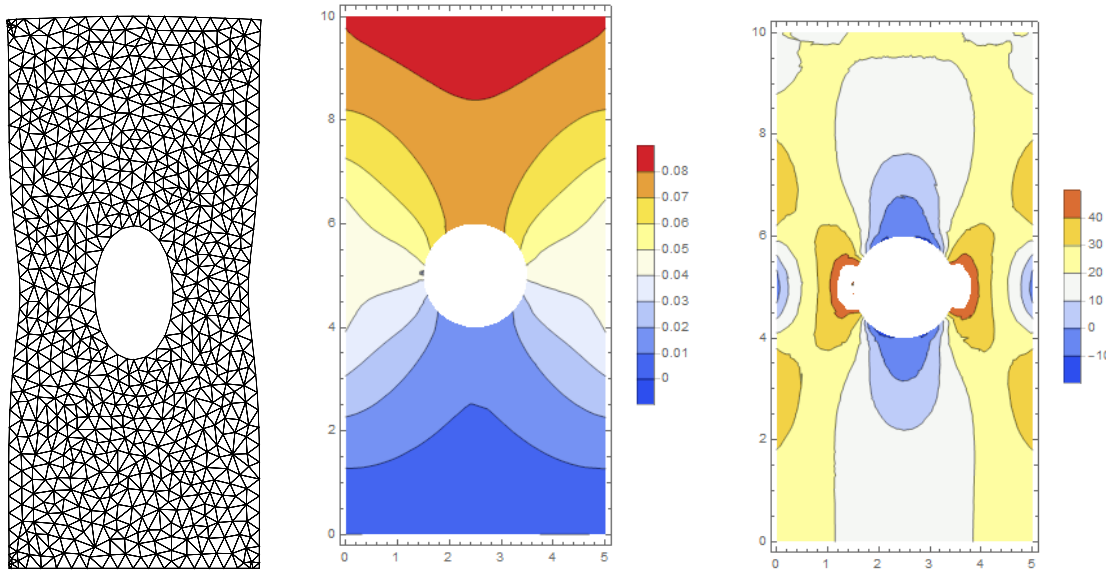
EN234: Computational methods in Structural and Solid Mechanics

Homework 4: FEA with Mathematica
Due Fri Oct 13, 2017

School of Engineering
Brown University



If you want to use FEA to solve solid mechanics problems, a special purpose package such as ABAQUS is probably your best option. It can be harder to adapt solid mechanics codes to solve materials science problems or multi-physics problems. In this case you might want to try one of the many high-level commercial or open-source codes – your options include

- Mathematica, which has a general purpose package for solving PDEs with finite elements
- COMSOL - a commercial general PDE solver designed for multi-physics problems
- FEniCS - a high-level open-source python based package designed to solve general PDEs
- MOOSE - a high-level open-source C++ based code for solving PDEs

In this homework, you will explore the PDE solver in Mathematica.

Start by working through a few tutorials (you don't need to work through all of them – you will learn quickly if you scan through them and then try solving your own problems and search the manuals for what you want to do).

1. General introduction to solving PDEs in Mathematica -try solving the heat transfer problem for the plate with a hole (you might need to change the dimensions a bit to get the geometry to work properly in some versions of Mathematica) and the beam bending problem
2. Mesh Generation
3. Controlling the Mathematica FEA solver

Constantinescu and Korsunsky "Elasticity with Mathematica" is a useful reference if you want to use Mathematica to do the various tensor and index notation manipulations that arise in solid mechanics.

The main attraction of Mathematica is that it hides the programming step between mathematical equations and a numerical solution. With that in mind, set up a Mathematica finite element solution to the field equations of linear elasticity (you can assume an isotropic solid):

(1) Strain-displacement relation $\boldsymbol{\varepsilon} = \frac{1}{2}\left\{\nabla \mathbf{u} + (\nabla \mathbf{u})^T\right\}$

(2) Constitutive relation $\boldsymbol{\sigma} = \mathbf{C} : \boldsymbol{\varepsilon}$. (There are several ways to set up this tensor product in Mathematica. One way is to define a function that initializes the 4th order tensor of moduli using (for an isotropic solid with Lame moduli λ, μ)

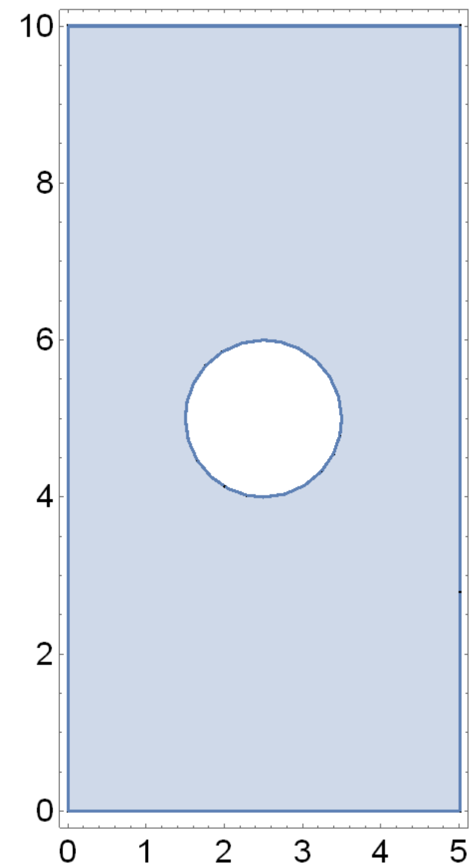
```
IsotropicStiffness2D[lambda_, mu_] :=
 Array[lambda KroneckerDelta[#1, #2] KroneckerDelta[#3, #4] +
    mu (KroneckerDelta[#1, #3] KroneckerDelta[#2, #4] + KroneckerDelta[#1, #4] KroneckerDelta[#2, #3]) &,
  {2, 2, 2, 2}]
```

You can then do the tensor product (after using the function to initialize CC) with

```
TensorContract[TensorProduct[CC, ϵ], {{3, 5}, {4, 6}}]
```

You can also use the matrix form of the constitutive equation, if that's easier.

(3) Equilibrium equation $\nabla \cdot \boldsymbol{\sigma} = \mathbf{0}$



Solve the following boundary value problem for the region shown in the figure:

(1) $u_2 = 0 \quad x_2 = 0 \quad 0 < x_1 < 5.01$

(2) $u_1 = 0 \quad x_1 = x_2 = 0$

(3) $\sigma_{22} = 20 \quad x_2 = 10, \; 0 < x_1 < 5.01$

(You might need to experiment with the Neumann boundary condition to work out the sign convention)

Try simulations for a material with shear modulus $\mu = 1000$ and Poisson's ratio $\nu = 0.3$.

Plot

(1) Contours of vertical displacement,

(2) See if you can work out how to plot contours of vertical stress σ_{22} in the plate – this requires some post-processing. The operations to define strains and stresses can be applied to the solution.

(3) Experiment with changing the mesh size, and compare results from linear and quadratic interpolation (try, e.g. a mesh size of 0.1 with linear and quadratic interpolation – this shows a large difference)

(4) Try a solution with Poisson's ratio $\nu = 0.499$. You will find that the displacements look physically reasonable (but are totally wrong), but the stresses will be clearly garbage. This is caused by locking – it's a good demonstration of the limits of current 'automatic' finite element solvers.

As a solution to this homework, please upload your Mathematica notebook to Canvas.