AMS 529/691 Homework 3

Due: Wednesday 10/18

1. (50 points) Use FEniCS to solve the heat equation

\[
\frac{\partial u}{\partial t} = \Delta u + f \quad \text{in } \Omega \times (0, T)
\]

with initial condition

\[ u = u_0 \quad \text{at } t = 0 \]

and boundary conditions

\[ u = u_D \quad \text{on } \partial \Omega \times (0, T). \]

(a) Construct a test problem with the exact solution

\[ u(x, y, t) = e^{-4\pi^2 t} \cos(2\pi x) \cos(2\pi y). \]

Use SymPy to compute \( \partial_t u, -\Delta u \) and \( f \).

(b) Solve the problem with \( T = 0.1 \) and a fixed time step 0.001 using backward Euler in time. Plot the exact solution \( u \), approximate solution \( u_h \), and error \( u - u_h \).

(c) Solve the problem using trapezoid method in time.

(d) Vary the mesh sizes and time steps to verify the convergence rates both in space and time.

2. (50 points) We looked at the hyperelasticity model in class based on a neo-Hookean stored energy model. In this exercise, we will implement a different hyperelasticity model. The governing equations can be expressed as

\[
-\nabla \cdot P = B \quad \text{in } \Omega
\]

\[ u = u_D \quad \text{on } \partial_D \Omega
\]

\[ P \cdot n = T \quad \text{on } \partial_N \Omega,
\]

where \( u \) is displacements, \( P(u) \) is the first Piola-Kirchoff stress tensor, \( B \) is a given body force per unit volume, \( u_D \) is a given boundary displacement, \( n \) is outward normal, and \( T \) is a given boundary traction. To define \( P(u) \), we need

- \( F = I + \nabla u \), the deformation gradient
- \( C = F^T F \), the Cauchy-Green deformation tensor
- \( E = \frac{1}{2}(C - I) \), the Green-Lagrange strain tensor
- \( W = \frac{\lambda}{2}(\text{tr}(E))^2 + \mu \text{tr}(E^2) \), the St. Venant-Kirchoff strain energy density
- \( S_{ij} = \frac{\partial W}{\partial E_{ij}} \), the second Piola-Kirchoff stress tensor

Then, the first Piola-Kirchoff stress tensor \( P(u) \) is

\[ P = FS, \]

Change the hyperelasticity code to implement this model for the test problem used in Jupyter Notebook.