Linear Elasticity : Constructing the Element Stiffness Matrix and Forcing Vector

We construct the element stiffness matrix and forcing vector in the same manner as the setting of heat conduction, namely by exploiting the isoparametric concept, Bézier extraction, and quadrature.

With this in mind, the element stiffness matrix entries are computed as follows:

$$
\begin{aligned}
k^e_{pq} &= \int_{\Omega^e} \vec{\varepsilon}\left(N^e_a \hat{e}_A\right)^T \vec{\vec{D}}\, \vec{\varepsilon}\left(N^e_b \hat{e}_B\right) d\Omega^e \\
&= \hat{e}_A^T \int_{\Omega^e} \left(\vec{\vec{B}}^e_a\right)^T \vec{\vec{D}}\, \vec{\vec{B}}^e_b \, d\Omega^e \, \hat{e}_B \\
&\overset{\text{n}}{=} \hat{e}_A^T \left(\sum_{q_1=1}^{n_q} \sum_{q_2=1}^{n_q} \left(\vec{\vec{B}}^e_a\left(\vec{x}^e\left(\tilde{\xi}_{q_1}, \tilde{\xi}_{q_2}\right)\right)\right)^T \vec{\vec{D}} \left(\vec{x}^e\left(\tilde{\xi}_{q_1}, \tilde{\xi}_{q_2}\right)\right) \vec{\vec{B}}^e_b \left(\vec{x}^e\left(\tilde{\xi}_{q_1}, \tilde{\xi}_{q_2}\right)\right) * \right. \\
&\qquad \left. \tilde{w}_{q_1} * \tilde{w}_{q_2} * j\left(\tilde{\xi}_{q_1}, \tilde{\xi}_{q_2}\right) \right) \hat{e}_B
\end{aligned}
$$

where we have utilized the same notation as the setting of heat conduction.

We split the element forcing vector into two contributions:

$$
f^e_p = \underbrace{\int_{\Omega^e} N^e_a f_A \, d\Omega^e}_{\text{Term 1}} + \underbrace{\int_{\Gamma^1_{N_A} \cap \Gamma^e} N^e_a h_A \, d\Gamma^e}_{\text{Term 2}}
$$

We construct Term 1 in the same manner as k^e_{pq} above:

$$
\text{Term 1} \overset{\text{n}}{=} \sum_{q_1=1}^{n_q} \sum_{q_2=1}^{n_q} N^e_a\left(\vec{x}^e\left(\tilde{\xi}_{q_1}, \tilde{\xi}_{q_2}\right)\right) * f_A\left(\vec{x}^e\left(\tilde{\xi}_{q_1}, \tilde{\xi}_{q_2}\right)\right) * \tilde{w}_{q_1} * \tilde{w}_{q_2} * j\left(\tilde{\xi}_{q_1}, \tilde{\xi}_{q_2}\right)
$$

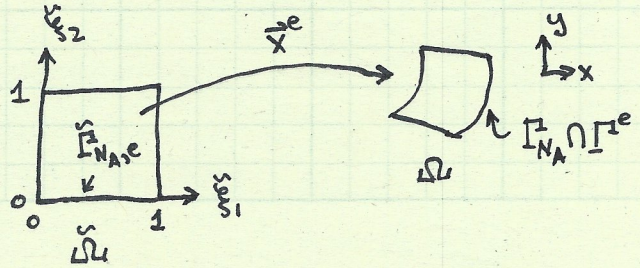
To deal with Term 2, we need to precisely identify the boundary $\Gamma^1_{N_A} \cap \Gamma^e$. As in the setting of heat conduction, let us define:

$$
\tilde{\Gamma}^1_{N_A, e} := \vec{x}^{e^{-1}}\left(\Gamma^1_{N_A} \cap \Gamma^e\right) \quad \text{(preimage of Neumann boundary)}
$$

Suppose that:

$$
\tilde{\Gamma}^1_{N_A, e} = \left\{ \tilde{\xi} = (t, 0) : t \in (0, 1) \right\}
$$

Visually:



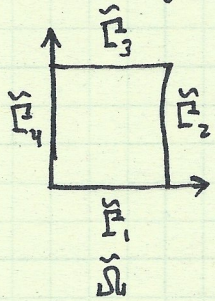
Then we approximate:

$$\text{Term 2} \cong \sum_{q=1}^{n_q} N_a^e(\vec{x}^e(\tilde{\xi}_{q,0})) * h_A(\vec{x}^e(\tilde{\xi}_{q,0})) * \tilde{w}_q * j_\Gamma(\tilde{\xi}_{q,0})$$

where:

$$j_\Gamma(\tilde{\xi}_{q,0}) = |\underline{\underline{J}}(\tilde{\xi}_{q,0})\underline{\tilde{t}}| = |\underline{\underline{J}}(\tilde{\xi}_{q,0})\begin{bmatrix}1\\0\end{bmatrix}|$$

