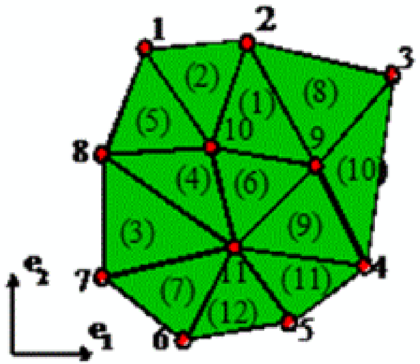
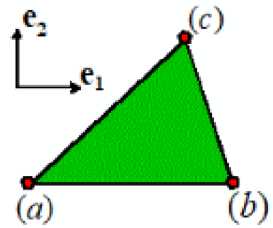
Review – Simple FEA for plane linear elasticity

- Approach: compute displacement field in an elastic solid by
  - Interpolating displacement field
  - Calculating total potential energy of solids in terms of discrete displacements
  - Minimize potential energy
- Interpolation – constant strain triangles



$$u_i(x_1,x_2) = u_i^{(a)} N_a(x_1,x_2) + u_i^{(b)} N_b(x_1,x_2) + u_i^{(c)} N_c(x_1,x_2)$$

$$N_a(x_1,x_2) = \frac{\left(x_2 - x_2^{(b)}\right)\left(x_1^{(c)} - x_1^{(b)}\right) - \left(x_1 - x_1^{(b)}\right)\left(x_2^{(c)} - x_2^{(b)}\right)}{\left(x_2^{(a)} - x_2^{(b)}\right)\left(x_1^{(c)} - x_1^{(b)}\right) - \left(x_1^{(a)} - x_1^{(b)}\right)\left(x_2^{(c)} - x_2^{(b)}\right)} \qquad \textbf{etc}$$



- Potential Energy

$$\Phi = \int_A \phi \, dA - \int_{S_2} \mathbf{t}^* \cdot \mathbf{u} \, ds$$

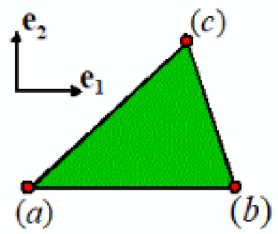
$$\Phi = \frac{1}{2}\left(\mathbf{u}^{Global}\right)^T [\mathbf{K}]\mathbf{u}^{Global} - \left(\mathbf{u}^{Global}\right)^T \cdot \mathbf{f}^{Global}$$

$$\mathbf{u}^{Global} = \begin{bmatrix} u_1^1 \\ u_2^1 \\ u_1^2 \\ u_2^2 \\ \vdots \\ u_1^N \\ u_2^N \end{bmatrix}$$

$$\Phi = \sum_{elements} \phi^{element} - \sum_{element\ faces} \int_{L_{face}} \mathbf{t}^* \cdot \mathbf{u} \, ds$$

Review – Strain energy density in an element

$$\underline{\varepsilon} = [B]\underline{u}^{\text{element}} \equiv \begin{bmatrix} \varepsilon_{11} \\ \varepsilon_{22} \\ 2\varepsilon_{12} \end{bmatrix} = \begin{bmatrix} \frac{\partial N_a}{\partial x_1} & 0 & \frac{\partial N_b}{\partial x_1} & 0 & \frac{\partial N_c}{\partial x_1} & 0 \\ 0 & \frac{\partial N_a}{\partial x_2} & 0 & \frac{\partial N_b}{\partial x_2} & 0 & \frac{\partial N_c}{\partial x_2} \\ \frac{\partial N_a}{\partial x_2} & \frac{\partial N_a}{\partial x_1} & \frac{\partial N_b}{\partial x_2} & \frac{\partial N_b}{\partial x_1} & \frac{\partial N_c}{\partial x_2} & \frac{\partial N_c}{\partial x_1} \end{bmatrix} \begin{bmatrix} u_1^{(a)} \\ u_2^{(a)} \\ u_1^{(b)} \\ u_2^{(b)} \\ u_1^{(c)} \\ u_2^{(c)} \end{bmatrix}$$



