University of California San Diego

MAE290B: Numerical Methods for Differential Equations

Final Project Report

March 2022

Author	*Student ID*
Daibo Zhang	A13591601

Contents

(0) Preliminaries

Notation

In this report, I will denote the running index for spatial grid points in the x direction as with the subscript i and the y direction with j. Time points are denoted with the superscript $[n]$. Therefore, the value of function T on the i, j-th spatial grid point and the n-th time point is noted as $T_{i,j}^{[n]}$.

Bold capital letters will be used to name matrices, bold lower case letters will be used to name vectors, and unbold letters will be used to represent scalar-valued quantities.

I will write the first order partial derivative of function f with respect to x as $\partial_x f$ and the k-th order partial derivative as $\partial_x^k f$.

The Alternating Direction Implicit Method

We aim to use the Alternating Direction Implicit (ADI) method to solve the reaction-diffusion equation

$$\partial_t T = \alpha(\partial_x^2 T + \partial_y^2 T) + Q(x, y, t) \tag{1}$$

on a two-dimensional rectangular domain $U = \{(x, y) \in \mathbb{R}^2 \mid L_0 \leq x \leq L_e,\ H_0 \leq y \leq H_e\}$. Homogeneous Dirichlet boundary conditions are imposed. Initial conditions are also provided. The initial condition and the reaction/source term Q are assumed to be at least twice continuously differentiable.

To obtain a finite difference formulation of the problem, we discretize the domain with N grid points in the x direction and M grid points in the y direction with uniform grid size $\Delta x = \Delta y = h$. We mark the grid points with indeces $i = 0, 1, \dots, N-1, N$ and $j = 0, 1, \dots, M-1, M$ so that $x_0 = L_0$, $y_0 = H_0$, $x_N = L_e$, and $y_M = H_e$. Then, for interior grid points, we discretize the spatial derivatives with a second-order central difference scheme. The value of T on each grid point is subsequently assumed to be a function of t alone. Therefore, we obtain the spatially-discretized equation

$$\frac{dT_{i,j}}{dt} = \alpha\left(\frac{T_{i+1,j} - 2T_{i,j} + T_{i-1,j}}{h^2} + \frac{T_{i,j+1} - 2T_{i,j} + T_{i,j-1}}{h^2}\right) + Q(x_i, y_j, t) \tag{2}$$

Next, we advance by half a time step by applying Implicit Euler to the x direction and Explicit Euler to y. The source function Q is split in two parts with Explicit Euler applied to one of them and Implicit to the other. This is possible because the time dependence of Q is explicitly defined here

$$\begin{aligned}\frac{T_{i,j}^{[n+1/2]} - T_{i,j}^{[n]}}{\Delta t/2} = \frac{\alpha}{h^2}\Big(T_{i+1,j}^{[n+1/2]} - 2T_{i,j}^{[n+1/2]} + T_{i-1,j}^{[n+1/2]} + T_{i,j+1}^{[n]} - 2T_{i,j}^{[n]} + T_{i,j-1}^{[n]}\Big)\\ + \frac{1}{2}\Big(Q(x_i, y_j, t^{[n]}) + Q(x_i, y_j, t^{[n+1/2]})\Big)\end{aligned} \tag{3}$$

For simplicity, we write the diffusion number $\alpha\Delta t/2h^2$ as κ and $Q(x_i, y_j, t^{[n]})$ as $Q_{i,j}^{[n]}$, then

$$\begin{aligned}(1+2\kappa)T_{i,j}^{[n+1/2]} - \kappa T_{i+1,j}^{[n+1/2]} - \kappa T_{i-1,j}^{[n+1/2]} = (1-2\kappa)T_{i,j}^{[n]} + \kappa T_{i,j+1}^{[n]} + \kappa T_{i,j-1}^{[n]}\\ + \frac{\Delta t}{4}\Big(Q_{i,j}^{[n]} + Q_{i,j}^{[n+1/2]}\Big)\end{aligned} \tag{4}$$

Next, we advance further by half a time step. This time, we apply Implicit Euler to the y direction and Explicit Euler to x

$$\frac{T_{i,j}^{[n+1]} - T_{i,j}^{[n+1/2]}}{\Delta t/2} = \frac{\alpha}{h^2}\Big(T_{i+1,j}^{[n+1/2]} - 2T_{i,j}^{[n+1/2]} + T_{i-1,j}^{[n+1/2]} + T_{i,j+1}^{[n+1]} - 2T_{i,j}^{[n+1]} + T_{i,j-1}^{[n+1]}\Big) + \frac{1}{2}\Big(Q_{i,j}^{[n+1]} + Q_{i,j}^{[n+1/2]}\Big) \tag{5}$$

Rearrange and obtain

$$(1+2\kappa)T_{i,j}^{[n+1]} - \kappa T_{i,j+1}^{[n+1]} - \kappa T_{i,j-1}^{[n+1]} = (1-2\kappa)T_{i,j}^{[n+1/2]} + \kappa T_{i+1,j}^{[n+1/2]} + \kappa T_{i-1,j}^{[n+1/2]} + \frac{\Delta t}{4}\Big(Q_{i,j}^{[n+1]} + Q_{i,j}^{[n+1/2]}\Big) \tag{6}$$

To write the ADI scheme in vector form, we define $\boldsymbol{\theta}^{[n]},\ \mathbf{q}^{[n]} \in \mathbb{R}^{(N-1)(M-1)}$ as

$$\boldsymbol{\theta}^{[n]} = \begin{pmatrix} T_{1,1}^{[n]} \\ T_{2,1}^{[n]} \\ \vdots \\ T_{M-1,1}^{[n]} \\ T_{1,2}^{[n]} \\ T_{2,2}^{[n]} \\ \vdots \\ T_{M-1,2}^{[n]} \\ \vdots \\ T_{1,N-1}^{[n]} \\ T_{2,N-1}^{[n]} \\ \cdots \\ T_{M-1,N-1}^{[n]} \end{pmatrix},\ \mathbf{q}^{[n]} = \begin{pmatrix} Q_{1,1}^{[n]} \\ Q_{2,1}^{[n]} \\ \vdots \\ Q_{M-1,1}^{[n]} \\ Q_{1,2}^{[n]} \\ Q_{2,2}^{[n]} \\ \vdots \\ Q_{M-1,2}^{[n]} \\ \vdots \\ Q_{1,N-1}^{[n]} \\ Q_{2,N-1}^{[n]} \\ \cdots \\ Q_{M-1,N-1}^{[n]} \end{pmatrix} \tag{7}$$

