Homework Assignment No. 3

Assigned Tuesday, February 19, 2015.

Due Thursday, March 5, 2015.

Part I — A Membrane Finite Element

In this part of the assignment you will implement the nodal residual force vector and tangent stiffness matrix for an isoparametric finite element using the shape functions and quadrature tables from HW 2 and the 3-D neo-Hookean hyperelastic model from HW 1. Implement the computation of element integrals for three- and six-node isoparametric triangles of the strain energy,

$$W = \int_{\Omega_0} w dV$$

the internal nodal force,

$$f_{ia}^{\text{int}} = \int_{\hat{\Omega}} n_i^\alpha N_{a,\alpha} \sqrt{A} d\theta^1 d\theta^2, \quad \boldsymbol{n}^\alpha = \mathbf{P} \cdot \boldsymbol{G}^\alpha \mu H = \tau^{\alpha i} \boldsymbol{g}_i \mu H$$

the external nodal force due to a constant distributed load $\boldsymbol{f}$

$$f_{ia}^{\text{ext}} = \int_{\hat{\Omega}} f_i N_a \sqrt{A} d\theta^1 d\theta^2$$

and tangent stiffness

$$K_{iakb} = \int_{\hat{\Omega}} [\tilde{C}^{\alpha\beta\mu\nu} (\boldsymbol{a}_\beta \otimes \boldsymbol{a}_\nu)_{ik} + \tau^{\alpha\beta} \delta_\beta^\mu] N_{a,\alpha} N_{b,\mu} \sqrt{A} d\theta^1 d\theta^2,$$

$$\tilde{C}^{\alpha\beta\mu\nu} = C^{\alpha\beta\mu\nu} - \frac{C^{\alpha\beta 33} C^{33\mu\nu}}{C^{3333}}, C^{ijkl} = \frac{\partial \tau^{ij}}{\partial g_{kl}}.$$

Your element function(s) should take as input the reference positions of the nodes X_{ia} and the current positions x_{ia}, and should return the element energy W^e, and force and stiffness arrays f_{ia}^e and K_{iakb}^e.

To compute the stress resultants and tangent moduli, you will need to impose the plane stress condition, $\tau^{33} = \frac{\partial w}{\partial g_{33}} = 0$. Solve this by a local Newton iteration at each surface quadrature point. My suggestion is to split that calculation into a separate function, for instance one that takes as arguments the basis vectors $\boldsymbol{a}_\alpha$ and $\boldsymbol{A}_\alpha$, and thickness stretch g_{33}, and from them computes the stress resultants $\boldsymbol{n}^\alpha$ and their derivatives $\frac{\partial \boldsymbol{n}^\alpha}{\partial \boldsymbol{a}_\beta}$ making calls when necessary to the 3-D constitutive routine.

Make sure to verify the correctness of your element code by also approximating internal forces and stiffness by numerical differentiation using the 3-point rule, and test the rank of the stiffness matrix at zero and random finite deformations.

Part II — Membrane Equilibrium

Here you will code the assembly of the energy, force, and stiffness arrays of the elements in a mesh, and solve some nonlinear equilibrium problems.

1. Program the assembly of the global energy, force, and stiffness for a mesh of membrane elements. You can enforce prescribed displacement boundary conditions by simply skipping over the corresponding nodal force and stiffness components. You should encapsulate the global assembly operation within a function/subroutine that takes as arguments the current deformed positions of the nodes x_{ia} and returns W, $r_{ia} = f_{ia}^{\text{int}} - f_{ia}^{\text{ext}} = \partial W / \partial x_{ia}$, and $k_{iakb} = \partial r_{ia} / \partial x_{kb}$. (Your function may need to have other arguments as well, depending on what language and data structures you use. But the point is that you should have a function you can call to get energy, force, and stiffness for a specific configuration.)

   Test your global assembly function for consistency by computing r_{ia} and k_{iakb} numerically with the 3-point numerical differentiation rule. Also check to see that the stiffness has full rank (both at zero and non-zero deformation).

2. Solve the problem of a square sheet stretched by planar isotropic deformation ($F_{11} = F_{22} = \lambda$; $F_{12} = F_{21} = 0$), imposed by prescribing displacements of nodes on the boundaries. You may use a small mesh of only a few elements. Verify that the solution exhibits a uniform deformation gradient, and that the stresses agree with your results from HW 1.

3. Solve the problem of a simply supported square membrane under a uniform transverse distributed load. Let the edges be $L = 10$ cm, and the reference thickness be $H = 0.1$ cm. Use A shear modulus of $\mu_0 = 4 \times 10^5$ N/m^2 and $\lambda_0 = 10\mu_0$. Apply the load in increments of $\Delta f = 10$ N/m^2 up to $f_{\text{max}} = 10^3$N/m, and for each increment solve the nonlinear equilibrium equations $r_{ia} = 0$ using the Newton-Raphson method. Plot the load magnitude vs. the maximum deflection. Show plots of the deformed shape at the maximum load, and one-half the maximum load. Do this with meshes of linear and quadratic elements. Make the meshes fine enough that results appear to be converged.

4. Solve the problem of a spherical balloon of initial radius $R = 10$ cm and thickness $H = 0.1$ cm under a uniform internal pressure. Use the same material constants and load incrementation procedure as for the square membrane. How large a pressure load can you reach before the N-R iteration no longer converges? Plot the inflation pressure p vs. the lateral stretch ratio $\lambda = r/R$. Compare your answers to the exact analytical solution.

Note: to create meshes for the square and spherical membranes, you may wish to use the `delaunay` function in Matlab. This takes a set of points in 2-D and computes the Delaunay triangulation of its convex hull. For the spherical membrane, my suggestion is to model only one octant of the sphere. You can mesh this by first meshing an equilateral triangle in the plane; then rotate it so that each corner is equidistant from the origin in 3-D, and project all nodal positions radially outward to be equidistant from the origin.