0 Instructions

This problem set is **due on Tuesday, April 23 by midnight**. Solutions can be submitted on Canvas or by email (to cmwa@umich.edu). Include the names of everyone you worked with on this problem set. Include any code you used to solve the problems as part of your submission.

1 The 1D heat equation

Partial differential equations are differential equations involving functions of multiple variables, and we will focus on the 1D heat equation

$$\frac{\partial u}{\partial t}(x,t) = \alpha^2 \frac{\partial^2 u}{\partial x^2}(x,t) \qquad 0 \leq x \leq L \qquad t > 0$$

where the function $u(x,t)$ is a function depending on one spatial coordinate $0 \leq x \leq L$ and one temporal coordinate $t > 0$. In order to have a well-defined problem, we further require

$$u(0,t) = u(L,t) = 0 \qquad t > 0$$

$$u(x,0) = f(x) \qquad 0 < x < L$$

for some function $f(x)$ specified as part of the data. The first condition says the value of the solution is clamped to 0 at the ends $x = 0$ and $x = L$ and the second condition says that the initial value along the entire length is fixed at the starting time $t = 0$.

1.1 Finite difference methods

One approach to partial differential equations is to use *finite difference methods*. The basic idea is to use *forward, backward, and centered differences* (and even more complicated ideas) to approximate the differential equation and compute updates for each timestep. In our problem, we'll use the *mesh points* $(x_i = ih, t_j = jk)$ where h is the step size for x and k is the step size for t. Let $m = \frac{L}{h}$ denote the total number of spatial steps.

Problem 1.1. Derive the forward difference formula

$$\frac{\partial u}{\partial t}(x_i, t_j) = \frac{u(x_i, t_{j+1}) - u(x_i, t_j)}{k} - \frac{k}{2}\frac{\partial^2 u}{\partial t^2}(x_i, \mu_j)$$

where $\mu_j \in (t_j, t_{j+1})$ is some unknown point, using Taylor expansion with error term of $u(x_i, t)$ around $t = t_j$ and evaluating at $t = t_{j+1}$.

Problem 1.2. Derive the backward difference formula

$$\frac{\partial u}{\partial t}(x_i, t_j) = \frac{u(x_i, t_j) - u(x_i, t_{j-1})}{k} + \frac{k}{2}\frac{\partial^2 u}{\partial t^2}(x_i, \mu_j)$$

where $\mu_j \in (t_{j-1}, t_j)$ is some unknown point, using Taylor expansion with error term of $u(x_i, t)$ around $t = t_j$ and evaluating at $t = t_{j-1}$.

Problem 1.3. Derive the centered difference formula

$$\frac{\partial^2 u}{\partial x^2}(x_i, t_j) = \frac{u(x_{i+1}, t_j) - 2u(x_i, t_j) + u(x_{i-1}, t_j)}{h^2} - \frac{h^2}{12}\frac{\partial^4 u}{\partial x^4}(\xi_i, t_j)$$

where $\xi_i \in (x_{i-1}, x_{i+1})$ is some unknown point, using the Taylor expansion with error term of $u(x, t_j)$ around $x = x_i$ with one evaluation at x_{i+1} and another evaluation at x_{i-1}.

1.2 FTCS

FTCS stands for "forward time, centered space" meaning we will use the forward difference formula for the time variable t and the centered difference formula for the space variable x. To simplify notation, let $w_{i,j} \approx u(x_i, t_j)$ denote our approximations. Then we approximate

$$\frac{\partial u}{\partial t}(x, t) = \alpha^2 \frac{\partial^2 u}{\partial x^2}(x, t)$$

with the finite difference equation

$$\frac{w_{i,j+1} - w_{ij}}{k} = \alpha^2 \frac{w_{i+1,j} - 2w_{ij} + w_{i-1,j}}{h^2}.$$

Problem 1.4. Solve the finite difference equation

$$\frac{w_{i,j+1} - w_{i,j}}{k} = \alpha^2 \frac{w_{i+1,j} - 2w_{i,j} + w_{i-1,j}}{h^2}.$$

for $w_{i,j+1}$ in terms of $w_{i,j}$ and $w_{i-1,j}$.

Problem 1.5. Use the previous problem (along with the conditions $w_{0,j} = w_{m,j} = 0$) to express each update step of the FTCS scheme as a matrix multiplication operation for some matrix A and some starting vector $(w_{1,0}, w_{2,0}, \ldots, w_{m-1,0})$.

$$\begin{pmatrix} w_{1,j+1} \\ w_{2,j+1} \\ \vdots \\ w_{m-1,j+1} \end{pmatrix} = A \begin{pmatrix} w_{1,j} \\ w_{2,j} \\ \vdots \\ w_{m-1,j} \end{pmatrix}$$

Because you can compute updates directly once you know A, the FTCS scheme is called explicit. One consideration in solving these equations is whether the method is *stable* with respect to roundoff errors in the initial data. If we consider u_0 as the exact initial conditions and w_0 as the approximations with $e_0 = w_0 - u_0$ the roundoff errors, then our iterations will produce

$$w_i = A^i w_0 = A^i(u_0 + e_0).$$

Therefore, our method will be stable with respect to errors in the initial data if and only if $A^i e_0 \xrightarrow{i\to\infty} 0$. Recall that this will only happen when the spectral radius $\rho(A) < 1$. In general, it can be fairly difficult to compute the eigenvalues of a matrix. However, the A you should have found above is both *tridiagonal* and *Toeplitz*. This means that the eigenvalues of your matrix are given by the formulas

$$1 - 4\frac{\alpha^2 k}{h^2} \cos\left(\frac{i\pi}{2m}\right)^2.$$

Problem 1.6. Find the condition(s) on h and k such that the eigenvalues above have absolute value bounded above by 1.