Then we can write equation (4) and (6) as

$$(\mathbf{I} - \kappa\mathbf{A_{xx}})\boldsymbol{\theta}^{[n+1/2]} = (\mathbf{I} + \kappa\mathbf{A_{yy}})\boldsymbol{\theta}^{[n]} + \frac{\Delta t}{4}\Big(\mathbf{q}^{[n]} + \mathbf{q}^{[n+1/2]}\Big) \tag{8}$$

$$(\mathbf{I} - \kappa\mathbf{A_{yy}})\boldsymbol{\theta}^{[n+1]} = (\mathbf{I} + \kappa\mathbf{A_{xx}})\boldsymbol{\theta}^{[n+1/2]} + \frac{\Delta t}{4}\Big(\mathbf{q}^{[n+1/2]} + \mathbf{q}^{[n+1]}\Big) \tag{9}$$

where $\mathbf{A_{xx}},\ \mathbf{A_{yy}} \in \mathbb{R}^{(N-1)(M-1)\times(N-1)(M-1)}$ are matrices representing the second order central difference apprximation of the second x and y partial derivatives respectively. They are block tridiagonal matrices

$$\mathbf{A_{xx}} = \begin{pmatrix} -2\mathbf{I} & \mathbf{I} & & & & \\ \mathbf{I} & -2\mathbf{I} & \mathbf{I} & & & \\ & \mathbf{I} & -2\mathbf{I} & \mathbf{I} & & \\ & & \ddots & \ddots & \ddots & \\ & & & \mathbf{I} & -2\mathbf{I} & \mathbf{I} \\ & & & & \mathbf{I} & -2\mathbf{I} \end{pmatrix},\ \mathbf{A_{yy}} = \begin{pmatrix} \mathbf{A} & & & & & \\ & \mathbf{A} & & & & \\ & & \mathbf{A} & & & \\ & & & \ddots & & \\ & & & & \mathbf{A} & \\ & & & & & \mathbf{A} \end{pmatrix} \tag{10}$$

Where $\mathbf{A}$ and $\mathbf{I}$ are $(M-1)\times(M-1)$ matrices defined as

$$\mathbf{A} = \begin{pmatrix} -2 & 1 & & & & \\ 1 & -2 & 1 & & & \\ & 1 & -2 & 1 & & \\ & & \ddots & \ddots & \ddots & \\ & & & 1 & -2 & 1 \\ & & & & 1 & -2 \end{pmatrix}, \ \mathbf{I} = \begin{pmatrix} 1 & & & & & \\ & 1 & & & & \\ & & 1 & & & \\ & & & \ddots & & \\ & & & & 1 & \\ & & & & & 1 \end{pmatrix} \tag{11}$$

These spatial FDA matrices can be constructed using Kronecker tensor product. We now return to the the derivation of the scheme and note that Equation (8) implies

$$\boldsymbol{\theta}^{[n+1/2]} = (\mathbf{I} - \kappa\mathbf{A_{xx}})^{-1}(\mathbf{I} + \kappa\mathbf{A_{yy}})\boldsymbol{\theta}^{[n]} + \frac{\Delta t}{4}(\mathbf{I} - \kappa\mathbf{A_{xx}})^{-1}\Big(\mathbf{q}^{[n]} + \mathbf{q}^{[n+1/2]}\Big) \tag{12}$$

Substituting this into Equation (9) gives

$$\begin{aligned}(\mathbf{I} - \kappa\mathbf{A_{yy}})\boldsymbol{\theta}^{[n+1]} &= (\mathbf{I} + \kappa\mathbf{A_{xx}})(\mathbf{I} - \kappa\mathbf{A_{xx}})^{-1}(\mathbf{I} + \kappa\mathbf{A_{yy}})\boldsymbol{\theta}^{[n]} \\ &+ \frac{\Delta t}{4}(\mathbf{I} + \kappa\mathbf{A_{xx}})(\mathbf{I} - \kappa\mathbf{A_{xx}})^{-1}\Big(\mathbf{q}^{[n]} + \mathbf{q}^{[n+1/2]}\Big) \\ &+ \frac{\Delta t}{4}\Big(\mathbf{q}^{[n+1/2]} + \mathbf{q}^{[n+1]}\Big)\end{aligned} \tag{13}$$

Since the matrices $\mathbf{I} \pm \kappa\mathbf{A_{xx}}$ and $\mathbf{I} \pm \kappa\mathbf{A_{yy}}$ represent partial derivatives and T is sufficiently smooth, these matrix products are commutable. Therefore, we have

$$\begin{aligned}(\mathbf{I} - \kappa\mathbf{A_{xx}})(\mathbf{I} - \kappa\mathbf{A_{yy}})\boldsymbol{\theta}^{[n+1]} &= (\mathbf{I} + \kappa\mathbf{A_{xx}})(\mathbf{I} + \kappa\mathbf{A_{yy}})\boldsymbol{\theta}^{[n]} \\ &+ \frac{\Delta t}{4}(\mathbf{I} + \kappa\mathbf{A_{xx}})\Big(\mathbf{q}^{[n]} + \mathbf{q}^{[n+1/2]}\Big) \\ &+ \frac{\Delta t}{4}(\mathbf{I} - \kappa\mathbf{A_{xx}})\Big(\mathbf{q}^{[n+1/2]} + \mathbf{q}^{[n+1]}\Big)\end{aligned} \tag{14}$$

