Linear Elasticity: Assembling the Matrix System

Just as in the setting of heat conduction, we build the global stiffness matrix and forcing vector by looping through the elements and constructing dense local stiffness matrices and forcing vectors, which are then assembled into the global system. With d spatial dimensions (where $d = 2$ is the setting of interest) and n_{loc} local basis functions, we calculate the local stiffness matrix on element Ω^e as:

$$k^e_{pq} = \int_{\Omega^e} \vec{\varepsilon}(N^e_a \hat{e}_A)^T \vec{\vec{D}}\, \vec{\varepsilon}(N^e_b \hat{e}_B)\, d\Omega^e$$

where:

$$p = d(a-1) + A$$
$$q = d(b-1) + B$$

Noting that:

$$\vec{\varepsilon}(N^e_a \hat{e}_A) = \vec{\vec{B}}^e_a \hat{e}_A$$

where:

$$\vec{\vec{B}}^e_a = \begin{bmatrix} N^e_{a,1} & 0 \\ 0 & N^e_{a,2} \\ N^e_{a,2} & N^e_{a,1} \end{bmatrix} \quad \text{for } d = 2$$

we have the simplified form:

$$k^e_{pq} = \hat{e}_A^T \int_{\Omega^e} (\vec{\vec{B}}^e_a)^T \vec{\vec{D}} (\vec{\vec{B}}^e_b)\, d\Omega^e\, \hat{e}_B$$

Similarly, the entries in the element forcing vector are given by:

$$f^e_p = \int_{\Omega^e} N^e_a f_A\, d\Omega^e + \int_{\Gamma_{N_A} \cap \Gamma^e} N^e_a h_A\, d\Gamma^e$$

To assemble the local stiffness matrices and forcing vectors into the global system, it helps to employ the following arrays:

$$\underbrace{P}_{\text{Global Equation Number}} = LM(\underbrace{A}_{\text{DOF Number}}, \underbrace{a}_{\text{Local Basis Index}}, \underbrace{e}_{\text{Element}}) = ID(A, IEN(a,e))$$

$$BC(\underbrace{A}_{\text{DOF Number}}, \underbrace{i}_{\text{Global Basis Index}}) = \begin{cases} 0 & \text{if } N_i|_{\Gamma_{D_A}} = 0 \\ 1 & \text{otherwise} \end{cases}$$

If $BC(A,i) = 1$, we set:

$$K_{PQ} = \delta_{PQ}$$

$$F_P = (\vec{g}_i)_A$$

where $P = ID(A,i)$. This corresponds to the Dirichlet condition $(\vec{d}_i)_A = (\vec{g}_i)_A$.

As in the setting of heat conduction, if during the assembly process we come across i, j, A, B such that $BC(A,i) = 0$ and $BC(B,j) = 1$, then we do not update K_{PQ} with k^e_{pq} but rather F_P to account for the Dirichlet condition where $P = ID(A,i)$ and $Q = ID(B,j)$.

With the above in mind, the assembly process for element e looks as follows:

Assembly for Element e

```
for A = 1,...,d
    for a = 1,...,n_loc
        Set p = d(a-1) + A
        Set P = LM(A,a,e)
        Set i = IEN(a,e)

        if BC(A,i) = 0
            for B = 1,...,d
                for b = 1,...,n_loc
                    Set q = d(b-1) + B
                    Set Q = LM(B,b,e)
                    Set j = IEN(b,e)

                    if BC(B,j) = 0
                        Update K_PQ = K_PQ + k^e_pq
                    else
                        Update F_P = F_P - k^e_pq (g_j)_B
                    endif
                endloop
            endloop

            Update F_P = F_P + f^e_p
        endif
    endloop
endloop
```