HOMEWORK 9

CEE 361-513: Introduction to Finite Element Methods

Due: Friday Dec. 15 @ Midnight

NB: Students taking CEE 513 must complete all problems. All other students will not be graded for problems marked with $\star$, but are encourage to attempt them anyhow.

PROBLEM 1:

1. Read and summarize Sections 4.1 - 4.4.1 (excluded)

```
Solution :
Refer the book.
```

PROBLEM 2: Incompressible Elasticity

The problem looks at 2-D incompressible elasticity. The problem reads : find $\boldsymbol{u} : \Omega \to \mathbb{R}^2$ and $p : \Omega \to \mathbb{R}$ such that :

$$\nabla \cdot \boldsymbol{\sigma}(\nabla \boldsymbol{u}, p) = \boldsymbol{f}, \quad \forall \boldsymbol{x} \in \Omega$$
$$\nabla \cdot \boldsymbol{u} + \frac{p}{\lambda} = 0, \quad \forall \boldsymbol{x} \in \Omega$$

and

$$\boldsymbol{u} = \boldsymbol{g} \quad \text{on } \Gamma_D$$
$$\boldsymbol{\sigma}(\nabla \boldsymbol{u}, p) \cdot \boldsymbol{n} = \boldsymbol{t} \quad \text{on } \Gamma_N$$

where Γ_D and Γ_N are the Dirichlet and Neumann boundaries respectively. Further,

$$\boldsymbol{\sigma}(\nabla \boldsymbol{u}, p) = -p\mathbf{1} + 2\mu\nabla^S \boldsymbol{u}$$

The specific problem we are are looking at is called "cook-membrane" problem. The basic problem configuration is summarized in the image below (1). A beam of specific dimensions is fixed at one end and a uniform traction load is applied at the other end such that the total force acting on this surface totals 1 N. Further, plane strain condition is assumed.

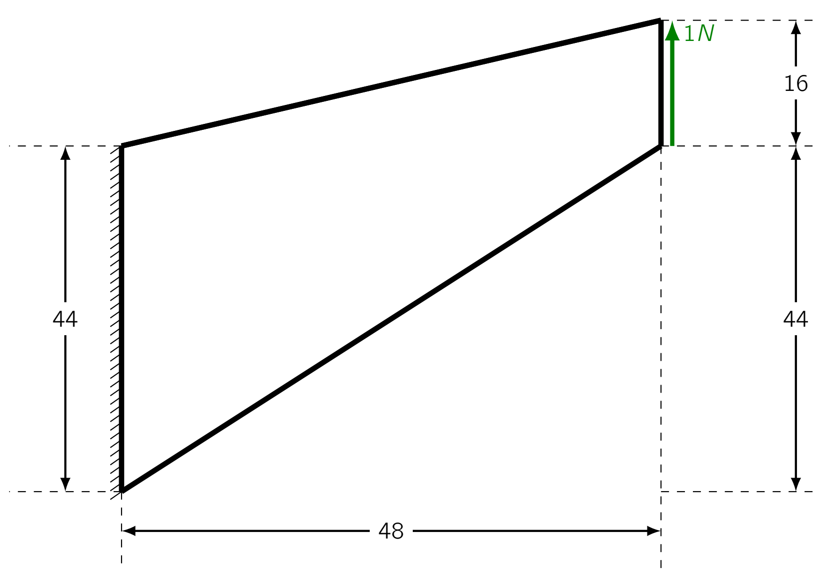

Figure 1: Cook Membrane

1. First we try to solve the problem using standard elasticity in the compressible region ($\nu = 0.3$). Modify the code provided `standard_elasticity.py` to solve the above problem. You would have to perform the following steps:

   (a) Redefine the Dirichlet and Neumann boundaries

   Solution :

```
class dirichlet_boundary(SubDomain):
  def inside(self, x, on_boundary):
      return abs(x[0]) < DOLFIN_EPS and on_boundary #<--- Fill here

# Describe the Neumann boundary
class neumann_boundary(SubDomain):
  def inside(self, x, on_boundary):
      return abs(x[0]-48.0) < DOLFIN_EPS and on_boundary #<--- Fill here
```

   (b) Modify the forcing functional

   Solution :

```
# Define the forcing functional
F = dot(t,v)*ds(1)   #<---- Fill here
```

   What is the maximum y-displacement?
   Plot the deformed shape.