Of note, this equation represent a scheme where the x spatial derivative is used in implicit Euler for the advancement of the first half step. To avoid the build up of truncation error in one direction, we alternate between whether x or y derivative is used for Implicit Euler first every step. Here, I will use Equation (14) to obtain $\boldsymbol{\theta}^{[n+1]}$ from $\boldsymbol{\theta}^{[n]}$ when n is even. In the case where k is odd, we obtain $\boldsymbol{\theta}^{[k+1]}$ with

$$\begin{aligned}(\mathbf{I} - \kappa\mathbf{A_{xx}})(\mathbf{I} - \kappa\mathbf{A_{yy}})\boldsymbol{\theta}^{[k+1]} &= (\mathbf{I} + \kappa\mathbf{A_{xx}})(\mathbf{I} + \kappa\mathbf{A_{yy}})\boldsymbol{\theta}^{[k]} \\ &+ \frac{\Delta t}{4}(\mathbf{I} + \kappa\mathbf{A_{yy}})\Big(\mathbf{q}^{[k]} + \mathbf{q}^{[k+1/2]}\Big) \\ &+ \frac{\Delta t}{4}(\mathbf{I} - \kappa\mathbf{A_{yy}})\Big(\mathbf{q}^{[k+1/2]} + \mathbf{q}^{[k+1]}\Big)\end{aligned} \tag{15}$$

(a) Theoretical Analysis

Our objective is to show that the ADI method is consistent when applied to Equation (1) with $Q = 0$ and that is it $\mathcal{O}(\Delta t^2, h^2, h^2)$ accurate in t, x, y respectively. We will proceed by finding the modified differential equation associated with the ADI scheme. From this point on, n is assumed to be even. The argument for odd n is analogous. First, getting $T_{i,j}^{[n]}$ from $T_{i,j}^{[n-1/2]}$ involves the following equation

$$\frac{T_{i,j}^{[n]} - T_{i,j}^{[n-1/2]}}{\Delta t/2} = \frac{\alpha}{h^2}\left(T_{i+1,j}^{[n]} - 2T_{i,j}^{[n]} + T_{i-1,j}^{[n]} + T_{i,j+1}^{[n-1/2]} - 2T_{i,j}^{[n-1/2]} + T_{i,j-1}^{[n-1/2]}\right) \tag{16}$$

summing Equation (3) and (16) and multipying the resulting equation on both sides by $\Delta t/2$ lead to

$$\begin{aligned} T_{i,j}^{[n+1/2]} - T_{i,j}^{[n-1/2]} &= \kappa\left(T_{i+1,j}^{[n+1/2]} - 2T_{i,j}^{[n+1/2]} + T_{i-1,j}^{[n+1/2]}\right) + \kappa\left(T_{i+1,j}^{[n]} - 2T_{i,j}^{[n]} + T_{i-1,j}^{[n]}\right) \\ &+\kappa\left(T_{i,j+1}^{[n]} - 2T_{i,j}^{[n]} + T_{i,j-1}^{[n]}\right) + \kappa\left(T_{i,j+1}^{[n-1/2]} - 2T_{i,j}^{[n-1/2]} + T_{i,j-1}^{[n-1/2]}\right) \end{aligned} \tag{17}$$

The modified differential equation can then be obtained by first considering a Taylor Series expansion of every term of Equation (17) around $T_{i,j}^{[n}$. We begin with the left hand side

$$\begin{aligned} T_{i,j}^{[n+1/2]} - T_{i,j}^{[n-1/2]} &= T_{i,j}^{[n]} + \frac{1}{2}\Delta t \partial_t T_{i,j}^{[n]} + \frac{1}{8}\Delta t^2 \partial_t^2 T_{i,j}^{[n]} + \frac{1}{48}\Delta t^3 \partial_t^3 T_{i,j}^{[n]} \\ &\quad - T_{i,j}^{[n]} + \frac{1}{2}\Delta t \partial_t T_{i,j}^{[n]} - \frac{1}{8}\Delta t^2 \partial_t^2 T_{i,j}^{[n]} + \frac{1}{48}\Delta t^3 \partial_t^3 T_{i,j}^{[n]} + \dots \\ &= \Delta t \partial_t T_{i,j}^{[n]} + \frac{1}{24}\Delta t^3 \partial_t^3 T_{i,j}^{[n]} + \mathcal{O}(\Delta t^5) \end{aligned} \tag{18}$$

For the right hand side, among the three terms of each parenthetical group, the t direction terms in their Taylor Series expansion cancels one another out. Therefore, only the x and y directional expansion matter. We group terms with changes in the x or y directions together

$$\begin{aligned} &\kappa\left(T_{i+1,j}^{[n+1/2]} - 2T_{i,j}^{[n+1/2]} + T_{i-1,j}^{[n+1/2]}\right) + \kappa\left(T_{i+1,j}^{[n]} - 2T_{i,j}^{[n]} + T_{i-1,j}^{[n]}\right) \\ &\qquad = 2\kappa\Big(T_{i,j}^{[n]} + h\partial_x T_{i,j}^{[n]} + \frac{1}{2}h^2\partial_x^2 T_{i,j}^{[n]} + \frac{1}{6}h^3\partial_t^3 + \frac{1}{24}h^4\partial_x^4 T_{i,j}^{[n]} \\ &\qquad\quad + T_{i,j}^{[n]} - h\partial_x T_{i,j}^{[n]} + \frac{1}{2}h^2\partial_x^2 T_{i,j}^{[n]} - \frac{1}{6}h^3\partial_t^3 + \frac{1}{24}h^4\partial_x^4 T_{i,j}^{[n]} - 2T_{i,j}^{[n]} + \dots\Big) \\ &\qquad = 2\kappa\left(h^2\partial_x^2 T_{i,j}^{[n]} + \frac{1}{12}h^4\partial_x^4 T_{i,j}^{[n]} + \mathcal{O}(h^6)\right) \\ &\kappa\left(T_{i,j+1}^{[n]} - 2T_{i,j}^{[n]} + T_{i,j-1}^{[n]}\right) + \kappa\left(T_{i,j+1}^{[n-1/2]} - 2T_{i,j}^{[n-1/2]} + T_{i,j-1}^{[n-1/2]}\right) \\ &\qquad = 2\kappa\left(h^2\partial_y^2 T_{i,j}^{[n]} + \frac{1}{12}h^4\partial_y^4 T_{i,j}^{[n]} + \mathcal{O}(h^6)\right) \end{aligned} \tag{19}$$