$$\begin{bmatrix} \sigma_{11} \\ \sigma_{22} \\ \sigma_{12} \end{bmatrix} = \frac{E}{(1+\nu)(1-2\nu)} \begin{bmatrix} 1-\nu & \nu & 0 \\ \nu & 1-\nu & 0 \\ 0 & 0 & \frac{1-2\nu}{2} \end{bmatrix} \begin{bmatrix} \varepsilon_{11} \\ \varepsilon_{22} \\ 2\varepsilon_{12} \end{bmatrix}$$

$$W^{\text{element}} = \frac{1}{2}\underline{u}^{\text{element}T}\left(A_{element}[B]^T[D][B]\right)\underline{u}^{\text{element}}$$

(Total Strain Energy)

$$W^{\text{element}} = \frac{1}{2}\underline{u}^{\text{element}T}K^{\text{element}}\underline{u}^{\text{element}}$$

Review – Total strain energy

$$W = \frac{1}{2}\begin{bmatrix} u_1^{(1)} \\ u_2^{(1)} \\ u_1^{(2)} \\ u_2^{(2)} \\ u_1^{(3)} \\ u_2^{(3)} \\ u_1^{(4)} \\ u_2^{(4)} \end{bmatrix}^T \begin{bmatrix} k_{11}^{(1)} & k_{12}^{(1)} & k_{13}^{(1)} & k_{14}^{(1)} & \cdots & \\ k_{21}^{(1)} & k_{22}^{(1)} & k_{32}^{(1)} & & & \\ & & k_{33}^{(1)}+k_{11}^{(2)} & k_{34}^{(1)}+k_{12}^{(2)} & & \\ & & k_{43}^{(1)}+k_{21}^{(2)} & k_{44}^{(1)}+k_{22}^{(2)} & & \\ & & k_{53}^{(1)}+k_{31}^{(2)} & & & \\ & & & & \ddots & \\ & & \vdots & & & k_{56}^{(2)} \\ & & & & k_{65}^{(2)} & k_{66}^{(2)} \end{bmatrix} \begin{bmatrix} u_1^{(1)} \\ u_2^{(1)} \\ u_1^{(2)} \\ u_2^{(2)} \\ u_1^{(3)} \\ u_2^{(3)} \\ u_1^{(4)} \\ u_2^{(4)} \end{bmatrix}$$

$$W = \frac{1}{2}\underline{u}^T [K] \underline{u}$$

Review – Boundary loading

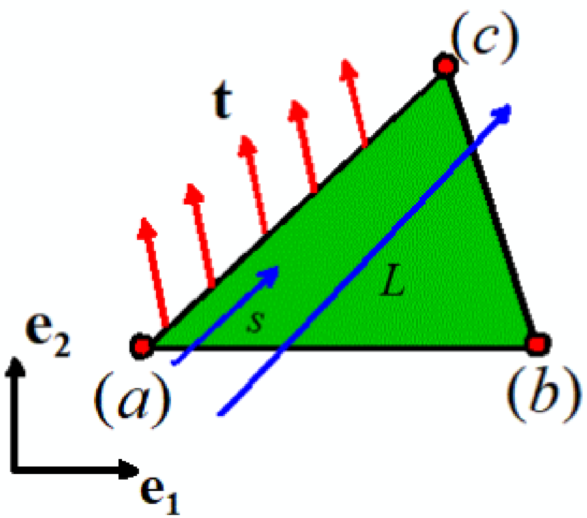
$$\rho^{element} = -\int_0^L t_i u_i ds$$

$$\rho^{\text{element}} = -t_i u_i^{(c)} \int_0^L \frac{s}{L} ds - t_i u_i^{(a)} \int_0^L \left(1 - \frac{s}{L}\right) ds$$

$$= -t_i u_i^{(a)} \frac{L}{2} - t_i u_i^{(c)} \frac{L}{2}$$

$$= -\left[t_1 \frac{L}{2} \quad t_2 \frac{L}{2} \quad t_1 \frac{L}{2} \quad t_2 \frac{L}{2} \right] \cdot \left[u_1^{(a)} \quad u_2^{(a)} \quad u_1^{(c)} \quad u_2^{(c)} \right]$$