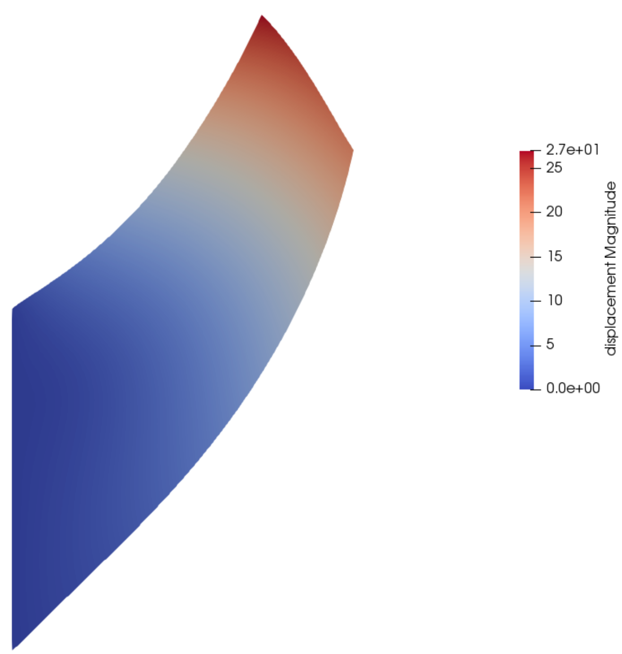

Figure 2: Deformed shape

2. Now we would solve the same problem using standard elasticity equations but in near incompressible region. How could you modify the code to achieve near incompressible condition? What is the maximum

y-displacement. Plot your deformed shape. Comment on your observation.

Solution :
To achieve the incompressible condition we change the poisson's ratio to ≈ 0.5

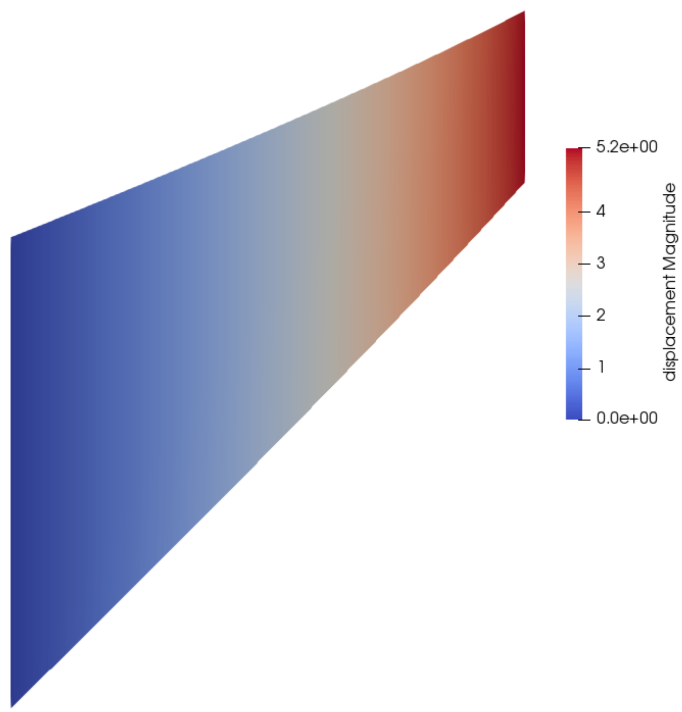

Figure 3: Deformed shape

The displacement is greatly reduced. This is expected as the standard elasticity formulation cannot capture the solution correctly for the incompressible problems.

3. Next, we solve the problem using the `incompressible_elasticity.py`. But before you could run the code, you need to perform the following:

   (a) Derive the weak form for the problem.

   Solution :
   The set of trial and test functions are:

$$\mathcal{S} = \{\boldsymbol{u} | \boldsymbol{u} \in [H^1(\Omega)]^d, \boldsymbol{u} = \boldsymbol{g} on \Gamma_D\}$$

$$\mathcal{V} = \{\boldsymbol{v} | \boldsymbol{v} \in [H^1(\Omega)]^d, \boldsymbol{v} = \boldsymbol{0} on \Gamma_D\}$$

$$\mathcal{P} = \{p, q | p, q \in [H^1(\Omega)]\}$$

   The weak form is given as:

$$\int_\Omega (\boldsymbol{\sigma} : \nabla \boldsymbol{v}) d\Omega - \int_\Omega (\nabla \cdot u + p/\lambda) q d\Omega = \int_{\Gamma_N} (\boldsymbol{t} \cdot \boldsymbol{v}) dS - \int_{\Gamma_D} \boldsymbol{f} \cdot \boldsymbol{v} dS$$

   (b) Define a function for $\boldsymbol{\sigma}$, basically complete the function sigma(u, p) on line# 68

Solution :

```
def sigma(u,p):
  eps = strain(u) #<--- Fill here
  return -p*Identity(space_dim)+ 2.0*mu*eps
```

