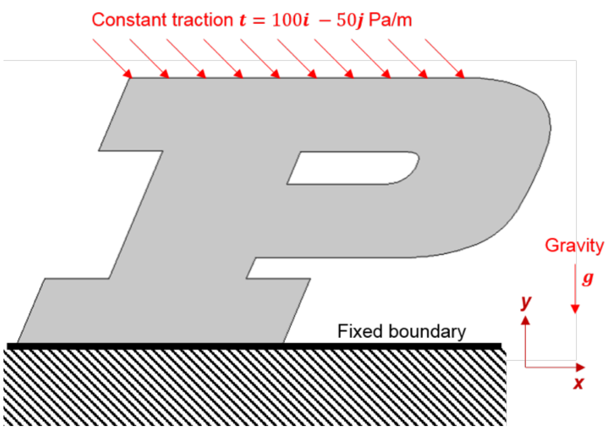
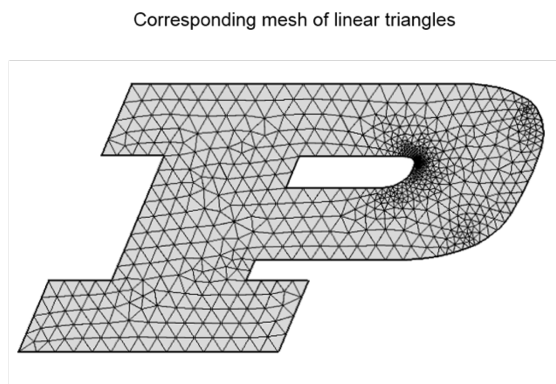
ME 489: Introduction to Finite Element Analysis

Homework 6
Released: 04/15/2019
Due: 04/21/2019

Problem 1:
Consider the geometry shown in the figure. The ***P*** geometry is made out of a homogeneous material with $E = 7 \times 10^{10}$ Pa, Poisson's ratio $\nu = 0.3$, and density $\rho = 2000$ kg/m^3. The thickness of the structure is $h = 0.2$m. This is a plane stress problem. The bottom of the structure, $y = 0.0$m, is fixed, $u_x, u_y = 0$. The top of the geometry, $y = 1.0$m, has a constant traction boundary condition. The effect of gravity should also be considered. A mesh of the domain is provided.





Determine the displacement contours that satisfy the weak form of the linear elasticity problem and plot the corresponding stress profile. For that you will need to:

- Compute the element stiffness matrix for the constant strain triangle (you can use the Python code from Github)
- Assemble the element stiffness into the global stiffness matrix. This is similar but not exactly the same as the code for the heat transfer problem, in this case there is x and y equations for each node.
- Compute the body source term $\boldsymbol{f}^{e}_{\boldsymbol{b},i} = \int_{\Omega^e} N_i \, \rho \, h \, \boldsymbol{g}$. Note that the corresponding element body force vector has 6 entries, there is 3 shape functions, and then, for each shape function, the body force has x and y contributions. That's why I wrote the gravity $\boldsymbol{g}$ as a vector.
- Assemble the body force for each element into the global body force vector
- Compute the boundary traction force vector for the elements that have an edge on the top boundary.
- Assemble the global boundary force vector
- Partition the system of equations (the mesh is ordered in such a way that the first 15 nodes are the ones on the fixed boundary, this means that the first 30 degrees of freedom are fixed).
- Solve the system of equations to get the displacement field
- **Compute and plot the stress field based on the displacement field (a function in Python is given for this step)**

NOTE: You can check your solution against Abaqus. A finer mesh is also available on Github. There is almost no difference in the code, the only thing you would need to change is the partition because the finer mesh has 51 nodes on the bottom boundary.