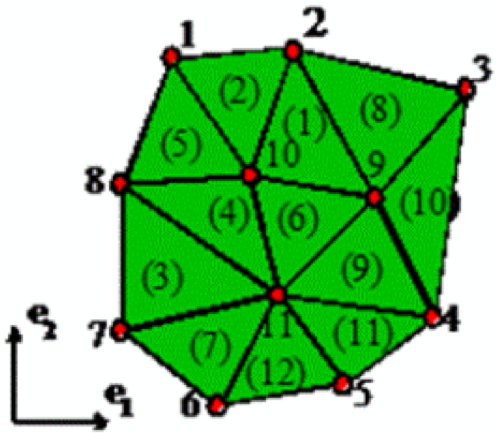
$$= \underline{u}^{elT} \underline{r}^{el}$$



Review – Minimizing PE

$$V = \frac{1}{2}\underline{u}^T\left[K\right]\underline{u} - \underline{r}\cdot\underline{u}$$

$$\equiv \frac{1}{2}\sum_{j=1}^{2N} u_j \sum_{i=1}^{2N} K_{ji}u_i - \sum_{j=1}^{2N} r_j u_j$$



$$\frac{\partial V}{\partial u_k} = \frac{1}{2}\sum_{i=1}^{2N} K_{ki}u_i + \frac{1}{2}\sum_{j=1}^{2N} u_j K_{kj} - r_k$$

$$= \sum_{i=1}^{2N} K_{ki}u_i - r_k = 0$$

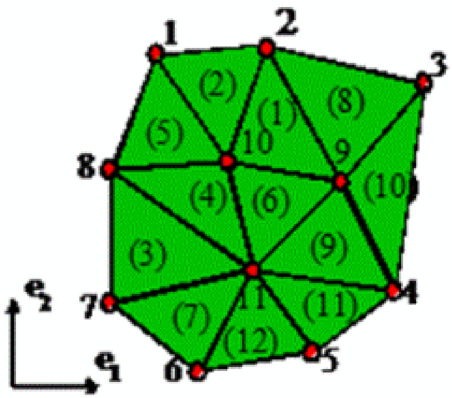
$$\Rightarrow \quad \left[K\right]\underline{u} = \underline{r}$$

Review – Constraining DOF

Modify equation system to impose known values of displacement

Original equations:

$$\begin{bmatrix} k_{11} & k_{12} & \cdots & k_{12N} \\ k_{21} & k_{22} & & k_{22N} \\ \vdots & & \ddots & \\ k_{2N1} & k_{2N2} & & k_{2N2N} \end{bmatrix} \begin{bmatrix} u_1^{(1)} \\ u_2^{(1)} \\ \vdots \\ u_2^{(N)} \end{bmatrix} = \begin{bmatrix} r_1 \\ r_2 \\ \vdots \\ r_4 \end{bmatrix}$$

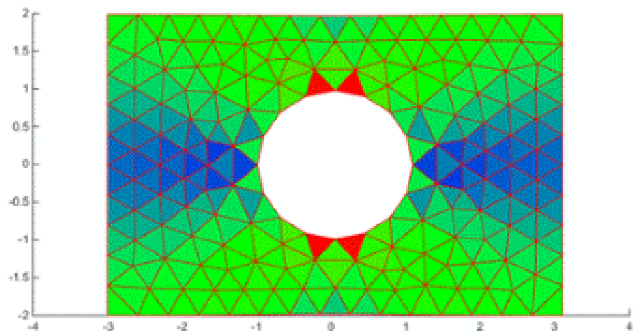


To constrain 2nd displacement component at first node:

$$\begin{bmatrix} k_{11} & k_{12} & \cdots & k_{1\,2N} \\ 0 & 1 & & 0 \\ \vdots & & \ddots & \\ k_{2N\,1} & k_{2N\,2} & & k_{2N\,2N} \end{bmatrix} \begin{bmatrix} u_1^{(1)} \\ u_2^{(1)} \\ \vdots \\ u_2^{(N)} \end{bmatrix} = \begin{bmatrix} r_1 \\ \Delta \\ \vdots \\ r_4 \end{bmatrix}$$

