INTRODUCTION

The goal of this project is to determine the displacement and stress field associated with the loading of a cross-section of steel in plane strain. As shown in the figure below, the cross-section is fixed on all sides and subjected to an in-plane body force. A body force density (Joules per meter cubed), \( b(x, y) \), is applied to the steel. The body force causes an in-plane deformation of the square section. The steel has a Young’s modulus of 200 GPa and a Poisson ratio of 0.33. The side length of the section is \( L = 10 \text{ cm} \).

APPLICATION

Step 1 (5 points)

State the relevant PDE’s for modeling the displacement of this system. Nondimensionalize these equations and provide an explanation of how you accomplished this.

Step 2 (10 points)

Derive a set of nondimensional, algebraic equations that approximate the solution to these PDE’s. You may certainly draw on the notes provided in class and from you lab website.
Step 3 (10 points)

Construct an iterative procedure whereby these algebraic equations can be solved numerically. Again, you may draw on the notes provided in class and from you lab website.

Step 4 (25 points)

Implement your iterative solution algorithm. Provide a copy of your documented code. The points associated with this section are awarded for the clarity, commenting and logical development of your code. Make sure that all fields and parameters are clearly identified.

Step 5 (20 points)

Check your algorithm using the following procedure. First construct a smooth displacement field (both u and v components) that die out to essentially zero at the boundaries. These can be arbitrary, but an example is shown below.

Now calculate the stress components associated with these fields. Then construct a body force in terms of these stresses using:

\[
\begin{align*}
    b_x &= -\partial_x \sigma_x - \partial_y \tau \\
    b_y &= -\partial_x \tau - \partial_y \sigma_y
\end{align*}
\]  

(1)

Use this body force field in your numerical algorithm to calculate the displacement field. Use a grid with 25x25 interior points. The displacement field that you calculate should be, in principle, the arbitrary field that you constructed. You will be graded on how quantitatively you are able to demonstrate that this check is satisfied. As as aside, can this approach be generalized to check codes that allow for nonzero boundary data? Explain your reasoning.
Step 6 (15 points)

Now construct a body force that will pull all of the mass points in the steel section towards the center of the section. State the body force that you created (in nondimensional units). Apply this body force and solve for the displacement field. Show a plot of the displacement field that indicates that all points are, indeed drawn towards the center of the steel section.

Step 7 (15 points)

Now try to make your algorithm as fast as possible. Use timing commands to determine how fast your code runs the following problem. Start with a displacement field that is uniformly zero. Apply a nondimensional body force that is of unit value in the x-direction and is zero in the y-direction. Use a 50x50 spatial grid of interior points. Set up your code so that it automatically stops when your maximum value of horizontal displacement is equal to $0.029706 + \frac{1}{2} \leq 0.001$.

State how long it took to accomplish this goal. State the clock speed of your computer. Now take the product of calculation time and clockspeed. State this final figure of merit for the efficiency of your code. Also state, for the record, whether you used Fortran, Mathematica etc. to get this number. The people with the fastest codes, for each programming platform, will be announced in class and given a superb award (that may look similar to a physics trading card).