Putting all of these together yields

$$\begin{aligned} \Delta t \partial_t T_{i,j}^{[n]} + \frac{1}{24}\Delta t^3 \partial_t^3 T_{i,j}^{[n]} &= 2\kappa\left(h^2\partial_x^2 T_{i,j}^{[n]} + \frac{1}{12}h^4\partial_x^4 T_{i,j}^{[n]} + h^2\partial_y^2 T_{i,j}^{[n]} + \frac{1}{12}h^4\partial_y^4 T_{i,j}^{[n]}\right) + \dots \\ \partial_t T_{i,j}^{[n]} + \frac{1}{24}\Delta t^2 \partial_t^3 T_{i,j}^{[n]} &= \alpha\left(\partial_x^2 T_{i,j}^{[n]} + \partial_y^2 T_{i,j}^{[n]} + \frac{1}{12}h^2\partial_x^4 T_{i,j}^{[n]} + \frac{1}{12}h^2\partial_y^4 T_{i,j}^{[n]}\right) + \dots \\ \partial_t T_{i,j}^{[n]} - \alpha\left(\partial_x^2 T_{i,j}^{[n]} + \partial_y^2 T_{i,j}^{[n]}\right) &= -\frac{1}{24}\Delta t^2 \partial_t^3 T_{i,j}^{[n]} + \frac{\alpha}{12}h^2\partial_x^4 T_{i,j}^{[n]} + \frac{\alpha}{12}h^2\partial_y^4 T_{i,j}^{[n]} + \dots \end{aligned} \tag{20}$$

In the limit $\Delta t \to 0$ and $h \to 0$, Equation (20) is equivalent to the homogeneous heat equation we set off to solve, indicating that the ADI method is consistent. Furthermore, this analysis also reveals the leading order truncation error $-1/24\Delta t^2\partial_t^3 T_{i,j}^{[n]} + 1/12h^2\partial_x^4 T_{i,j}^{[n]} + 1/12h^2\partial_y^4 T_{i,j}^{[n]}$, which shows that the ADI method is $\mathcal{O}(\Delta t^2, h^2, h^2)$ accurate in t, x, and y respectively.

(b) Implementation

We want to numerically solve the reaction-diffusion equation given by Equation (1) with $\alpha = 0.1$. The source term is given as

$$Q(x, y, t) = 2.5 \sin(4\pi x) \sin(8\pi y) \big(1 - e^{-2t} \sin(50t) \cos(100t)\big) \tag{21}$$

The temporal dependence decays exponentially, so a finite steady state solution is expected. The initial condition is given as

$$T(x, y, 0) = 0.01 \sin(\pi x) \sin(\pi y) \tag{22}$$

Of note, the source term, boundary conditions, and initial condition are all finite and smooth, so the solution $T(x, y, t)$ should also be well behaved.

Strategy

All computational modalities are implemented in MATLAB Version R2017b.

A function `tridiagSolve` is designed to solve linear systems with tridiagonal structure via the Thomas Algorithm. Given $\mathbf{Ax} = \mathbf{b}$ with $\mathbf{A}$ being a tridiagonal sparse matrix, `tridiagSolve` takes in $\mathbf{A}$ and $\mathbf{b}$ and outputs the solution $\mathbf{x}$. To test the function, it is used to solve randomly generated tridiagonal systems. The resulting solutions are compared with the expected ones, and only small errors are incurred.

The `tridiagSolve` function is invoked in `rdeStepADI` to advance one timestep in effort to solve a Reaction-Diffusion Equation using the ADI method. The function `rdeStepADI` solves the tridiagonal systems represented by Equation (14) and (15). It takes the previous timepoint number n as an input and decides which of the two equations to use based on whether n is even or odd. Since the matrices $\mathbf{I} \pm \kappa \mathbf{A_{xx}}$ and $\mathbf{I} \pm \kappa \mathbf{A_{yy}}$ are the same for every timepoint (assuming even timestep size), they are formed outside of `rdeStepADI` and passed into the function as inputs. The tridiagonal matrices $\mathbf{A_{xx}}$ and $\mathbf{A_{yy}}$ are created by helper functions `fdaMatX` and `fdaMatY` using Kronecker tensor product. A function handle describing the time dependence of the source term is also passed into `rdeStepADI`, and the source vectors $\mathbf{q}$ are computed at each timepoint.

The solution obtianed at each timestep is a vector. The vector is reshaped into a matrix with each entry corresponding to the solution $T(x, y, t)$ at each interior spatial grid point. The homogeneous Dirichlet boundary condition is added to the matrix as "a ring of zeros." We expect that the sequence of solutions at all timepoint to converge to the steady state solution, so the sequence is Cauchy and consecutive solutions should eventually become closer together. Thus, we assume that the system has reached a steady state when when the difference between the matrix-form solutions at the current and previous timepoint has a Frobenius norm less than 10^{-5}. We use Frobenius norm here to capture the cumulative differences at every grid point.

On Timestep Size

For a two dimensional problem, the ADI method is unconditionally stable, so timestep size selection mostly concerns accuracy. We look to the characteristic timescales of temporal changes. The diffusion term captures concentration changes induced by spatial distribution heterogeneities, and the fastest temporal changes in solution due to diffusion should arise from the term $\sin(4\pi x) \sin(8\pi y)$. On the other hand, from the inherent temporal fluctuations of the source term, we identify a faster timescale arising from the term $\sin(50t) \cos(100t)$ which has a period of $2\pi/150 \approx 0.04$. Knowing that ADI is $\mathcal{O}(\Delta t^2)$ accurate, sampling each period 20 times should be sufficient. Therefore, we set the timestep size to be $\Delta t = 0.002$.

However, we must also consider the fact that the solution is expected to be continuous in t. If the timestep size is too small, then the difference between two consecutive solutions may be smaller than the steady state tolerance even at earlier timepoints. To avoid this "false steady state" solution, our tolerance also need to be concomitantly small. I hypothesize here that our combination of a 0.002 timestep size and 10^{-5} steady state fluctuation tolerance should effectively circumvent this situation.

Results of Part (b)

All computational steps were carried out on a 2017 MacBook Air with a dual core Intel Core i5 CPU and 8 GBs of RAM. The laptop was running on a macOS Catalina (Version 10.15.7) operating system. A total of 1358 steps were required to reach the steady state defined above. Each step takes approximately 0.30 second, and a total computational time of 6.8 minutes were recorded.