Optional: can symmetrize stiffness

Review – Structure of a basic FEA code



- Read data defining problem:
  - Material properties
  - Nodal coordinates
  - Element connectivity
  - List of nodes with prescribed DOF
  - List of elements with loaded faces
- Loop over elements
  - Compute element stiffness, add to global stiffness
- Loop over elements with loaded faces
  - Compute element force vector, add to global force vector
- Modify stiffness and RHS to impose prescribed disps.
- Solve FEA equations for unknown nodal displacements
- Post-processing – compute element strains & stresses

pag

Topics for todays class

- Observations from example FEA Matlab code
- Brief look at HW2
- Generalizing FEA procedures for elasticity
  - Deriving the FE equations from Virtual Work
  - 2D and 3D isoparametric interpolation
  - Calculating the element stiffness by numerical integration

Observations on structure of stiffness matrix

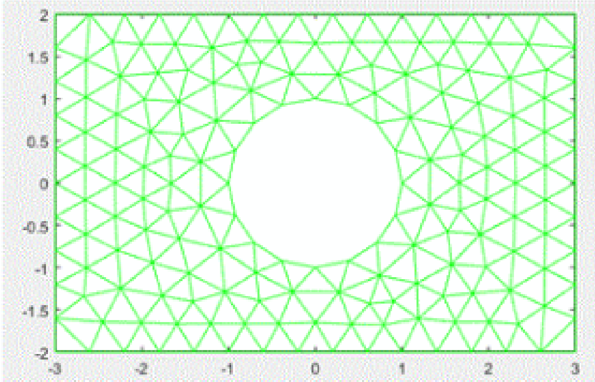
Note that much of stiffness is zero

Solve system with a sparse solver

Storage schemes:
(a) Banded or skyline
(b) Indexed storage

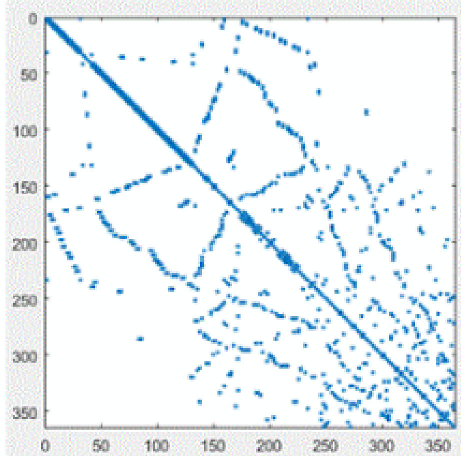
We often re-number nodes to minimize bandwidth

