Numerical Analysis Homework 4

Due day: Nov 21, 2025

Solve the heat equation using the rectangular finite element. (This assignment will count toward your course participation and quiz grade.)

Consider the time-dependent heat equation as follows:

$$\frac{\partial u}{\partial t} - \nu \nabla^2 u = f(x, y, t) \quad \text{in} \quad \Omega = (-2, 2)^2, \quad t \in (0, 1],$$

with diffusion coefficient $\nu = 0.05$, and corresponding homogeneous Dirichlet boundary conditions:

$$u(x, y, \cdot) = 0 \quad \text{for } (x, y) \in \partial\Omega.$$

We assume that the exact solution is given as follows:

$$u_{\text{exact}}(x, y, t) = e^{-8\pi^2 \nu t} \sin(2\pi x) \sin(2\pi y).$$

In this project, you will be asked to using the bilinear Q_1 element as your basis function to solve the heat equation. The corresponding four shape functions defined in the reference element $(\xi, \eta) \in [-1, 1] \times [-1, 1]$ are:

$$\begin{aligned} \Phi_1(\xi, \eta) &= \tfrac{1}{4}(1 - \xi)(1 - \eta), & \Phi_2(\xi, \eta) &= \tfrac{1}{4}(1 + \xi)(1 - \eta), \\ \Phi_3(\xi, \eta) &= \tfrac{1}{4}(1 + \xi)(1 + \eta), & \Phi_4(\xi, \eta) &= \tfrac{1}{4}(1 - \xi)(1 + \eta). \end{aligned}$$

(1). By substituting u_{exact} into the PDE, determine the forcing term $f(x, y, t)$ such that

$$\frac{\partial u_{\text{exact}}}{\partial t} - \nu \nabla^2 u_{\text{exact}} = f.$$

(2). Discretize the spatial domain $\Omega = (-2, 2)^2$ into 16 square elements arranged in a 4×4 grid, with the node coordinates are:

$$(x, y) \in \{-2, -1.6, -1.2, -0.8, 2\} \times \{-2, -1.6, -1.2, -0.8, 2\}.$$

Draw this mesh, define your own global node numbering, label all global node numbers, and generate corresponding elemental connectivities.

For instance:

Element 1: (Consists of) (2,4,5,8) nodes...

Element 2: (3,4,5,15)...

(3). For one physical element E_i, write the mapping from the standard element $(\xi, \eta) \in [-1, 1]^2$ to physical coordinates (x, y) in terms of the nodal coordinates (x_n, y_n) and the shape functions $\Phi_n(\xi, \eta)$. Also derive the Jacobian matrix $J(\xi, \eta)$ of this mapping.

Hint:

Assume that $x(\xi,\eta) = \sum_{n=1}^{4} x_n \Phi_n(\xi,\eta), \quad y(\xi,\eta) = \sum_{n=1}^{4} y_n \Phi_n(\xi,\eta),$

(This is called "binary interpolation".) Then, the Jacobian matrix of this mapping is:

$$J(\xi,\eta) = \begin{bmatrix} \frac{\partial x}{\partial \xi} & \frac{\partial x}{\partial \eta} \\ \frac{\partial y}{\partial \xi} & \frac{\partial y}{\partial \eta} \end{bmatrix}.$$

(4). Using the Q_1 basis on the reference element, compute the following derivatives:

$$\frac{\partial \Phi_i}{\partial \xi}, \quad \frac{\partial \Phi_i}{\partial \eta}, \quad i = 1, 2, 3, 4.$$

Then express the physical gradients $\nabla \Phi_i = \left[\frac{\partial \Phi_i}{\partial x}, \frac{\partial \Phi_i}{\partial y}\right]^T$ using Jacobian.

(5). Using the following formulas for the elemental mass matrix M_e and elemental stiffness matrix K_e:

$$(M_e)_{ij} = \int_{\Omega_e} \Phi_i \Phi_j \, d\Omega, \qquad (K_e)_{ij} = \nu \int_{\Omega_e} \nabla \Phi_i \cdot \nabla \Phi_j \, d\Omega.$$

to evaluate these integrals explicitly for an abitrary square element in this mesh. Your M_e and K_e should be matrices of size 4×4.

Hint: For the mass matrix in the standard element $[-1, 1]^2$, the integral is calculated as follows:

$$(M_e)_{ij} = \int_{\Omega_e} \Phi_i \Phi_j d\Omega = \int_{-1}^{1} \int_{-1}^{1} \Phi_i(\xi,\eta) \Phi_j(\xi,\eta) d\xi d\eta.$$

The stiffness matrix is similar.

(6). Assemble the global mass matrix M and global stiffness matrix K for the entire 2×2 mesh using the connectivity derived in part (b). Impose homogeneous Dirichlet boundary conditions on all boundary nodes. Write the resulting discretized ODE system in the following format:

$$M \frac{dU}{dt} + KU = F(t).$$

You can choose any 3 of the approaches we mentioned in class for boundary implementation.

(7). Using the following initial condition

$$u(x, y, 0) = \sin(2\pi x) \sin(2\pi y),$$

to solve the reduced ODE system from $t = 0$ to $t = 1$. Write the matlab/python code.

(8). Write your own solver for solving the linear system. You can freely choose the methods from Gaussian, Jacobi, or Gauss-seidal.

(9). Performance the convergence study with different refinements on time steps. Plot the log-log plot of error vs time step size.

(10). Plot $U(t)$ at T=1.