The time evolution of temperature at the point $(x, y) = (0.55, 0.45)$ is plotted in Figure 1.

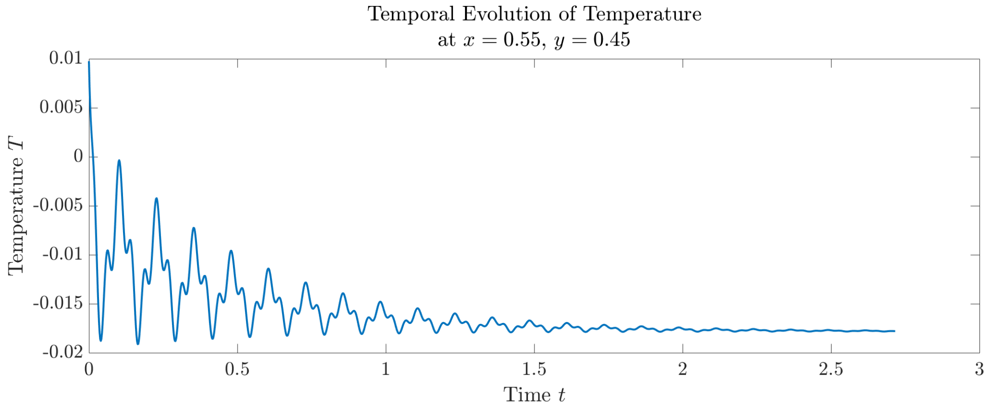

Figure 1: Temperature evolution at the point (0.55,0.45) with initial temperature distribution given by Equation (22)

It is evident that the system reached steady state after around 2.7 units time had elapsed. Recall the time dependence of source strength is described by $e^{-2t}\sin(50t)\cos(100t)$. When $t = 0$, this term is of order 1. Then, the size of this term decreases exponentially so that it is three orders of magnitude smaller after 2 units time has passed. As the time dependence of temperature variation becomes vanishingly small, a steady state temperature distribution is obtained, given by the contour plot in Figure 2

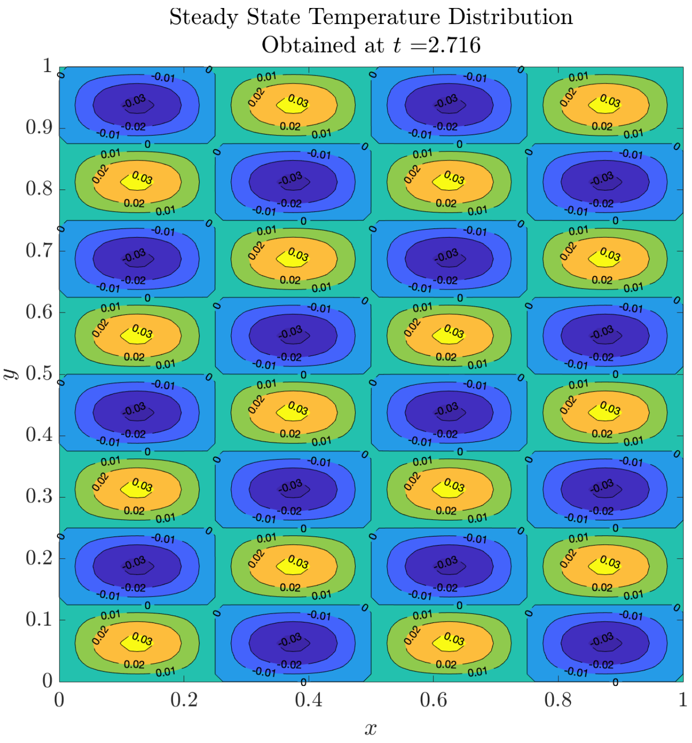

Figure 2: Steady state temperature distribution with with initial temperature distribution given by Equation (22)

(c) *A Posteriori* Error Analysis

Determining K

We find the steady state heat source strength by taking the limit as t approaches infinity in Equation (21)

$$Q_e(x,y) = \lim_{t\to\infty} Q(x,y,t) = 2.5\sin(4\pi x)\sin(8\pi y) \tag{23}$$

Thus, at steady state, the dynamics of temperature distribution is given as

$$0 = \alpha\big(\partial_x^2 T_e + \partial_y^2 T_e\big) + 2.5\sin(4\pi x)\sin(8\pi y) \tag{24}$$

Substituting $T_e = K\sin(4\pi x)\sin(8\pi y)$ into Equation (24) gives

$$\begin{aligned} 0 &= \alpha K\big(-16\pi^2\sin(4\pi x)\sin(8\pi y) - 64\pi^2\sin(4\pi x)\sin(8\pi y)\big) + 2.5\sin(4\pi x)\sin(8\pi y) \\ &= (-80\pi^2\alpha K + 2.5)\sin(4\pi x)\sin(8\pi y) \end{aligned} \tag{25}$$

Therefore

$$K = \frac{1}{32\pi^2\alpha} = \frac{1}{3.2\pi^2} \tag{26}$$

and the analytical steady solution is plotted below

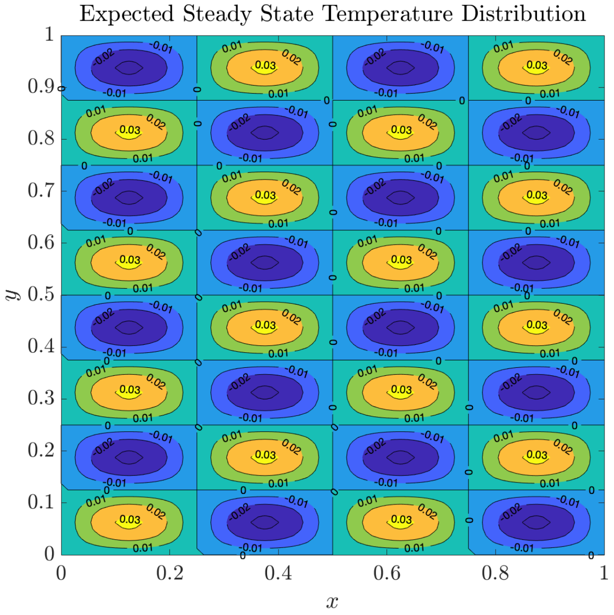

Figure 3: Expected steady state temperature distribution of the heat transfer process described by Equation (1) and (21)

Notably, this steady state temperature distribution is to be expected regardless of initial condition.

