Homework Assignment No. 2

Assigned Thursday, February 5, 2015.

Due Tuesday, February 19, 2015.

Part I — 2-D Shape Functions

In this part you will write code to implement and verify the calculation of the shape functions for linear and quadratic isoparametric triangular finite elements. As preparation for this you should read sections 4.1–4.8 of Chapter 4 of Zienkiewicz and Taylor, Volume 1: http://www.knovel.com/web/portal/browse/display?_EXT_KNOVEL_DISPLAY_bookid=1886

1. Code a function that calculates the shape functions and their parametric derivatives for a 3-node linear triangular element. This function should take as an input argument the values of the parametric coordinates $(\theta^1, \theta^2) \equiv (r, s)$ of an arbitrary point within the standard domain, which you should take to be the isosceles right triangle $\{0 \le r \le 1, 0 \le s \le r\}$, as discussed in class. The function should return the values of the three shape functions $N_a(\theta^1, \theta^2)$, and their derivatives $N_{a,\alpha} = \partial N_a / \partial \theta^\alpha$.

2. Code a second function, just like the one above, that computes shape functions and derivatives for the 6-node quadratic element.

3. Write a verification test program that tests the correctness of your linear and quadratic shape functions in two ways. Verify that for a random location in the element (θ^1, θ^2) the shape functions form a partition of unity
$$\sum_a N_a(\theta^1, \theta^2) = 1,$$
and their derivatives form a partition of nullity
$$\sum_a N_{a,\alpha}(\theta^1, \theta^2) = 0.$$

4. Write at program to verify the consistency of the shape function code you wrote in steps 1 and 2 by approximating the derivatives with the 3-point numerical derivative (just as you did for your constitutive code in HW 1). Run these tests on both your linear and quadratic shape functions, and verify that they pass.

5. Test the C^0-completeness of your shape function implementations. In other words, verify that your shape functions can interpolate a random linear polynomial exactly. Do this with the following sequence of steps

    (a) Define a random linear polynomial
    $$p(\boldsymbol{\theta}) = \sum_{|\alpha| \le 1} a_\alpha \theta^\alpha$$
    by choosing coefficients a_α randomly.

    (b) Evaluate $p(\boldsymbol{\theta})$ at a random point, $\boldsymbol{\theta}^*$, i.e., $p^* = p(\boldsymbol{\theta}^*)$.

    (c) Sample $p(\boldsymbol{\theta})$ at the nodes, $p_a = p(\boldsymbol{\theta}_a)$.

(d) Interpolate the sampled nodal values

$$p_I^* = \sum_{a=1}^{N} p_a N_a(\boldsymbol{\theta}^*).$$

(e) Compare p^* and p_I^*.

Part II — 2-D Plane Stress Element

In this part of the assignment you will derive the nodal internal force vector and tangent stiffness matrix for an isoparametric finite element using the shape functions from Part I and the plane-stress neo-Hookean hyperelastic model from HW 1. As preparation for this you should read most of Chapter 5 (just skim secs. 13–15, read the rest) of Zienkiewicz and Taylor, Volume 1: http://www.knovel.com/web/portal/browse/display?_EXT_KNOVEL_DISPLAY_bookid=1886

1. Program a function that returns tables of positions and weights for first and second-order triangular Gaussian quadrature rules (one-point and three-point rules, found in Sec. 5.11 of Zienkiewicz and Taylor, or in Appendix 3.1 of Hughes, or in the MAE M168 reader). You should demonstrate the verification of your tables by integrating arbitrary (random) polynomials

$$p(\boldsymbol{\theta}) = \sum_{|\alpha|\leq m} a_\alpha \theta^\alpha$$

where $m = 1$ for the first order rule and $m = 2$ for the second order rule.

2. Implement the computation of element integrals for three- and six-node isoparametric triangles using your plane-stress neo-Hookean elasticity routines from HW 1, your shape function routines from Part I, and the quadrature tables from above. In particular you should program one or more functions to compute the strain energy,

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

internal nodal force array,

$$f_{ia}^{\text{int}} = \int_{\Omega_0} P_{iJ} N_{a,\alpha} \mathbb{J}_{\alpha J}^{-1} dV, \quad i, J, \alpha \in \{1, 2\}$$

and stiffness array

$$K_{iakb} = \int_{\Omega_0} C_{iJkL}^{\text{2D}} N_{a,\alpha} N_{b,\beta} \mathbb{J}_{\alpha J}^{-1} \mathbb{J}_{\beta L}^{-1} dV, , \quad i, k, J, L, \alpha, \beta \in \{1, 2\}$$

(we will neglect body forces and surface tractions for now). 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. (You might also want to pass in some flag that specifies the quadrature order, but that's up to you.)

3. Verify the correctness of your element code by also approximating internal forces and stiffness by numerical differentiation using the 3-point rule.

4. Lastly, at both zero deformation ($x_{ia} = X_{ia}$), and a random finite deformation ($x_{ia} = X_{ia} + u_{ia}$, where u_{ia} = random), compute the rank of the stiffness matrix for the following combinations of shape functions and quadrature rules:

(a) Linear shape functions, first order quadrature.

(b) Quadratic shape functions, first order quadrature.

(c) Quadratic shape functions, second order quadrature.