



School of Engineering  
Brown University

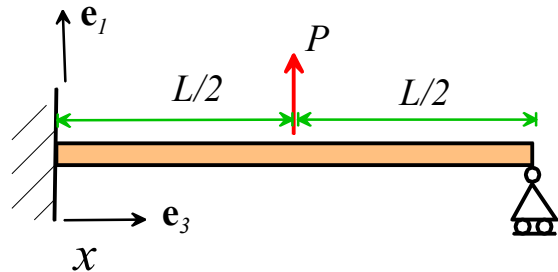
## EN1750: Advanced Mechanics of Solids

### Homework 7: Energy methods, Implementing FEA.

Due Friday Nov 8, 2019

1. The figure shows a beam that is clamped at  $x_3 = 0$  and pinned at  $x_3 = L$ . It is subjected to a point force  $P$  at mid-span  $x_3 = L/2$ . Its potential energy is

$$\Pi = \int_0^L \frac{1}{2} EI \left( \frac{d^2 v(x)}{dx^2} \right)^2 dx - P v(L/2)$$



where  $v$  is the upward deflection of the beam.

1.1 Start by re-writing the potential energy in dimensionless form, by defining dimensionless measures of deflection and position as follows

$$\hat{v} = \frac{vEI}{PL^3} \quad \hat{x} = \frac{x}{L} \quad \hat{\Pi} = \frac{\Pi EI}{P^2 L^3}$$

Show that the dimensionless form for potential energy is independent of loading, geometry, or material properties.

1.2 Use a displacement field of the form

$$\hat{v}(x) = \sum_{n=1}^N a_n \hat{x}^{n-1}$$

to obtain a Rayleigh-Ritz approximation to the deflected shape of the beam. Write a MATLAB script (you can modify the one from class – you will only need to change a few lines in the code) to calculate  $\hat{v}(\hat{x})$  for an arbitrary number of terms in the approximation. Plot a graph of the solution with 5, 10 and 15 terms in the series. Hand in a copy of your plot, there is no need to submit MATLAB code

**2. Health warning:** This is a challenging problem – to solve it you will need to understand how FEA works in some depth, and will also need to look at the sample FEA code provided to understand how the mesh, the boundary conditions, and the solution for displacements are stored as matrices.

Your goal is to modify the simple plane stress/plane strain FEA code provided in class so that it will solve problems involving *axially symmetric* solids. The figure shows a representative problem to be solved. The mesh represents a slice through an axially symmetric cylinder, which is prevented from stretching vertically, and pressurized on its interior surface.

The solid is meshed using axisymmetric triangular elements (so the elements extend all the way around the circumference of the cylinder; but the out-of-plane parts of the elements are all hard coded and are not displayed in the mesh). The radial and vertical displacements are interpolated in each element as

$$u_r(r, z) = u_r^{(a)} N_a(r, z) + u_r^{(b)} N_b(r, z) + u_r^{(c)} N_c(r, z)$$

$$u_z(r, z) = u_z^{(a)} N_a(r, z) + u_z^{(b)} N_b(r, z) + u_z^{(c)} N_c(r, z)$$

where

$$N_a(x_1, x_2) = \frac{(x_2 - x_2^{(b)})(x_1^{(c)} - x_1^{(b)}) - (x_1 - x_1^{(b)})(x_2^{(c)} - x_2^{(b)})}{(x_2^{(a)} - x_2^{(b)})(x_1^{(c)} - x_1^{(b)}) - (x_1^{(a)} - x_1^{(b)})(x_2^{(c)} - x_2^{(b)})}$$

$$N_b(x_1, x_2) = \frac{(x_2 - x_2^{(c)})(x_1^{(a)} - x_1^{(c)}) - (x_1 - x_1^{(c)})(x_2^{(a)} - x_2^{(c)})}{(x_2^{(b)} - x_2^{(c)})(x_1^{(a)} - x_1^{(c)}) - (x_1^{(b)} - x_1^{(c)})(x_2^{(a)} - x_2^{(c)})}$$