Example Mesh



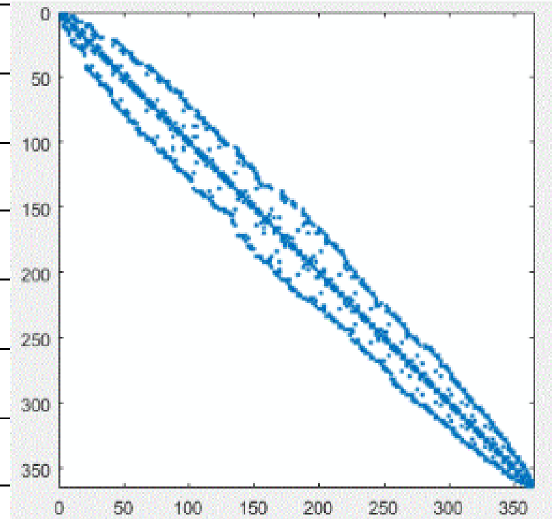
Stiffness Profile

Row #

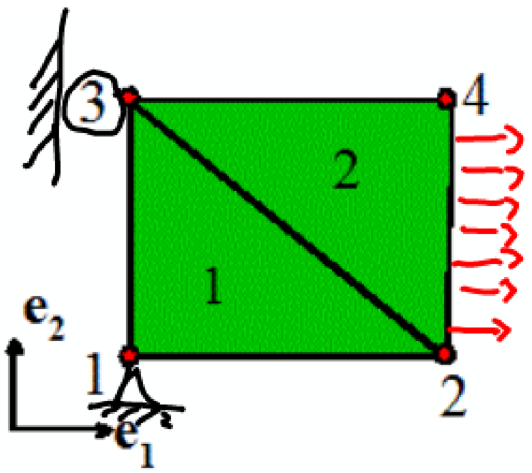


Col #

Stiffness after renumbering



Effect of constraints on stiffness



Global stiffness : Before constraint

$$[K] = \begin{bmatrix}
86.5385 & 48.0769 & -67.3077 & -19.2308 & -19.2308 & -28.8462 & 0 & 0 \\
48.0769 & 86.5385 & -28.8462 & -19.2308 & -19.2308 & -67.3077 & 0 & 0 \\
-67.3077 & -28.8462 & 86.5385 & 0 & 0 & 48.0769 & -19.2308 & -19.2308 \\
-19.2308 & -19.2308 & 0 & 86.5385 & 48.0769 & 0 & -28.8462 & -67.3077 \\
-19.2308 & -19.2308 & 0 & 48.0769 & 86.5385 & 0 & -67.3077 & -28.8462 \\
-28.8462 & -67.3077 & 48.0769 & 0 & 0 & 86.5385 & -19.2308 & -19.2308 \\
0 & 0 & -19.2308 & -28.8462 & -67.3077 & -19.2308 & 86.5385 & 48.0769 \\
0 & 0 & -19.2308 & -67.3077 & -28.8462 & -19.2308 & 48.0769 & 86.5385
\end{bmatrix}$$

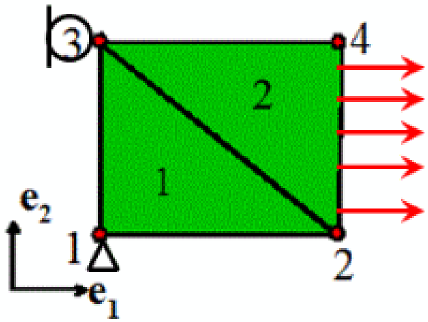
After imposing constraints

$$\begin{bmatrix}
1.0000 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & 1.0000 & 0 & 0 & 0 & 0 & 0 & 0 \\
-67.3077 & -28.8462 & 86.5385 & 0 & 0 & 48.0769 & -19.2308 & -19.2308 \\
-19.2308 & -19.2308 & 0 & 86.5385 & 48.0769 & 0 & -28.8462 & -67.3077 \\
0 & 0 & 0 & 0 & 1.0000 & 0 & 0 & 0 \\
-28.8462 & -67.3077 & 48.0769 & 0 & 0 & 86.5385 & -19.2308 & -19.2308 \\
0 & 0 & -19.2308 & -28.8462 & -67.3077 & -19.2308 & 86.5385 & 48.0769 \\
0 & 0 & -19.2308 & -67.3077 & -28.8462 & -19.2308 & 48.0769 & 86.5385
\end{bmatrix}$$

Improper constraints lead to singular stiffness matrix

eigenvecs =

0	0	-0.3152	-0.4728	0	-0.3152	0.4781	0.5908
0	0	0.6043	-0.4559	0	0.6043	0.0433	0.2449
0	0	0.0866	0.6137	0	0.0866	-0.2079	0.7518
0	0	-0.7071	-0.0000	0	0.7071	-0.0000	0.0000
0	0	-0.1673	-0.4381	0	-0.1673	-0.8522	0.1606
0.4779	0	0.4907	0.5123	0	0.0000	-0.0080	0.5179
0	0.4779	0.0000	0.5123	0	0.4907	-0.0080	0.5179
0	0	-0.0039	-0.5103	0.4759	-0.0039	0.4928	-0.5198



