# Plane Strain Elastostatics: Computer Implementation:

Now that we have discussed finite element approximation of the plane strain elastostatics, we turn again to a discussion of computer implementation. As we did for steady two-dimensional heat conduction, we form and assemble element and boundary element stiffness matrices and force vectors.

To begin, note we can write stiffness matrix and force vector entries in terms of integrals over elements and boundary edges:

$$K_{PQ} = \sum_{\Omega^e \in \mathcal{M}^h} \int_{\Omega^e} \underset{\sim}{\varepsilon}\left(N_A \vec{e}_i\right)^T \underset{\approx}{D}\, \underset{\sim}{\varepsilon}\left(N_B \vec{e}_j\right) d\Omega$$

$$F_P = \sum_{\Omega^e \in \mathcal{M}^h} \int_{\Omega^e} N_A f_i \, d\Omega - \sum_{\Omega^e \in \mathcal{M}^h} \int_{\Omega^e} \underset{\sim}{\varepsilon}\left(N_A \vec{e}_i\right)^T \underset{\approx}{D}\, \underset{\sim}{\varepsilon}\left(\vec{g}^h\right) d\Omega$$

$$+ \sum_{\Gamma^e \in \mathcal{E}^h_{N_i}} \int_{\Gamma^e} N_A h_i \, d\Gamma$$

where $P = ID(A,i)$ and $Q = ID(B,j)$. Now define for each interior element an element stiffness matrix with entries:

$$k^e_{pq} := \int_{\Omega^e} \underset{\sim}{\varepsilon}\left(N^e_a \vec{e}_i\right)^T \underset{\approx}{D}\, \underset{\sim}{\varepsilon}\left(N^e_b \vec{e}_j\right) d\Omega$$

$$p = 2(a-1)+i \quad \text{and} \quad q = 2(b-1)+j$$

and an element force vector with entries:

$$f^e_p := \int_{\Omega^e} N^e_a \, f_i \, d\Omega \qquad p = 2(a-1)+i$$

and define for each boundary edge a boundary element force vector with entries:

$$f^{e,b}_p := \int_{\Gamma^e \cap \Gamma^h_{N_i}} N^{e,b}_a \, h_i \, d\Omega \qquad p = 2(a-1)+i$$

There is also a boundary element stiffness matrix for each boundary edge but it is identically zero. Then we have:

$$K_{PQ} = \sum_{e=1}^{nel} \sum_{i=1}^{2} \underbrace{\sum_{a=1}^{n_{en}}}_{P = ID(IEN(a,e),i)} \sum_{j=1}^{2} \underbrace{\sum_{b=1}^{n_{en}}}_{Q = ID(IEN(b,e),j)} k^e_{\underset{2(a-1)+i}{p} \, q \to 2(b-1)+j}$$

and:

$$F_P = \sum_{e=1}^{nel} \sum_{i=1}^{2} \underbrace{\sum_{a=1}^{n_{en}}}_{P = ID(IEN(a,e),i)} \Big( f^e_{\underset{2(a-1)+i}{p}} - \sum_{j=1}^{2} \underbrace{\sum_{b=1}^{n_{en}}}_{Q = ID(IEN(b,e),j)} k^e_{\underset{2(a-1)+i}{p} \, \underset{2(b-1)+j}{q}} \, g_j(\vec{x}_{\underset{IEN(b,e)}{B}}) \Big)$$

$$+ \sum_{e=1}^{nelb} \sum_{i=1}^{2} \underbrace{\sum_{a=1}^{n_{enb}}}_{P = ID(IENB(a,e),i)} f^{e,b}_{\underset{2(a-1)+i}{p}}$$

In pseudocode:

## Assembly Pseudocode

Set $\underset{=}{K} = \underset{=}{0}$ and $\underset{-}{F} = \underset{-}{0}$

```
for e=1:n_el
    for i=1:2
        for a=1:n_en
            p = 2(a-1)+i
            A = IEN(a,e)
            P = ID(A,i)
            if P ≠ 0
                for j=1:2
                    for b=1:n_en
                        q = 2(b-1)+j
                        B = IEN(b,e)
                        Q = ID(B,j)
                        if Q ≠ 0
                            K_PQ = K_PQ + k^e_pq
                        else
                            F_P = F_P - k^e_pq g_j(x⃗_B)
                        endif
                    endfor
                endfor
                F_P = F_P + f^e_p
```

```
            endif
         endfor
      endfor
   endfor
   for e=1:n_elb
      for i=1:2
         for a=1:n_enb
            p = 2(a-1)+i
            A = IENB(a,e)
            P = ID(A,i)
            if P ≠ 0
               F_P = F_P + f_p^{e,b}
            endif
         endfor
      endfor
   endfor
```

To compute the entries of the element and boundary element stiffness matrices and force vectors, we use the same two tricks as we have before:

<u>Trick #1</u>: We pull integrals back to the parent element.

