1 FEM Solver Implementation

Full weak form:

$$-\delta W = \underbrace{\int_{\Omega_0} \boldsymbol{S} : \delta \boldsymbol{E} \mathrm{d}V}_{-\delta W_{\mathrm{int}}} \underbrace{- \int_{\Omega_0} \hat{\boldsymbol{b}}_0^T \delta \boldsymbol{u} \mathrm{d}V - \int_{\Gamma_{0;\sigma}} \hat{\boldsymbol{t}}_0^T \delta \boldsymbol{u} \mathrm{d}A}_{-\delta W_{\mathrm{ext}}} \stackrel{!}{=} 0. \tag{1.1}$$

We neglect δW_{ext} for now, and focus on internal work:

$$-\delta W_{\mathrm{int}} = \int_{\Omega_0} \boldsymbol{S} : \delta \boldsymbol{E} \mathrm{d}V \equiv \int_{\Omega_0} \delta \boldsymbol{E} : \boldsymbol{S} \mathrm{d}V \stackrel{!}{=} 0, \tag{1.2}$$

due to symmetry of both tensors.

We use the linear (engineering) strain

$$\boldsymbol{E} = \mathrm{symmetrize}(\nabla \boldsymbol{u}) = \frac{1}{2}(\nabla \boldsymbol{u} + (\nabla \boldsymbol{u})^T), \tag{1.3}$$

as well as the linear St.Venant-Kirchoff constitutive relation

$$\boldsymbol{S} = \boldsymbol{C}_{\mathrm{VK}} : \boldsymbol{E}, \tag{1.4}$$

$$(C_{\mathrm{VK}})_{ijkl} = \lambda \delta_{ij} \delta kl + \mu(\delta_{ik}\delta_{jl} + \delta_{il}\delta_{jk}), \tag{1.5}$$

with the Lamé constants λ and μ.

1.0.1 Discretization within an element

We use the discretized quantities (technically they should have an "h" subscript, but we leave it away here because we do not handle the analytical quantities anyways)

$$u_i = \sum_{k=1}^{\mathrm{nnode}} \mathrm{N}_k \mathrm{d}_{\mathrm{nodeldof2dof}(k,i)}, \tag{1.6}$$

$$\delta u_i = \sum_{k=1}^{\mathrm{nnode}} \mathrm{N}_k \delta \mathrm{d}_{\mathrm{nodeldof2dof}(k,i)}, \tag{1.7}$$

$$x_i = \sum_{k=1}^{\mathrm{nnode}} \mathrm{N}_k \mathrm{X}_{\mathrm{nodeldof2dof}(k,i)}, \tag{1.8}$$

where

$$\mathrm{nodeldof2dof}(k,i) = k \cdot \mathrm{nnode} + i \tag{1.9}$$

is the mapping from node k and node-local (node-scoped) dof i to the corresponding element-global (element-scoped) dof. The inverse of this mapping is

$$\mathrm{dof2nodeldof}(g) = \{\mathrm{floor}(g/\mathrm{ndofn}), \mathrm{mod}(g, \mathrm{ndofn})\}. \tag{1.10}$$

Given the need for the gradient of the unknown displacement in 1.3, we need to discretize it. In index notation,

$$\frac{\partial u_i}{\partial x_j} = \frac{\partial (\mathrm{Nd})_i}{\partial x_j} = \sum_{k=1}^{\mathrm{nnode}} \frac{\partial \mathrm{N}_k}{\partial x_j} \mathrm{d}_{\mathrm{nodeldof2dof}(k,i)}. \tag{1.11}$$

Now, we can find the gradient of the shape functions from the parametric coordinates and (inverse) Jacobian:

$$\frac{\partial \mathrm{N}_k}{\partial x_j} = \sum_{l=1}^{\mathrm{ndofn}} \frac{\partial \mathrm{N}_k}{\partial \xi_l} \frac{\partial \xi_l}{\partial x_j}. \tag{1.12}$$

To find the inverse Jacobian, we need the Jacobian:

$$J_{lj} = \frac{\partial x_j}{\partial \xi_l} = \frac{\sum_{k=1}^{\mathrm{nnode}} \mathrm{N}_k \mathrm{X}_{\mathrm{nodeldof2dof}(k,j)}}{\partial \xi_l} = \sum_{k=1}^{\mathrm{nnode}} \frac{\mathrm{N}_k}{\partial \xi_l} \mathrm{X}_{\mathrm{nodeldof2dof}(k,j)} \tag{1.13}$$