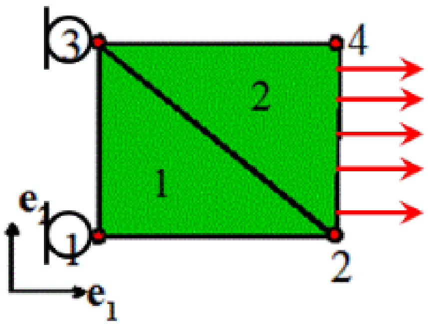
eigenvals =

199.8263	0	0	0	0	0	0	0
0	125.4412	0	0	0	0	0	0
0	0	13.8635	0	0	0	0	0
0	0	0	38.4615	0	0	0	0
0	0	0	0	55.0998	0	0	0
0	0	0	0	0	1.0000	0	0
0	0	0	0	0	0	1.0000	0
0	0	0	0	0	0	0	1.0000

eigenvecs =

0	0.3791	-0.3791	-0.3791	0	-0.4610	0.3791	0.4610
0	-0.4523	0.3015	-0.4523	0	0.4523	0.3015	0.4523
0	-0.5000	-0.5000	0.5000	0	-0.0000	0.5000	0.0000
0	-0.2132	-0.6396	-0.2132	0	0.2132	-0.6396	0.2132
0	-0.3260	0.3260	0.3260	0	-0.5361	-0.3260	0.5361
0	0.5000	0.0000	0.5000	0	0.5000	0.0000	0.5000
0.5638	-0.2906	0.5789	0.2929	0	-0.2984	-0.0046	0.2961
0	0.2929	-0.0046	-0.2906	0.5638	0.2961	0.5789	-0.2984

Eigenvector for zero eigenvalue



eigenvals =

216.4663	0	0	0	0	0	0	0
0	153.8462	0	0	0	0	0	0
0	0	76.9231	0	0	0	0	0
0	0	0	48.0769	0	0	0	0
0	0	0	0	23.9183	0	0	0
0	0	0	0	0	-0.0000	0	0
0	0	0	0	0	0	1.0000	0
0	0	0	0	0	0	0	1.0000

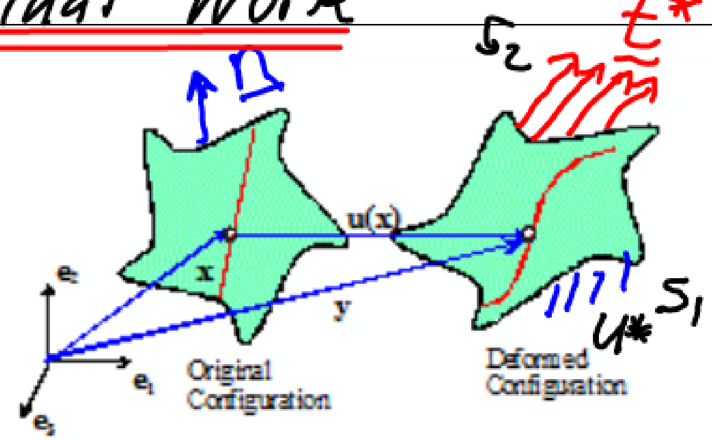
zero eigenvalue

> In fem_conststrain_triangles (line 93)
Warning: Matrix is close to singular or badly scaled. Results may be inaccurate. RCOND = 2.771117e-17.

4 Generalizing FEA for linear elasticity

Goals: Find a more general way to get FEA than energy minimization
Extend elasticity to 3D
Use better interpolations

4.1 Deriving FEA from principle of virtual work



PVW: Let σ_{ij} be any symmetric tensor field on R

Let δu_i be a "virtual displacement" satisfying
$\delta u_i = 0$ on S_1

Let $\delta \varepsilon_{ij} = \frac{1}{2}\left(\frac{\partial \delta u_i}{\partial x_j} + \frac{\partial \delta u_j}{\partial x_i}\right)$

Suppose that

$$\int_R \sigma_{ij}\,\delta\varepsilon_{ij}\,dV - \int_{S_2} t_i^*\,\delta u_i\,dA = 0 \quad \forall \text{ admiss } \delta u_i$$

Then $\underbrace{\dfrac{\partial \sigma_{ij}}{\partial x_i} = 0}$ equilibrium $\qquad \sigma_{ij} n_i = t_j^*$ Boundary Condition