<u>Trick #2</u>: We use quadrature to approximate parent element integrals.

Application of these tricks yields the approximations:

$$k^e_{pq} \approx \sum_{l=1}^{n_q} \left( \left( \underline{\underline{\varepsilon}} \left( N^e_a \vec{e}_i \right) \right) \left( \vec{x}^e ( \check{\vec{\xi}}_l ) \right) \right)^T \left( \underline{\underline{D}} \left( \vec{x}^e ( \check{\vec{\xi}}_l ) \right) \left( \left( \underline{\underline{\varepsilon}} \left( N^e_b \vec{e}_j \right) \right) \left( \vec{x}^e ( \check{\vec{\xi}}_l ) \right) \right) j^e ( \check{\vec{\xi}}_l ) \check{w}_l$$

for $p = 2(a-1)+i$ and $q = 2(b-1)+j$

$$f^e_p \approx \sum_{l=1}^{n_q} N^e_a \left( \vec{x}^e ( \check{\vec{\xi}}_l ) \right) f_i \left( \vec{x}^e ( \check{\vec{\xi}}_l ) \right) j^e ( \check{\vec{\xi}}_l ) \check{w}_l$$

for $p = 2(a-1)+i$

$$f^{e,b}_p \approx \begin{cases} \displaystyle\sum_{l=1}^{n_{qb}} N^{e,b}_a \left( \vec{x}^{e,b} ( \check{\vec{\xi}}^b_l ) \right) h_i \left( \vec{x}^{e,b} ( \check{\vec{\xi}}^b_l ) \right) j^e_\Gamma ( \check{\vec{\xi}}^b_l ) \check{w}^b_l & \text{if } \Gamma^e \in \mathcal{E}^h_{N_i} \\ 0 & \text{otherwise} \end{cases}$$

for $p = 2(a-1)+i$

Where just as for steady two-dimensional heat conduction we have:

$$N^e_a \left( \vec{x}^e ( \vec{\xi} ) \right) = \hat{N}_a ( \vec{\xi} )$$

$$\frac{\partial N^e_a}{\partial x_1} \left( \vec{x}^e ( \vec{\xi} ) \right) = \frac{\partial \hat{N}_a}{\partial \xi_1} ( \vec{\xi} ) \frac{\partial \xi^e_1}{\partial x_1} \left( \vec{x}^e ( \vec{\xi} ) \right) + \frac{\partial \hat{N}_a}{\partial \xi_2} ( \vec{\xi} ) \frac{\partial \xi^e_2}{\partial x_1} \left( \vec{x}^e ( \vec{\xi} ) \right)$$

$$\frac{\partial N^e_a}{\partial x_2} \left( \vec{x}^e ( \vec{\xi} ) \right) = \frac{\partial \hat{N}_a}{\partial \xi_1} ( \vec{\xi} ) \frac{\partial \xi^e_1}{\partial x_2} \left( \vec{x}^e ( \vec{\xi} ) \right) + \frac{\partial \hat{N}_a}{\partial \xi_2} ( \vec{\xi} ) \frac{\partial \xi^e_2}{\partial x_2} \left( \vec{x}^e ( \vec{\xi} ) \right)$$

for each element $\Omega^e \in \mathcal{M}^h$ and:

$$N_a^{e,b}\left(\vec{x}^{e,b}(\xi^b)\right) = \hat{N}_a^b(\xi^b)$$

for each boundary edge $\Gamma^e \in \mathcal{E}^h_{\Gamma}$. Note also that:

$$\underline{\underline{\varepsilon}}(N_a^e \vec{e}_i) = \underline{\underline{B}}_a^e \vec{e}_i \quad \text{and} \quad \underline{\underline{\varepsilon}}(N_b^e \vec{e}_j) = \underline{\underline{B}}_b^e \vec{e}_j$$

where:

$$\underline{\underline{B}}_a^e = \begin{bmatrix} \frac{\partial N_a^e}{\partial x} & 0 \\ 0 & \frac{\partial N_a^e}{\partial y} \\ \frac{\partial N_a^e}{\partial y} & \frac{\partial N_a^e}{\partial x} \end{bmatrix} \qquad \underline{\underline{B}}_b^e = \begin{bmatrix} \frac{\partial N_b^e}{\partial x} & 0 \\ 0 & \frac{\partial N_b^e}{\partial y} \\ \frac{\partial N_b^e}{\partial y} & \frac{\partial N_b^e}{\partial x} \end{bmatrix}$$

so we can equivalently write:

$$k^e_{pq} \approx \sum_{l=1}^{n_q} \left( \left( \underline{\underline{B}}_a^e(\vec{x}^e(\tilde{\xi}_l))\, \vec{e}_i \right)^T \left( \underline{\underline{D}}(\vec{x}^e(\tilde{\xi}_l)) \right) \left( \underline{\underline{B}}_b^e(\vec{x}^e(\tilde{\xi}_l))\, \vec{e}_j \right) j^e(\tilde{\xi}_l)\, \tilde{w}_l$$

for $p = 2(a-1)+i$ and $q = 2(b-1)+j$

Pseudocode for computing the entries of the element and boundary element stiffness matrices and force vectors using the above approximations is included below.

