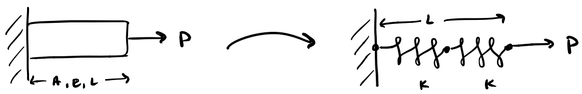
So why do linear approximations?
→ error if using 1 element (2 nodes)
→ what if use more elements?



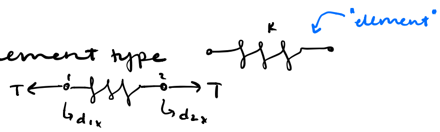
→ can fit/model nonlinear functions accurately as you use more and more linear elements!

Recap:

Idea: continuous to discrete transformation



Step 1: select an element type



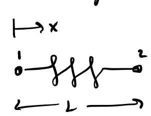
→ linear spring

Nodes: "degrees of freedom"
- where forces can be applied
- where discrete displacements occur

Step 2: select a displacement function
→ user defined → typically pick low order polynomial for the displacement field

say $u(x)$ = displacement
If $u(x)$ is linear: $u(x) = a_1 + a_2 x$ ⇒ in linear algebra form ⇒ $u(x) = \begin{bmatrix} 1 & x \end{bmatrix} \begin{bmatrix} a_1 \\ a_2 \end{bmatrix}$

1x2 2x1 = 1x1 so u(x) is scalar in 1D.



express u in terms of d_{1x} and d_{2x}

$u(x=0) = a_1 + a_2(0) = d_{1x}$

so $d_{1x} = a_1$

$u(x=L) = a_1 + a_2(L) = d_{2x}$

sub d_{1x} for a_1

so $d_{1x} + a_2(L) = d_{2x}$

solve for a_2:

$a_2 = \frac{d_{2x} - d_{1x}}{L}$ ⇒ now substitute back into expression for u. ⇒ $u(x) = a_1 + a_2 x$

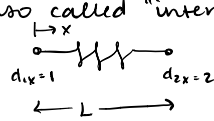
$u(x) = d_{1x} + \left(\frac{d_{2x} - d_{1x}}{L}\right) x$

now write in linear algebra form: $u(x) = \underbrace{\begin{bmatrix} 1 - \frac{x}{L} & \frac{x}{L} \end{bmatrix}}_{N(x)} \underbrace{\begin{bmatrix} d_{1x} \\ d_{2x} \end{bmatrix}}_{d}$

– or – $u(x) = \underbrace{N(x)}_{\text{"shape functions}} d$ ← unknown nodal displacements

N(x) == shape functions because express shape of assumed displacement field mathematically (linear in this case)
→ N(x) also called "interpolation" functions – how we get answers inbetween (but only works for linear displacement, not cubic/quadratic)

ex.



what is $u(x = \frac{L}{3})$?

$u(x=\frac{L}{3}) = \begin{bmatrix} 1 - \frac{L/3}{L} & \frac{L/3}{L} \end{bmatrix} \begin{bmatrix} d_{1x}=1 \\ d_{2x}=2 \end{bmatrix} = \left[\left(1 - \frac{L/3}{L}\right)(1) + \left(\frac{L/3}{L}\right)(2)\right] = \frac{2}{3} + \frac{2}{3} = \frac{4}{3}$

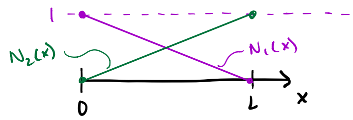
We can write shape functions like this:

$$N(x) = [N_1(x) \quad N_2(x)] = [1-\frac{x}{L} \quad \frac{x}{L}] \Rightarrow N_1(x) + N_2(x) = 1$$

What does it mean that shape functions are always equal to one?

- if functions don't add up to 1 then won't interpolate correctly
  - $N_1(x) + N_2(x) \neq 1$ means incorrect interpolation

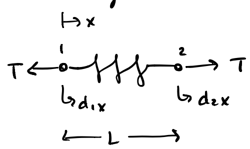
→ plot shape functions



... always add up to 1

(*) both shape functions are 1 at node, 0 at other node
→ this makes it easy to impose boundary conditions

Step 3: Define the element's material properties, i.e. Define strain-displacement or stress-strain relationship
→ allows you to model many different types of material behavior



change in element length: δ

$$\delta = u(x=L) - u(x=0) = d_{2x} - d_{1x}$$

because linear spring: $T = k\delta = k(d_{2x} - d_{1x})$

Step 4: Derive element stiffness matrix

$f_{1x} = -T$ ... in equilibrium, forces cancel out
$f_{2x} = +T$

$$f_{1x} = -T = k(d_{1x} - d_{2x})$$
$$f_{2x} = +T = k(d_{2x} - d_{1x})$$

matrix form:

$$\underbrace{\begin{bmatrix} f_{1x} \\ f_{2x} \end{bmatrix}}_{\text{external forces}} = \underbrace{\begin{bmatrix} k & -k \\ -k & k \end{bmatrix}\begin{bmatrix} d_{1x} \\ d_{2x} \end{bmatrix}}_{\text{internal forces}}$$

the amount structure deforms, d, is dependent on the stiffness, k

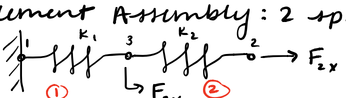
$$F^{ext} = F^{int} \rightarrow \boxed{kd = F^{ext}}$$

the k can be very different:
k is larger, displacements smaller (think silicone, steel, any stiff material)
while if soft material like rubber then k is smaller and displacements are greater

---

Now how do we extend what we've done to more than one element?

