Part A: Shape Functions for Linear Tetrahedral Finite Elements

Problem 1

Start with the continuous definition of the potential energy for a linearly elastic object given by:

$$E = \frac{1}{2}\int_{\Omega} \epsilon\colon C\colon \epsilon d\Omega$$

For linear elasticity we can rewrite this using the

$$C = \frac{Y}{(1+\mu)\cdot(1-2\mu)}\begin{pmatrix} 1-\mu & \mu & \mu & 0 & 0 & 0 \\ \mu & 1-\mu & \mu & 0 & 0 & 0 \\ \mu & \mu & 1-\mu & 0 & 0 & 0 \\ 0 & 0 & 0 & \frac{1}{2}(1-2\mu) & 0 & 0 \\ 0 & & 0 & 0 & \frac{1}{2}(1-2\mu) & 0 \\ 0 & 0 & 0 & 0 & 0 & \frac{1}{2}(1-2\mu) \end{pmatrix}$$

$$\epsilon = \begin{pmatrix}\epsilon_{xx} & \epsilon_{yy} & \epsilon_{zz} & \epsilon_{yz} & \epsilon_{xy} & \epsilon_{xx}\end{pmatrix}^T \qquad \text{and} \qquad \epsilon_{ij} = \frac{1}{2}\left(\frac{\partial \mathbf{u}_i}{\partial \mathbf{x}_j} + \frac{\partial \mathbf{u}_j}{\partial \mathbf{x}_i}\right)$$

Here $\mathbf{u} \in \mathbb{R}^3$ is the displacement at a point $\mathbf{x}$ in space. Y is the stiffness of the material (Young's modulus) and μ is the poisson's ratio which determines how incompressible a material is. Start by rewriting E using the matrix C and the vector definition of ϵ.

$$E = \frac{1}{2}\int_{\Omega} \epsilon^t C \epsilon d\Omega$$

Problem 2

Next you will discretize E using linear finite element shape functions on a tetrahedron. The shape functions for a linear, tetrahedral element are the barycentric coordinates of each tetrahedron the formula here Let's use barycentric coordinates to represent the displacement of all points inside a tetrahedron: $\mathbf{u}(\mathbf{x}) = \mathbf{u}_1\lambda_1 + \mathbf{u}_2\lambda_2 + \mathbf{u}_3\lambda_3 + \mathbf{u}_4\lambda_4$, where $\mathbf{u}_i$ is the displacement of the i^{th} vertex and λ_i is the i^{th} barycentric coordinate. You should be able to build a discrete version of ϵ using this formula. Try to express your formula in the form $B\mathbf{u}$, where B is a matrix and $\mathbf{u}$ is the 12×1 vector of stacked nodal displacements: $\begin{pmatrix}u1_x & u1_y & u1_z & \cdots & u4_x & u4_y & u4_z\end{pmatrix}^T$

Part B: Implement Linear Shape Functions

1. Open 'SRC_DIRECTORY/A4/include/ShapeFunctionTetLinearA4.h'.
2. The function `*double phi(double x) { ... }` returns the value of the linear shape function λ_{VERTEX}, evaluated at $\mathbf{x}$. Implement this function.
3. The function `*std::array<DataType,3> dphi(double x) { ... }` returns the gradient linear shape function $\nabla\lambda_{VERTEX}$, evaluated at $\mathbf{x}$. Implement this function.
4. The function `*Eigen::Matrix<DataType, 3,3> F(double x, const State &state)` returns the 3×3 deformation gradient for a tetrahedral element where $F_{ij} = \frac{\partial \mathbf{u}_i}{\partial \mathbf{X}_j}$. Use **dphi** to implement this method.

Part C: Implement Quadrature and Simulate

1. Open 'SRC_DIRECTORY/A4/include/QuadratureLinearElasticity.h'
2. Complete the **getEnergy** function to return the value of E from Part A.

3. Compute the stiffness matrix and the forces from E. The forces are the negative gradient of E wrt to the degrees of freedom, $\mathbf{u}$ and the stiffness matrix is the hessian matrix, i.e $K_{ij} = -\frac{\partial^2 E}{\partial \mathbf{u}_i \partial \mathbf{u}_j}$. Derive these terms and complete the **getGradient** and **getHessian** functions.

4. Run the assignment code which will compute the stress on a test object and plot the result.