(c) Use the Dirichlet and Neumann boundaries from the previous problem

Solution :

```
# First describe the dirichlet boundary
# The left edge is clammed
class dirichlet_boundary(SubDomain):
  def inside(self, x, on_boundary):
      return abs(x[0]) < DOLFIN_EPS and on_boundary #<--- Fill here

# Describe the Neumann boundary
class neumann_boundary(SubDomain):
  def inside(self, x, on_boundary):
      return abs(x[0]-48.0) < DOLFIN_EPS and on_boundary #<--- Fill here
```

(d) Fill the bilinear form on line#103

Solution :

```
# Define the variational form
a = inner(sigma(u,p),grad(v))*dx - q*(p/lmbda + div(u))*dx #<--- Fill here
```

Now run the code and:

(a) Report the maximum y-displacement

Solution :
The maximum y-displacement is 5.2

(b) Plot the deformed shape.

Solution :

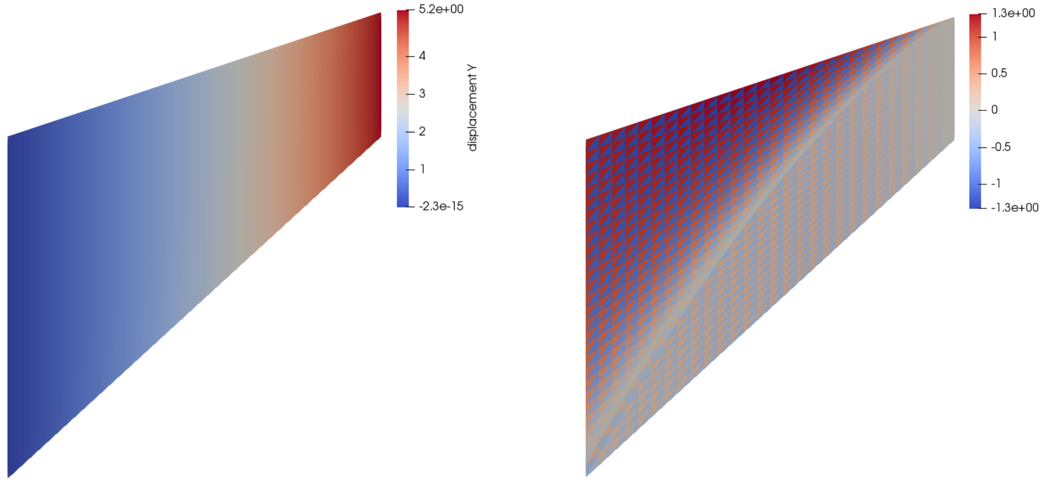

Figure 4: Deformed shape and pressure

(c) Comment on your observations.

Solution :
The above deformation and pressure is obtained because of the issue of locking. There are more constraints than degrees of freedom.

Now we modify the type of element used.

(a) Modify the type of displacement element to use polynomial order 2 and the pressure to use polynomial order 1.

(b) Plot the deformed shape.

Solution :

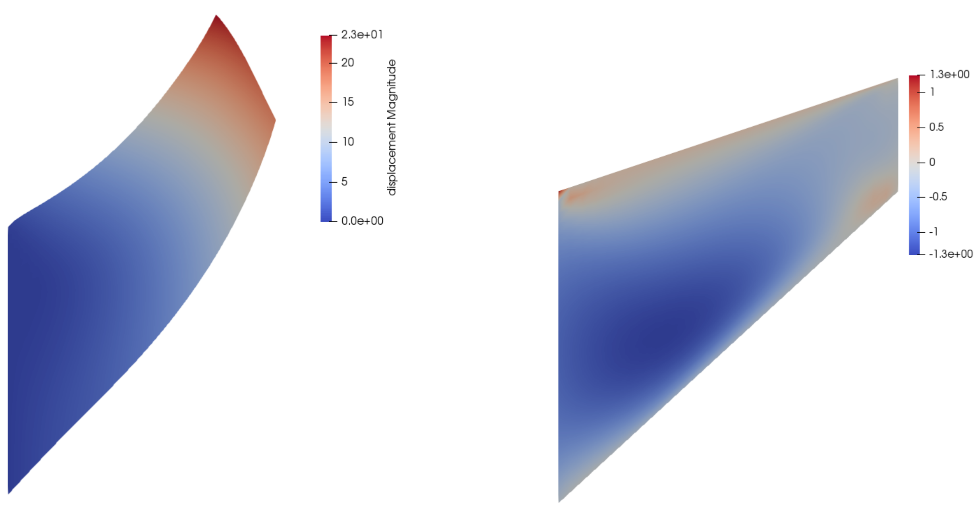

Figure 5: Deformed shape and pressure

(c) Why did it change the solution? [Hint : Refer Sec 4.3 of Hughes book]

Solution :
The solution was modified because the constraint count for this type of element is 2 which is optimal for solving the incompressible elasticity problem.