



EN2340: Computational Methods in Structural Mechanics

Homework 2: Basic FEA procedures Due Friday September 29 3017

School of Engineering
Brown University

The main goal of the homework is to get comfortable with the book-keeping operations needed perform the various FEA tasks.

You can use the constant strain linear elastic finite element MATLAB code posted on the [Notes page](#) of the class website as a general template for your calculations.

The goal is to develop a simple finite element code to predict the velocity field in an incompressible fluid, in the quasi-static (Stokes flow) limit. Suppose that the fluid occupies a two-dimensional area A in the (x_1, x_2) plane. Let $v_i(x_1, x_2)$ denote the velocity field in A . The velocity field satisfies

$$\frac{\partial^2 v_i}{\partial x_j \partial x_j} + M \frac{\partial v_j}{\partial x_i \partial x_j} = 0$$

Where M is a (large) penalty coefficient that enforces approximately the incompressibility condition $\partial v_i / \partial x_i = 0$.

It can be shown that (if body forces and tractions on the exterior surface vanish) the velocity field minimizes the functional

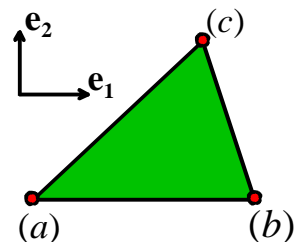
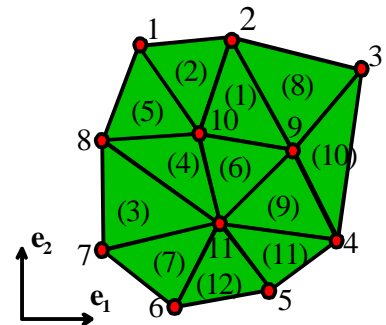
$$\Pi = \int_A \frac{1}{2} \left(\frac{\partial v_i}{\partial x_j} \frac{\partial v_i}{\partial x_j} + M \frac{\partial v_i}{\partial x_i} \frac{\partial v_j}{\partial x_j} \right) dA$$

In this problem you will develop a two-dimensional FEA code in MATLAB to solve this equation. The procedure follows the basic scheme for linear elasticity discussed in class (see also chapter 7 of <http://solidmechanics.org>):

1. We identify a set of 'nodes' in the solid where we want to calculate \mathbf{v}
2. Divide the solid into triangles with the nodes at the corners
3. Interpolate the velocity field v_i in each triangle linearly between values at the nodes. So, for a generic triangle with corners a, b, c

$$v_i = N^a v_i^a + N^b v_i^b + N^c v_i^c$$

where the interpolation functions are exactly the same as those used for the elasticity problem.



4. We now need to calculate the contribution to Π from each element. We must express this in matrix form

$$\pi^{elem} = \frac{1}{2} \left(\underline{v}^{elem T} k^{elem} \underline{v}^{elem} \right)$$

$$\underline{v}^{elem} = \begin{bmatrix} v_1^a \\ v_2^a \\ v_1^b \\ v_2^b \\ v_1^c \\ v_2^c \end{bmatrix}$$

where k^{elem} is a matrix that you must determine, as described below.

5. Add up the contributions from each element to express the total potential Π in terms of the displacements at each node, where

$$\Pi = \frac{1}{2} \left(\underline{U}^T [K] \underline{U} \right) \quad [K] = \sum_{elements} k^{elem}$$

where \underline{U} is a big vector consisting of the unknown transverse displacements at each node in the entire mesh.

6. Minimize the functional – follow the procedure discussed in class for linear elasticity
7. Solve the linear system, and post-process the results to display them.

With this preamble, please work through the following problems:

1. Find a matrix $[B]$ and a vector \underline{b} (in terms of derivatives of N^a, N^b, N^c) that have the following properties

$$\begin{bmatrix} \frac{\partial v_1}{\partial x_1} \\ \frac{\partial v_2}{\partial x_1} \\ \frac{\partial v_1}{\partial x_2} \\ \frac{\partial v_2}{\partial x_2} \end{bmatrix} = [B] \underline{v}^{elem} \quad \frac{\partial v_1}{\partial x_1} + \frac{\partial v_2}{\partial x_2} = \underline{b} \cdot \underline{v}^{elem}$$

2. Hence, find an expression for the element stiffness matrix k^{elem} in terms of $[B]$, \underline{b} the area of the element, and the constant M .
3. Modify the constant strain triangle MATLAB FEA code provided in class to compute k^{elem}

4. Modify the MATLAB FEA code provided in class to generate the mesh shown in the figure (you don't need to duplicated it exactly – just generate something similar). To do this you will need to:

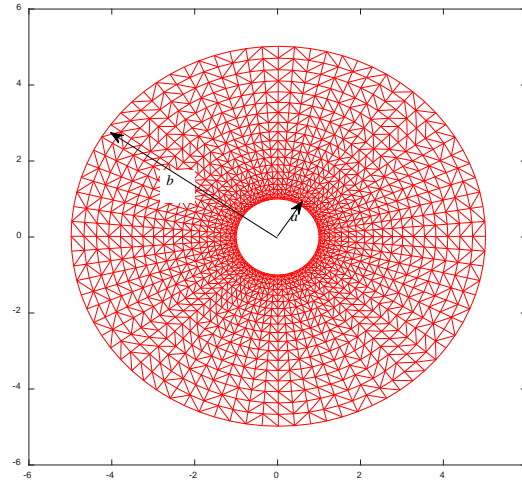
(1) Generate the array `coord(a,i)` that specifies the i th coordinate of the a th node in the mesh

$$x_i^a$$

(2) Generate the connectivity array `connect(lmn,a)`, which specifies the a th node attached to element number `lmn`. You can use the matlab function `Delaunay()` to do this, as follows

```
connect =  
    delaunay(coord(:,1),coord(:,2));
```

This will generate elements to connect all the nodes in the mesh (and will fill the hole at the center of the mesh with elements). You can then loop over all the elements and delete the ones which have centroids inside the hole.



The exact location of the nodes and the number of elements in your mesh is not important, but make the mesh fine enough to give a reasonable resolution in both the radial and hoop directions. The mesh shown in the figure has 80 nodes in the hoop direction and 20 in the radial direction.

5. Generate boundary conditions corresponding to

- (i) a unit prescribed radial velocity on the inner boundary ($r=a$) and zero traction on $r=b$
- (ii) a unit prescribed tangential velocity on $r=a$ and prescribed velocity of zero on the outer boundary $r=b$.

You can specify boundary conditions in the array `fixednodes()` in the code, which stores data as follows:

`fixednodes(n,1)` specifies the node number for the n th constraint

`fixednodes(n,2)` specifies the degree of freedom number (1 or 2) that is to be constrained

`fixednodes(n,3)` specifies the value of the degree of freedom.

Zero traction is the 'natural' boundary condition so you don't need to do anything to create a traction free surface.

6. For 5(i) and 5(ii) respectively, plot (i) the radial velocity and (ii) the tangential velocity as a function of radius. Compare the numerical solution to the analytical results (i) $v_r = a / r$ and (ii)

$$v_\theta = b(r^2 - a^2) / [(b^2 - a^2)r] .$$

You can try various values for M , but values in the range $5 < M < 500$

give a reasonable compromise between enforcing the incompressibility constraint and producing (for large M) an ill-conditioned stiffness matrix.

As a solution to this homework, please (1) Upload your MATLAB code to Canvas; and (ii) upload a typed or scanned pdf file with your solutions to problems 1,2 and 6 to Canvas.