PVW = "weak form" of equilibrium

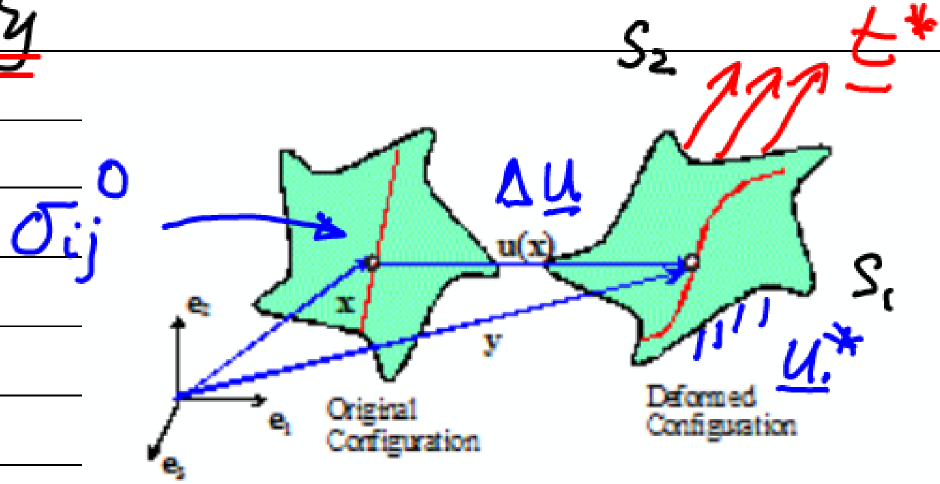
Use this to generate FEA approximations

- Interpolate u_i and δu_i between discrete vals
- Satisfy PVW for all admiss $\delta u_i \rightarrow u_i$ should give an equilibrium stress field

Apply PVW to linear elasticity

Suppose $\sigma_{ij} = \sigma_{ij}^{0}$ @ $t=0$

$$\sigma_{ij} = \sigma_{ij}^{0} + C_{ijkl} \frac{\partial \Delta u_k}{\partial x_l}$$



Introduce interpolation :

$$\Delta \underline{u} = N^a(\underline{x})\, \underline{u}^a \qquad \text{Sum on } a$$

$$\delta \underline{u} = N^a(\underline{x})\, \delta \underline{u}^a$$

Substitute into PVW:

$$\int_{R} C_{ijkl} \frac{\partial N^b}{\partial x_l} \Delta U_k^b \frac{\partial N^a}{\partial x_j} \delta U_i^a \, dV + \int_{R} \sigma_{ij}^0 \frac{\partial N^a}{\partial x_j} \delta U_i^a \, dV - \int_{S_2} t_i^* N^a \delta U_i^a \, dA$$

$$= 0 \quad \forall \text{ admiss } \delta U_i^a$$

Admiss δU_i^a must satisfy δU_i^a for a on S_1

Interpolation must satisfy

$$N^a(\underline{x}^b) = 1 \quad \text{if } a = b$$

$$\sum_a N^a = 1$$

We usually choose $N^a = 0$ if $\underline{x}$ is outside element with node a on it

. Rewrite PVW in matrix form

$$\boxed{\left\{ K_{aibk}\, \Delta U_k^b + R_{ai} - f_{ai} \right\} \delta u_i^a = 0 \quad \forall \text{ admiss } \delta u_i^a}$$

$$K_{aibk} = \int_R C_{ijkl} \frac{\partial N^b}{\partial x_l} \frac{\partial N^a}{\partial x_j}\, dV$$

$$R_{ai} = \int_R \sigma_{ij}^{o} \frac{\partial N^a}{\partial x_j}\, dV \qquad f_{ai} = \int_{S_2} t_i^* N^a\, dA$$

K_{aibk} ← DOF

Node # → a, ↑ DOF (i), Node ← b

We store K_{aibk} as a $3n \times 3n$ (3D) / $2n \times 2n$ (2D) matrix

where $n = \#$ nodes

$$\begin{bmatrix} K_{1111} & K_{1112} & K_{1113} & K_{1121} & \cdots \\ K_{1211} & & & & \\ K_{1311} & & & & \\ K_{2111} & & & & \end{bmatrix}$$

Similarly u_i^a, R_{ai} f_{ai} etc are $3n$ long vecs

eg: $u_i^a = [\, u_1^1 \;\; u_2^1 \;\; u_3^1 \;\; \cdots \,]$ etc