Error Analysis

We compare our results from Part (b) with $T_e = \sin(4\pi x)\sin(8\pi y)/3.2\pi^2$. At the point $(0.55, 0.45)$, the error is about -0.51%. The error distribution is plotted as a heatmap in Figure 4.

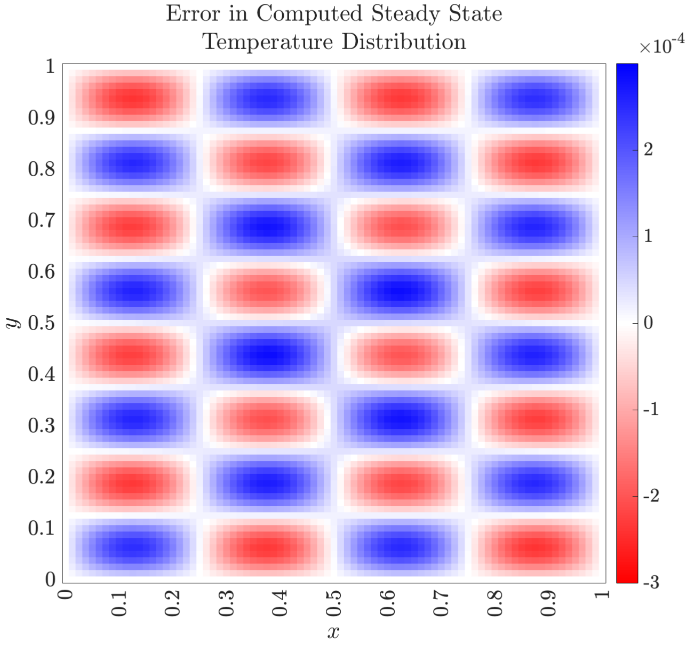

Figure 4: Errors in computed steady state solution obtained with the initial condition given by Equation (22).

Over the entire domain, error is very small at about two orders of magnitudes smaller than the expected value. Furthermore, by overlapping the steady state solution with the error at steady state, we can see that our computed steady state temperature distribution has higher peaks and lower troughs, indicating that ADI produced higher than expected amplitudes. Additionally, the over-representation of blue squares in the error distribution heatmap shows that there is a slight bias for the computed temperature to be higher than expected. Lastly, amplitude errors in peak values seem to be larger in the interior of the domain while errors in trough values seem larger near the boundaries. There is no prominent sign of phase error.

(d) Zero Initial Condition

With zero initial condition, a total of 1201 steps were required for the system to reach steady state. Each step takes around 0.3 second, and approximately 6 minutes of computational time was required. The evolution of temperature at the point $(0.55, 0.45)$ with both sets of initial conditions are given below

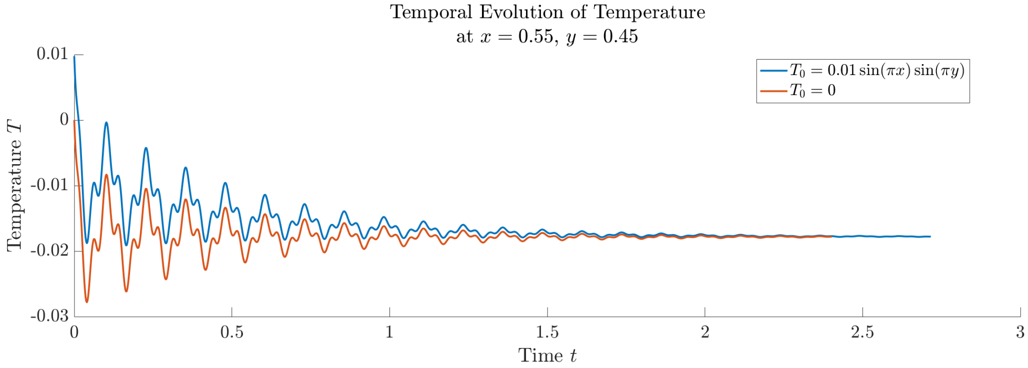

Figure 5: Temperature evolution at the point of interest with both sets of initial conditions.

The temperature distributions at a two early timepoints and near steady state are plotted below