**Element Formation Pseudocode**

Set $\underline{\underline{k}}^e = \underline{\underline{0}}$ and $\underline{f}^e = \underline{0}$

```
for l=1:n_q
    for i=1:2
        for a=1:nen
            p = 2(a-1)+i
            for j=1:2
                for b=1:nen
                    q = 2(b-1)+j
```

$$k^e_{pq} = k^e_{pq} + \left( \left( \underline{\underline{B}}^e_a \left(\vec{x}^e(\hat{\xi}_l)\right) \vec{e}_i \right)^T \left( \underline{\underline{D}} \left(\vec{x}^e(\hat{\xi}_l)\right) \right) \left( \underline{\underline{B}}^e_b \left(\vec{x}^e(\hat{\xi}_l)\right) \vec{e}_j \right) j^e(\hat{\xi}_l) \hat{w}_l \right.$$

```
                endfor
            endfor
```

$$f^e_p = f^e_p + N^e_a\left(\vec{x}^e(\hat{\xi}_l)\right) f_i\left(\vec{x}^e(\hat{\xi}_l)\right) j^e(\hat{\xi}_l) \hat{w}_l$$

```
        endfor
    endfor
endfor
```

## Boundary Element Formation Pseudocode

Set $\underline{f}^{e,b} = \underline{0}$

for $\ell = 1 : n_{qb}$

  for $i = 1:2$

    if $\Gamma^e \in \mathcal{E}^h_{N_i}$

      for $a = 1 : n_{enb}$

        $p = 2(a-1) + i$

        $f_p^{e,b} = f_p^{e,b} + N_a^{e,b}\left(\vec{x}^{e,b}(\tilde{\xi}_\ell^b)\right) h_i\left(\vec{x}^{e,b}(\tilde{\xi}_\ell^b)\right) j_\Gamma^e(\tilde{\xi}_\ell^b)\, \tilde{w}_\ell^b$