$$N_c(x_1, x_2) = \frac{(x_2 - x_2^{(a)})(x_1^{(b)} - x_1^{(a)}) - (x_1 - x_1^{(a)})(x_2^{(b)} - x_2^{(a)})}{(x_2^{(c)} - x_2^{(a)})(x_1^{(b)} - x_1^{(a)}) - (x_1^{(c)} - x_1^{(a)})(x_2^{(b)} - x_2^{(a)})}$$

are the same interpolation functions used to analyze plane deformations, and

$(u_r^{(a)}, u_z^{(a)})$ ,  $(u_r^{(b)}, u_z^{(b)})$ ,  $(u_r^{(c)}, u_z^{(c)})$  are the 2D displacement vectors at the three corners of the elements

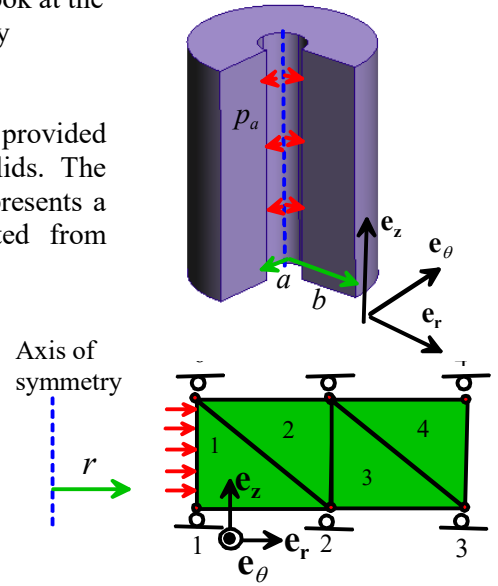
2.1 The nonzero strain components in the element (in cylindrical-polar cords) can be expressed as

$$[\varepsilon] = [B][u]$$

$$[\varepsilon] = \begin{bmatrix} \varepsilon_{rr} \\ \varepsilon_{zz} \\ \varepsilon_{\theta\theta} \\ 2\varepsilon_{rz} \end{bmatrix} \quad [u] = \begin{bmatrix} u_r^{(a)} & u_z^{(a)} & u_r^{(b)} & u_z^{(b)} & u_r^{(c)} & u_z^{(c)} \end{bmatrix}^T$$

where [B] is a 4x4 matrix containing derivatives of the interpolation functions, and (for axisymmetric elements) some additional terms that are functions the interpolation functions (not their derivatives). Find a formula for [B]. [B] will be very similar to the 3x4 matrix discussed in class, but needs an extra row to calculate  $\varepsilon_{\theta\theta}$ . You can assume that the nonzero strain components are

$$\varepsilon_{rr} = \frac{\partial u_r}{\partial r} \quad \varepsilon_{\theta\theta} = \frac{u_r}{r} \quad \varepsilon_{zz} = \frac{\partial u_z}{\partial z} \quad 2\varepsilon_{rz} = \frac{\partial u_z}{\partial r} + \frac{\partial u_r}{\partial z}$$



2.2 Let  $\sigma = [\sigma_{rr} \quad \sigma_{zz} \quad \sigma_{\theta\theta} \quad \sigma_{rz}]$  denote the stress in the element. Find a matrix  $[D]$  that satisfies  $[\sigma] = [D][\varepsilon]$

2.3 Write down an expression for the strain energy density  $U^{el}$  within the element, in terms of the vectors and matrices  $[u], [B], [D]$

2.4 The total strain energy of each element must be computed. Note that each element represents a cylindrical region of material around the axis of symmetry. The total strain energy in this material follows as

$$W^{el} = \int_{A_{el}} 2\pi r U^{el} dr dz$$

The energy can be computed with sufficient accuracy by evaluating the integrand at the centroid of the element, and multiplying by the area of the element, with the result

$$W^{el} = 2\pi A_{el} \bar{r} \bar{U}^{el}$$

where  $\bar{r}$  denotes the radial position of the element centroid, and  $\bar{U}^{el}$  is the strain energy density at the element centroid. Use this result to deduce an expression for the element stiffness, and modify the procedure `elstif()` in the Matlab code to compute the element stiffness. You will hand in your MATLAB code as a solution to this problem.

