Class Code Project

Due: 2023 Dec. 30

Notice: this assignment will be graded as the class coding project which counts for 10% of the final grade. It is closely related to the final project of this class.

Code development:
Work on the 2D heat equation solver developed in class. Modify the code to enable the following functionality.

a. In our class, we developed a code for the homogeneous Dirichlet boundary condition (i.e., $g(x, y) = 0$) only. Modify the code to account for non-zero Dirichlet boundary data imposed on the whole boundary (i.e., $\partial\Omega = \Gamma_g$).
b. Consider the case of $\partial\Omega = \Gamma_g \cup \Gamma_h$, where

$$\Gamma_g = \{x = 0, y \in (0,1)\} \cup \{x = 1, y \in (0,1)\},$$
$$\Gamma_h = \{y = 0, x \in (0,1)\} \cup \{y = 1, x \in (0,1)\}.$$

   Modify the code to account for the Neumann boundary condition.
c. Modify your code to enable transient analysis of the heat equation using the generalized trapezoidal family of methods.

You will have to create a private repository on github, allowing me and the TA to view your repository. In the code development part, we want to see a complete development history of your code. Do not commit all of your changes in one commit.

You are asked to run your code in the following examples and submit in a written report.

1. Verify the calculation from the code developed at step b by using the manufactured solution method. You need to decide your own manufactured solution. Your designed manufactured solution shall give non-trivial forms of f, g and h (meaning they are non-zero). In doing so, you may justify the correctness of your modification. Report the errors and convergence rates measured in L2 and H1 norms (refer to Homework 5).
2. It is known that the transient analysis is unconditionally stable if $\alpha \geq 0.5$ and conditionally stable with $\alpha < 0.5$. Let $\alpha = 0$, you get the explicit forward Euler method. The upper bound of the time step size is $2/\lambda_{eq}^h$, where λ_{eq}^h is the largest eigenvalue of the matrix $M^{-1}K$ (see 8.2.1 of the textbook for details). Determine the value of λ_{eq}^h in Matlab and verify this stability condition by your code.
3. It is known that the forward and backward Euler are both first-order accurate, and the mid-point rule is second-order accurate. Can you devise a numerical experiment to verify this claim? (optional and bonus)
4. Simulate the following problem. Let the right and left boundary of the domain be adiabatic, meaning $h = 0$; on the top boundary,

$$g = \begin{cases} t & \text{if } t < 1 \\ 1 & \text{otherwise} \end{cases};$$

   on the bottom boundary $g = 0$; the initial temperature is zero. Additionally, let the heat source be given as

$$f = \begin{cases} 1 & \text{if } \sqrt{(x-0.5)^2 + (y-0.5)^2} < 0.05 \\ 0 & \text{otherwise} \end{cases}.$$

Investigate the temperature distribution over the time period (0,T) with T=10 using your code. Use two meshes and two different time step sizes to make sure your results are mesh-independent. Report your investigation including your numerical settings and the results.