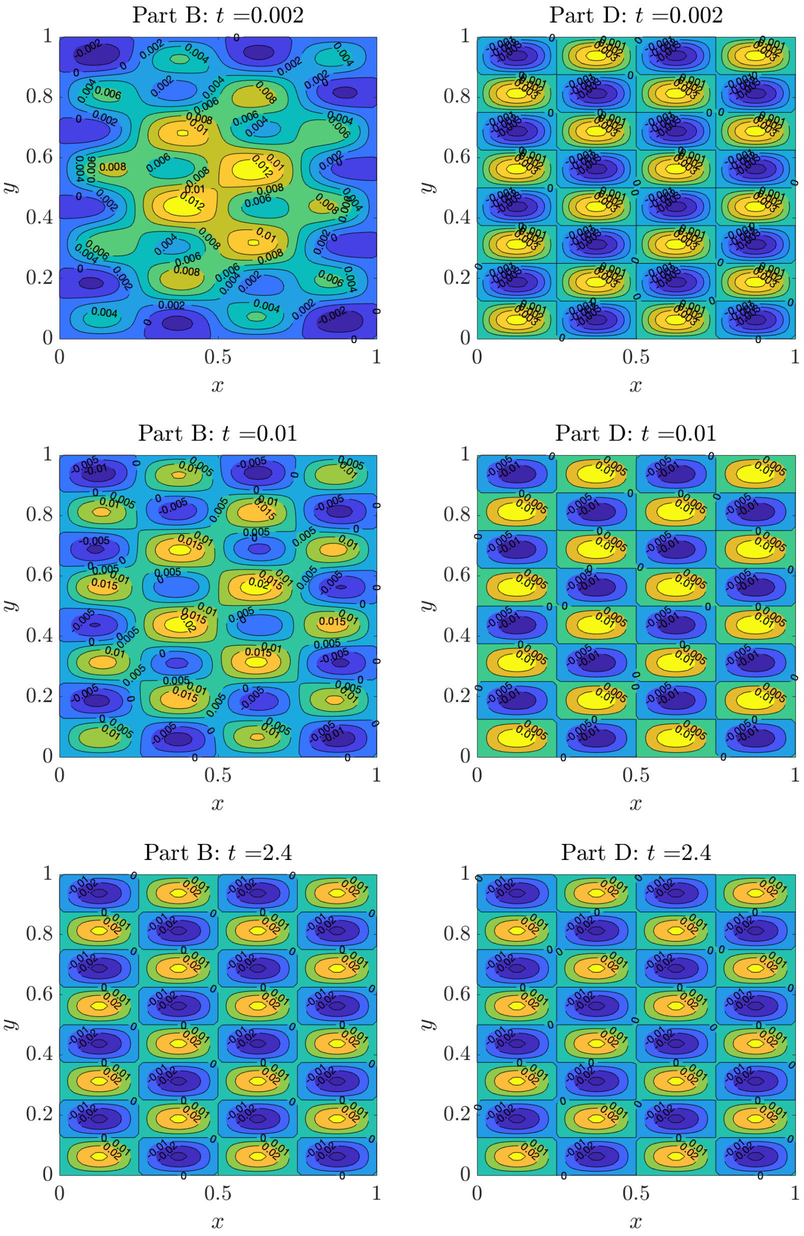

Figure 6: Temperature evolution at early and late timepoints for two different sets of initial conditions.

It can be observed that the system with both sets of initial conditions converges to the same steady state. This is to be expected by the argument in Part (c). The system with initial temperature distribution given in Part (b), Equation (22), takes around 0.2 seconds or 8% longer to converge to steady state. Besides numerical inaccuracies, this discrepancy could be a result of the time required for the non-zero initial heat stored in the system to dissipate away.

The dissipation of initially stored heat is evident in Figure 6. For the system described in Part (d) (zero initial condition), at all three timepoints considered, the temperature distribution pattern reflects the spatial heterogeneity in heat source strength. Since the system is in equilibrium initially, the initial dynamic of its temperature distribution is entirely governed by the heat source. On the other hand, the system described in Part (b) is storing heat internally in the form of a single peak in the middle of the domain. At early timepoints (e.g. when $t = 0.002$), while the effect of the heat source emerges, the temperature distribution is still at the most parts dominated by the initial condition. As time progresses (e.g. when $t = 0.01$), the effect of the initial condition diminishes as heat stored in the system dissipates, and the effect of the heat source becomes increasingly prominent in comparison.

(e) Radiative Cooling

A heat transfer process with radiative cooling can be represented by a nonlinear reaction-diffusion equation

$$\partial_t T = \alpha(\partial_x^2 T + \partial_y^2 T) - \beta T^4 \tag{27}$$

Since the equation is nonlinear, we would want to avoid factoring the nonlinear term into any implicit scheme. I will propose a method of time-integration in which ADI is applied to the linear diffusion term and second-order Runge-Kutta (RK2) is applied to the non-linear radiative cooling term. For even n,

$$\frac{T^{[n+1/2]} - T^{[n]}}{\Delta t/2} = \alpha\Big(\partial_x^2 T^{[n+1/2]} + \partial_y^2 T^{[n]}\Big) - \beta\left(T^{[n]} - \frac{\beta\Delta t}{4}\Big(T^{[n]}\Big)^4\right)^4 \tag{28}$$

Rearranging this equation results in

$$T^{[n+1/2]} - \frac{\alpha\Delta t}{2}\partial_x^2 T^{[n+1/2]} = T^{[n]} - \frac{\alpha\Delta t}{2}\partial_y^2 T^{[n]} - \frac{\beta\Delta t}{2}\left(T^{[n]} - \frac{\beta\Delta t}{4}\Big(T^{[n]}\Big)^4\right)^4 \tag{29}$$

Now, we apply a second order central difference approximation to the spatial derivatives. Let $\boldsymbol{\theta}^{[n]}$, $\mathbf{A_{xx}}$, and $\mathbf{A_{yy}}$ be the same as defined before in Chapter (0), write the diffusion number $\alpha\Delta t/2h^2$ as κ, and define the vector

$$\boldsymbol{\varphi}^{[n]} = \begin{pmatrix} \varphi\big(T_{1,1}^{[n]}\big) \\ \varphi\big(T_{2,1}^{[n]}\big) \\ \vdots \\ \varphi\big(T_{M-1,1}^{[n]}\big) \\ \varphi\big(T_{1,2}^{[n]}\big) \\ \varphi\big(T_{2,2}^{[n]}\big) \\ \vdots \\ \varphi\big(T_{M-1,2}^{[n]}\big) \\ \vdots \\ \varphi\big(T_{1,N-1}^{[n]}\big) \\ \varphi\big(T_{2,N-1}^{[n]}\big) \\ \cdots \\ \varphi\big(T_{M-1,N-1}^{[n]}\big) \end{pmatrix} \tag{30}$$

Where $\varphi : \mathbb{R} \to \mathbb{R}$ comes from applying RK2 to the radiative cooling term, so

$$\varphi(\xi) = -\frac{\beta \Delta t}{2}\left(\xi - \frac{\beta \Delta t}{4}\xi^4\right)^4 \tag{31}$$

Then, we can rewrite Equation (29) as

$$(\mathbf{I} - \kappa \mathbf{A_{xx}})\boldsymbol{\theta}^{[n+1/2]} = (\mathbf{I} + \kappa \mathbf{A_{yy}})\boldsymbol{\theta}^{[n]} + \boldsymbol{\varphi}^{[n]} \tag{32}$$

Since $T^{[n]}$ is know at every position $(\mathbf{I} + \kappa \mathbf{A_{yy}})\boldsymbol{\theta}^{[n]} + \boldsymbol{\varphi}^{[n]}$ can be readily computed, resulting in a tridiagonal linear system. We can then solve the system via the Thomas Algorithm to obtain $\boldsymbol{\theta}^{[n+1/2]}$, apply φ to its components to obtain $\boldsymbol{\varphi}^{[n+1/2]}$, and produce another linear system of equation

$$(\mathbf{I} - \kappa \mathbf{A_{yy}})\boldsymbol{\theta}^{[n+1]} = (\mathbf{I} + \kappa \mathbf{A_{xx}})\boldsymbol{\theta}^{[n+1/2]} + \boldsymbol{\varphi}^{[n+1/2]} \tag{33}$$