We have analogous expressions for boundary integrals over the other three sides of $\tilde{\Gamma}$. To aid in implementation, we create a Neumann array which identifies if a given side of a Bézier element belongs to a Neumann boundary.



$$\text{Neumann}(A, a, e) = \begin{cases} 1 & \text{if } \vec{x}^e(\tilde{\Gamma}_a) \subseteq \Gamma_{N_A} \\ 0 & \text{otherwise} \end{cases}$$

D.O.F. — Side — Element

Note that the overall flow of the formation and assembly process for linear elasticity is identical to that of heat conduction. Basis function values and derivatives are moreover also calculated using ShapeFunction. With this in mind, we have the following pseudocode for element formation.

```
for q1 = 1,...,nq
    for q2 = 1,...,nq

        Call ShapeFunction(e, ξ̃_{q1}, ξ̃_{q2}) which computes the NURBS basis functions
        R^e_a, their derivatives ∂R^e_a/∂x and ∂R^e_a/∂y, the location x⃗, and the
        Jacobian matrix J at x⃗^e(ξ̃_{q1}, ξ̃_{q2})

        Set j = det(J)
        Set D⃗_loc = D⃗(x⃗)
        Set f⃗_loc = f⃗(x⃗)

        for a = 1,...,n_loc
            Form B⃗^e_a = [ N^e_{a,1}   0
                           0           N^e_{a,2}
                           N^e_{a,2}   N^e_{a,1} ]
            for b = 1,...,n_loc
                Form B⃗^e_b = [ N^e_{b,1}   0
                               0           N^e_{b,2}
                               N^e_{b,2}   N^e_{b,1} ]
                Set k_loc = (B⃗^e_a)^T D⃗_loc B⃗^e_b * w̃_{q1} * w̃_{q2} * j
```

Loop over Element Interiors

for $A = 1, \ldots, d$
  for $B = 1, \ldots, d$

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

Update $k^e_{pq} = k^e_{pq} + \hat{e}_A^T \underline{\underline{k}}_{loc} \hat{e}_B$

endloop
endloop
endloop

for $A = 1, \ldots, d$

Set $p = d(a-1) + A$
Update $f^e_p = f^e_p + R^e_a * (\vec{f}_{loc})_A * \tilde{w}_{q_1} * \tilde{w}_{q_2} * j$

endloop
endloop
endloop
endloop

Loop over Element Boundaries

for $A = 1, \ldots, d$
  for side $= 1, \ldots, 4$

if Neumann$(A, side, e) = 1$
  for $q = 1, \ldots, n_q$

Set $\tilde{\xi}_1 = \begin{cases} \tilde{\xi}_q & \text{if side} = 1, 3 \\ 1 & \text{if side} = 2 \\ 0 & \text{if side} = 4 \end{cases}$

Set $\tilde{\xi}_2 = \begin{cases} \tilde{\xi}_q & \text{if side} = 2, 4 \\ 1 & \text{if side} = 3 \\ 0 & \text{if side} = 1 \end{cases}$

Set $\underline{\tilde{t}} = \begin{cases} \begin{bmatrix} 1 \\ 0 \end{bmatrix} & \text{if side} = 1, 3 \\ \begin{bmatrix} 0 \\ 1 \end{bmatrix} & \text{if side} = 2, 4 \end{cases}$

Call Shape Function $(e, \tilde{\xi}_1, \tilde{\xi}_2)$ to compute NURBS basis functions and derivatives, the location $\vec{x}$, and the Jacobian matrix $\underline{\underline{J}}$ at the point $\vec{x}^e(\tilde{\xi}_1, \tilde{\xi}_2)$

Set $j_{\Gamma^2} = |\underline{\underline{J}}\, \underline{\tilde{t}}|$

Set $h_{A\,loc} = h_A(\vec{x})$

for $a = 1, \ldots, n_{loc}$

Set $p = d(a-1) + A$

Update $f_p^e = f_p^e + R_a^e * h_{A\,loc} * \hat{w}_q * j_{\Gamma^2}$

endloop

endloop

end if

endloop

endloop