



EN2340: Computational Methods in Structural Mechanics

Homework 2: Basic FEA procedures

School of Engineering
Brown University

In this homework your goal is to write a simple MATLAB code to solve steady-state diffusion equations. The main goal is to get comfortable with the book-keeping operations needed to perform the various FEA tasks. You can use the constant strain linear elastic finite element MATLAB code posted here as a general template for your calculations. The biggest difference between the elasticity problem discussed in class and the problems you will solve here is that you only need to calculate values for a single degree of freedom (the concentration), instead of the displacement vector. This makes the calculation a lot simpler, but you will have to work through how to modify the book-keeping in the MATLAB code to change the number of degrees of freedom.

1. Let $0 < c(x_j) < 1$ denote the concentration of some species. The steady state diffusion equation is

$$\frac{\partial^2 c}{\partial x_i \partial x_i} = 0$$

with boundary conditions

- $c = c^*(x)$ on parts of the boundary S_1 where concentration is prescribed
- $D \frac{\partial c}{\partial x_i} n_i = j^*(x)$ on parts of the boundary S_2 where flux is prescribed. Here, D is the diffusion coefficient, n_i are components of a unit vector normal to the surface, and j^* is the flux of material into the solid.

It can be shown (many of you probably know how...) that the concentration field that satisfies $c = c^*(x)$ on S_1 and minimizes the function

$$\Pi = \frac{1}{2} \int_V \frac{\partial c}{\partial x_i} \frac{\partial c}{\partial x_i} dV - \frac{1}{D} \int_{S_2} j^* c dA$$

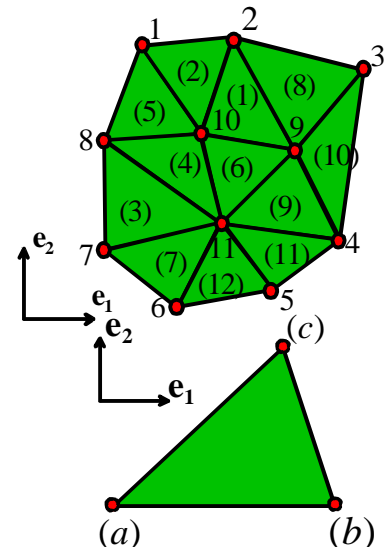
will satisfy the steady state diffusion equation and the boundary condition on S_2 .

In this problem you will develop a two-dimensional FEA code to solve this equation.

The basic scheme for linear elasticity discussed in class (see also chapter 7 of <http://solidmechanics.org>) will be modified as follows:

1. Identify a set of 'nodes' in the solid where we want to calculate c
2. Divide the solid into triangles with the nodes at the corners
3. Interpolate the voltage in each triangle linearly between values at the nodes. So, for a generic triangle with corners a, b, c

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



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

- Since the interpolation functions are linear, $\partial c / \partial x_i$ must be constant inside each triangle. We can compute the contribution to Π from just one element as

$$\pi^{elem} = \frac{1}{2} \underline{c}^{elem T} k^{elem} \underline{c}^{elem} \quad k^{elem} = A_{elem} [B]^T [B]$$

$$\underline{c}^{elem} = \begin{bmatrix} c^a \\ c^b \\ c^c \end{bmatrix} \quad [B] = \begin{bmatrix} \partial N^a / \partial x_1 & \partial N^b / \partial x_1 & \partial N^c / \partial x_1 \\ \partial N^a / \partial x_2 & \partial N^b / \partial x_2 & \partial N^c / \partial x_2 \end{bmatrix}$$

where A_{elem} is the area of the element.

- Compute the contribution to the potential from each element face with prescribed current

$$r^{elem} = \frac{L}{2D} J^* \begin{bmatrix} 1 \\ 1 \end{bmatrix}$$

where L is the length of the element face

- Add up the contributions from each element, and each element face, to express the total potential Π in terms of the voltages at each node, where

$$\Pi = \frac{1}{2} \underline{C}^T [K] \underline{C} - \underline{C}^T \underline{r} \quad [K] = \sum_{elements} k^{elem} \quad \underline{r} = \sum_{el \text{ faces}} r^{el}$$

where C is a big vector consisting of the unknown voltages at each node in the entire mesh.

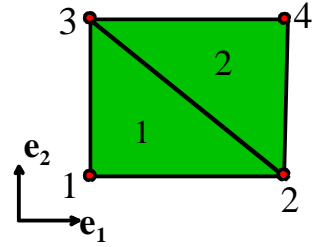
- Minimize – follow the procedure discussed in class for linear elasticity - this gives a system of linear equations for C
- Solve the equations, and post-process as needed.

Your mission is to implement this procedure as a MATLAB code, which should contain the following sections:

- Generate the arrays defining the mesh and boundary conditions. You could read the mesh definition from a file, following the structure of the linear elasticity code, or if you prefer just generate them directly using MATLAB.
- Assemble the global stiffness matrix and the boundary flux vector. This will require:
 - Loop over all the elements
 - For the current element, compute the element stiffness matrix (it is cleanest to do this by creating a function that calculates the element stiffness matrix)
 - Add the element stiffness matrix into the right rows and columns of the global stiffness matrix
 - Loop over all the elements with loaded faces
 - For the current element face, compute the element flux vector
 - Add the element flux vector into the global flux vector.
- Modify the global stiffness matrix to impose prescribed values for c at boundary nodes. This means
 - Loop over the nodes with prescribed concentrations
 - Find the row of the global stiffness matrix corresponding to the node, and set it to zero, with a 1 on the diagonal. Replace the value of the global flux vector \underline{r} corresponding to this row with the prescribed value of concentration.
- Solve the equation system

Test your code by

1. Calculating the concentration distribution in the simple two-element mesh shown in the figure. Apply a uniform flux to the top surface, and constrain the concentration at the bottom surface. Check your solution against the analytical result.
2. Calculating concentration distribution in a strip with a circular hole at its center. You can use the mesh in the example input file http://solidmechanics.org/FEM_MATLAB/FEM_conststrain_holeplate.txt for this purpose, or generate your own. Prescribe concentration on both left and right hand sides of the strip.



2. Optional – intended for people with prior FEA experience. As background read [Song et al Biophysical Journal 86 2017 \(2004\)](#). The Smoluchowski equation is a more complicated advection-diffusion equation of the form

$$\frac{\partial}{\partial x_i} D \left(\frac{\partial c}{\partial x_i} + c \frac{\partial W}{\partial x_i} \right) = 0$$

where W is a scalar potential. The weak form of the Smoluchowski equation is not symmetric, so some trickery is needed to turn it into a minimization problem. But introducing $\phi = c \exp(W)$ one can show that the solution for ϕ must minimize

$$\Pi = \frac{1}{2} \int_V \exp(-W) \frac{\partial \phi}{\partial x_i} \frac{\partial \phi}{\partial x_i} dV - \frac{1}{D} \int_{S_2} j^* \phi dA$$

One solution method (but not the one used in the article cited) is to compute ϕ by minimizing this expression, and then recover the value of c .

Modify the code you wrote in the preceding problem to solve the Smoluchowski equation using this approach. Test your code by comparing its predictions against analytical solutions. Try, for example

- (i) A rectangular strip (use at least 10 elements along the length) with prescribed concentration on each end, and a constant (but nonzero) value for W
- (ii) A strip with a prescribed flux on one end, and prescribed concentration on the other
- (iii) A strip with prescribed concentration one end, zero flux on the other, and a linearly varying function W (make the gradient of W large enough to see its effect)
- (iv) An annular region $a < r < b$ with potential $W(\mathbf{x}) = 1 / \sqrt{x_i x_i}$