2.5 The contribution to the potential energy from the pressure acting on element faces must also be computed. Following the procedure described in class, the potential energy is

$$P = - \int_0^L 2\pi r t_i u_i ds$$

where

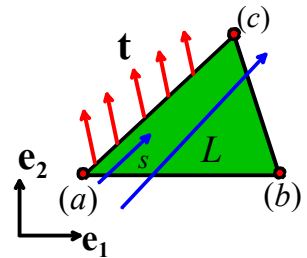
$$u_i = u_i^{(a)} \frac{s}{L} + u_i^{(c)} \left(1 - \frac{s}{L}\right) \quad r = r^{(a)} \frac{s}{L} + r^{(c)} \left(1 - \frac{s}{L}\right)$$

and  $u_i^{(a)}, u_i^{(c)}$  denote the displacements at the ends of the element face, and

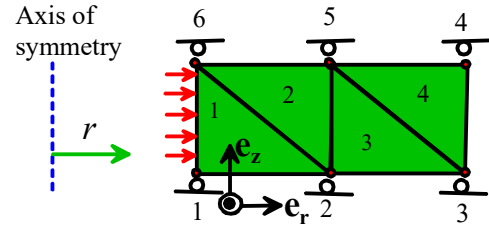
$r^{(a)}, r^{(c)}$  denote the radial position of the ends of the element face. Calculate an expression for  $P$  of the form

$$P^{\text{element}} = -[t_1 A \quad t_2 A \quad t_1 B \quad t_2 B] \cdot [u_1^{(a)} \quad u_2^{(a)} \quad u_1^{(c)} \quad u_2^{(c)}]$$

where  $A$  and  $B$  are constants (which depend on  $r^a, r^c$  that you must determine. Modify the procedure `elresid()` to implement modified element force vector.



2.6 Modify the code provided on the course website to calculate the stress in a pressurized cylinder, which has inner radius 1, exterior radius 3, and is subjected to pressure  $p=1$  on its internal bore (all in arbitrary units), and deforms under plane strain conditions (axially). The figure shows an example mesh with 6 nodes, with the appropriate boundary conditions. The length in the  $z$  direction is not important – choose anything sensible. You can run simulations with Young's modulus  $E=100$  and Poissons ratio 0.3.



Compare the FEA solution for displacements and stresses with the exact solution. Run tests with different mesh densities, and compare the results with the analytical solution (you can find the analytical solution in the online notes).

You will need to make the following changes to the code:

- (1) Change the D matrix to a 4x4 matrix
- (2) Modify the 'elstif' function to change the B matrix
- (3) Modify the 'elresid' function to find the axisymmetric force vector
- (4) Modify the 'elstrain' function to find the strain at the centroid of the element using the modified B matrix
- (5) Add functions to calculate the exact displacements at the nodes, and the exact stresses at the element centroids, for comparison to the FEA solution.

You will also need to set up the mesh and boundary conditions to solve the pressurized cylinder problem. To do this, you can either edit the text file provided with the sample code, but it will be a bit painful to type in all the coordinates for a fine mesh. If you prefer, you could also write code to set up the mesh automatically, instead of reading it from the file. To do this, you will need to replace the lines that read the text input file

```
cellarray=textscan(infile, '%s');
[E, nu, nnode, coord, nelem, connect, nfix, fixnodes, ndload, dloads] =
read_file(cellarray);
fclose(infile);
```

with code that defines each one of the variables

```
[E, nu, nnode, coord, nelem, connect, nfix, fixnodes, ndload, dloads]
```

For example, the code below would generate the coordinates for a mesh with 50 elements, using the same node numbering convention illustrated in the simple 4 element mesh.

```
a = 1;
b = 3;
p = 1;
```

```
E = 100;
nu = 0.3;
nelem = 50;
nnode = nelem + 2;
coord(1:nnode/2,1) = [a:2*(b-a)/nelem:b];
coord(1:nnode/2,2) = 0.;
coord(nnode/2+1:nnode,1) = [b:-2*(b-a)/nelem:a];
coord(nnode/2+1:nnode,2) = 1;
```