Element Assembly: 2 springs



note that elements 1 and 2 are connected at common node 3
→ called continuity or compatibility requirement
mathematically: $d_{3x}^{(1)} = d_{3x}^{(2)} = d_{3x}$

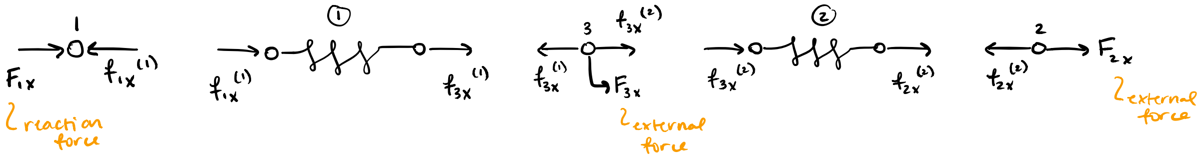
element

for element 1: $\begin{bmatrix} f_{1x}^{(1)} \\ f_{3x}^{(1)} \end{bmatrix} = \begin{bmatrix} k_1 & -k_1 \\ -k_1 & k_1 \end{bmatrix}\begin{bmatrix} d_{1x}^{(1)} \\ d_{3x}^{(1)} \end{bmatrix}$

node

for element 2: $\begin{bmatrix} f_{3x}^{(2)} \\ f_{2x}^{(2)} \end{bmatrix} = \begin{bmatrix} k_2 & -k_2 \\ -k_2 & k_2 \end{bmatrix}\begin{bmatrix} d_{3x}^{(2)} \\ d_{2x}^{(2)} \end{bmatrix}$

Free Body Diagrams (FBDs)



$F^{ext} = F^{int}$ at each node:

$F_{3x} = f_{3x}^{(1)} + f_{3x}^{(2)}$
$F_{2x} = f_{2x}^{(2)}$
$F_{1x} = f_{1x}^{(1)}$

... now substitute in equations we wrote for elements 1 and 2 ...

$$F_{3x} = f_{3x}^{(1)} + f_{3x}^{(2)} = (-k_1 d_{1x}^{(1)} + k_1 d_{3x}^{(1)}) + (k_2 d_{3x}^{(2)} - k_2 d_{2x}^{(2)})$$
$$F_{2x} = f_{2x}^{(2)} = -k_2 d_{3x}^{(2)} + k_2 d_{2x}^{(2)}$$
$$F_{1x} = f_{1x}^{(1)} = k_1 d_{1x}^{(1)} - k_1 d_{3x}^{(1)}$$

we write $F_{1x} = 0$ because it's not technically an applied load

bc rxn ≠ ext. force

F_2 & F_3 are loads applied to system, rxn force just generated because other forces applied

$$\underbrace{\begin{bmatrix} F_{1x} = 0 \\ F_{2x} \\ F_{3x} \end{bmatrix}}_{F^{ext}} = \underbrace{\begin{bmatrix} k_1 & 0 & -k_1 \\ 0 & k_2 & -k_2 \\ -k_1 & -k_2 & k_1 + k_2 \end{bmatrix}}_{k = 3\times 3} \underbrace{\begin{bmatrix} d_{1x} \\ d_{2x} \\ d_{3x} \end{bmatrix}}_{d = 3\times 1}$$

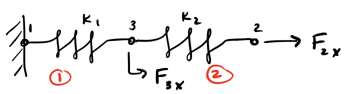
local → global
$d_{1x}^{(1)} = d_{1x}$
$d_{2x}^{(2)} = d_{2x}$
$d_{3x}^{(1)} = d_{3x}^{(2)} = d_{3x}$

observations about k:
- symmetric
- $k(3,3) = k_1 + k_2$ → represents that displacement of node 3 depends on both elements it is connected to

---

Assembly of Global Stiffness by Superposition

→ same 2 element assembly:



element stiffness for first element : $k^{(1)} = \begin{array}{cc} d_{1x} & d_{3x} \\ \begin{bmatrix} k_1 & -k_1 \\ -k_1 & k_1 \end{bmatrix} \end{array} \begin{array}{c} d_{1x} \\ d_{3x} \end{array}$

element stiffness for second element : $k^{(2)} = \begin{array}{cc} d_{3x} & d_{2x} \\ \begin{bmatrix} k_2 & -k_2 \\ -k_2 & k_2 \end{bmatrix} \end{array} \begin{array}{c} d_{3x} \\ d_{2x} \end{array}$

$$\begin{array}{ccc} d_{1x} & d_{2x} & d_{3x} \\ \begin{bmatrix} k_1 & 0 & -k_1 \\ 0 & 0 & 0 \\ -k_1 & 0 & k_1 \end{bmatrix} \end{array} \begin{bmatrix} d_{1x} \\ d_{2x} \\ d_{3x} \end{bmatrix} + \begin{array}{ccc} d_{1x} & d_{2x} & d_{3x} \\ \begin{bmatrix} 0 & 0 & 0 \\ 0 & k_2 & -k_2 \\ 0 & -k_2 & k_2 \end{bmatrix} \end{array} \begin{bmatrix} d_{1x} \\ d_{2x} \\ d_{3x} \end{bmatrix} = \begin{bmatrix} F_{1x} = 0 \\ F_{2x} \\ F_{3x} \end{bmatrix}$$

combine:

$$\begin{bmatrix} k_1 & 0 & -k_1 \\ 0 & k_2 & -k_2 \\ -k_1 & -k_2 & k_1 + k_2 \end{bmatrix} \begin{bmatrix} d_{1x} \\ d_{2x} \\ d_{3x} \end{bmatrix} = \begin{bmatrix} F_{1x} = 0 \\ F_{2x} \\ F_{3x} \end{bmatrix}$$

coupling we get at node 3 from elements 1 and 2

... same as what we discovered before, just faster