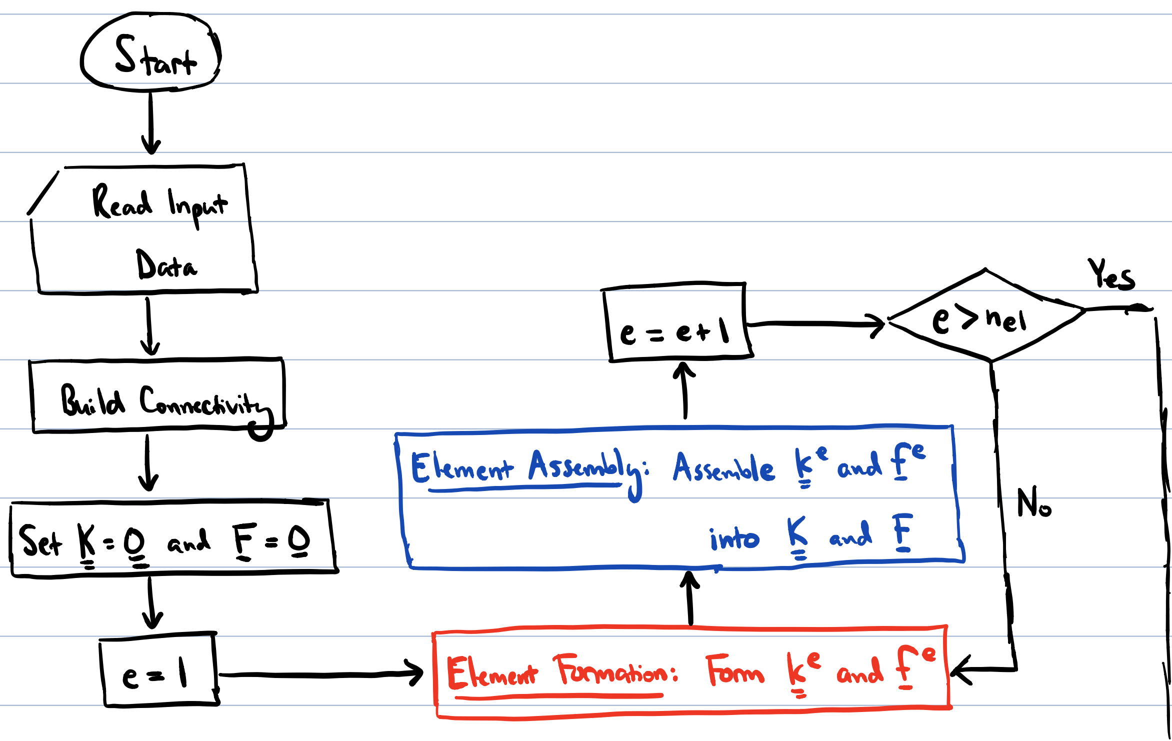
      endfor

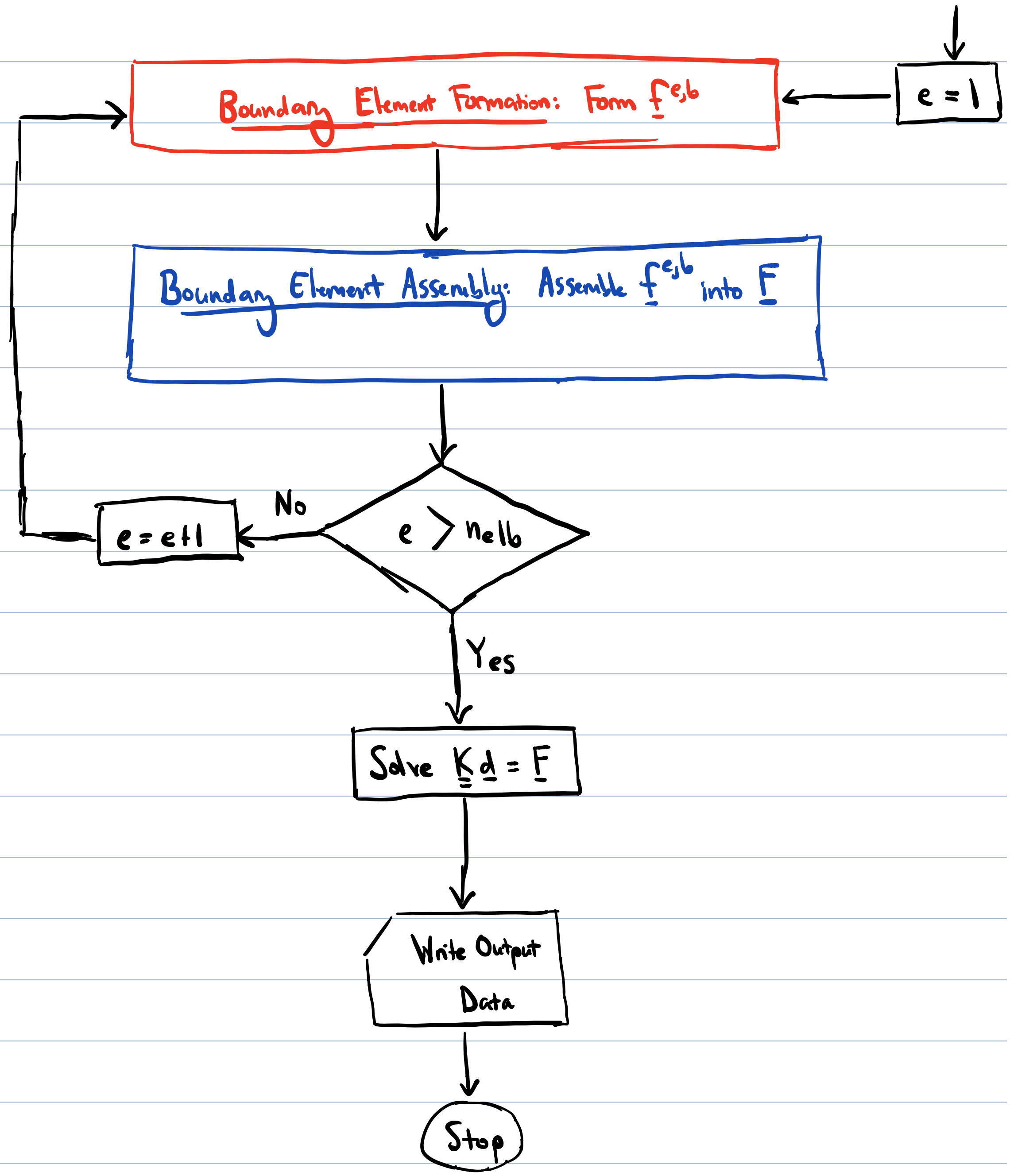
    endif

  endfor

endfor

With all of the above in hand, a finite element code for plane strain elastostatics has the following control flow:






Once the displacement field is known, one can find derived quantities of interest such as the stress tensor. Post processing for the stress tensor is called stress recovery. Many approaches to stress recovery have been proposed, and in the class code, stress is projected into the finite element approximation space.