Again, we can solve this using the Thomas Algorithm to obtain $\boldsymbol{\theta}^{[n+1]}$. Furthermore, we alternate between whether the x derivative or the y derivative is first in Implicit Euler. As such, for even n, we solve equation (32) and (33) in order to reach $n+1$. For odd n, the equations we solve will be

$$\begin{aligned}(\mathbf{I} - \kappa \mathbf{A_{yy}})\boldsymbol{\theta}^{[n+1/2]} &= (\mathbf{I} + \kappa \mathbf{A_{xx}})\boldsymbol{\theta}^{[n]} + \boldsymbol{\varphi}^{[n]} \\ (\mathbf{I} - \kappa \mathbf{A_{xx}})\boldsymbol{\theta}^{[n+1]} &= (\mathbf{I} + \kappa \mathbf{A_{yy}})\boldsymbol{\theta}^{[n+1/2]} + \boldsymbol{\varphi}^{[n+1/2]}\end{aligned} \tag{34}$$

This scheme should be second order accurate in t. It also should be manageable in terms of computational cost per step since it boils down to solving linear systems of algebraic equations with tridiagonal structures. However, we must note that RK2 is an explicit method that has a stability restriction on Δt. Since the equation is large and nonlinear, it may be difficult to determine an appropriate Δt, and Δt could be impractically small. Thus, it may be beneficial to implement adaptive time-stepping to improve the efficiency of this method.

(f) Alternative Method

Equation (1) is to be solved over a rectangular, thus compact, domain with zero boundary conditions. Assume that the source function Q and initial condition $f(x,y) = T(x,y,0)$ have at most countably many discontinuities, we may apply spectral method to the problem. This allows us to achieve higher accuracy without resorting to exotic finite difference methods that are complicated or computationally expensive.

The domain $U = \{(x,y) \in \mathbb{R}^2 \mid 0 \le x \le 1,\ 0 \le y \le 1\}$ is discretized with 81 grid points with $x_0 = y_0 = 0$ and $x_{80} = y_{80} = 1$. Let p be the running index through x grid points and q be the running index through y grid points. Let $\mathfrak{i}$ denote the unit imaginary number (not to be confused with the i used to track x indeces previously). We write T, Q, and f as their discrete Fourier transform

$$\begin{aligned} T(x,y,t) &\sim \sum_{p=-40}^{39}\sum_{q=-40}^{39} \tau_{p,q}(t) e^{\mathfrak{i}k_p x} e^{\mathfrak{i}k_q y} \\ Q(x,y,t) &\sim \sum_{p=-40}^{39}\sum_{q=-40}^{39} \sigma_{p,q}(t) e^{\mathfrak{i}k_p x} e^{\mathfrak{i}k_q y} \\ f(x,y) &\sim \sum_{p=-40}^{39}\sum_{q=-40}^{39} \eta_{p,q} e^{\mathfrak{i}k_p x} e^{\mathfrak{i}k_q y} \end{aligned} \tag{35}$$

Where wave numbers are given as $k_p = 2\pi p$, $k_q = 2\pi q$. On x and y grid points, equality between the left and right hand sides of these expressions can be established. $\eta_{p,q}$ and $\sigma_{p,q}(t^{[n]})$ can be obtained using Fast Fourier Transform. Substituting these into Equation (1) gives

$$\sum_{p=-40}^{39}\sum_{q=-40}^{39}\frac{d\tau_{p,q}}{dt}e^{\mathrm{i}k_p x}e^{\mathrm{i}k_q y} = -\alpha\sum_{p=-40}^{39}\sum_{q=-40}^{39}\left(k_p^2+k_q^2\right)\tau_{p,q}(t)e^{\mathrm{i}k_p x}e^{\mathrm{i}k_q y} + \sum_{p=-40}^{39}\sum_{q=-40}^{39}\sigma_{p,q}(t)e^{\mathrm{i}k_p x}e^{\mathrm{i}k_q y} \tag{36}$$

Fixing a p and a q, we divide both sides of the equation with $e^{\mathrm{i}k_p x}e^{\mathrm{i}k_q y}$, and obtain a linear ordinary differential equation for each Fourier coefficient $\tau_{p,q}(t)$

$$\frac{d\tau_{p,q}}{dt} = -\alpha\left(k_p^2+k_q^2\right)\tau_{p,q}+\sigma_{p,q} = -4\alpha\pi^2\left(p^2+q^2\right)\tau_{p,q}+\sigma_{p,q} \tag{37}$$

with initial conditions $\tau_{p,q}(0) = \eta_{p,q}$. Moreover, it is worth noting that if Q and f are real-valued functions, T is also expected to be real. By the symmetry of Fourier coefficients for real functions, we only need to solve Equation (37) for $p, q = 0, 1, 2, \ldots, 38, 39$ and invoke symmetry when reconstructing T in the spatial domain. Thus, the decoupled system of linear first-order ODEs can be converted to a single vector equation, which can be solved numerically to obtain $\tau_{p,q}$. We may take advantage of the simplicity of the system to choose a higher-order accurate method such as fourth-order Runge-Kutta or a highly efficient method such as the MATLAB stiff solver `ode23s`. Then, we can substitute the resulting $\tau_{p,q}(t^{[n]})$ back to Equation (35), perform an inverse Fourier Transform, and obtain the value of T on each grid point at each timepoint.

Bibliography

Evans, Lawrence C., *Partial Differential Equations*, American Mathematical Society, 1997.

Lee, William T., "Tridiagonal Matrices: Thomas Algorithm," *MS6021, Scientific Computation, University of Limerick*, 2011.

Moin, Parviz, *Fundamentals of Engineering Numerical Analysis*, Cambridge University Press, 2010.

Strang, Gilbert, *Computational Science and Engineering*, Wellesley-Cambridge Press, 2007.

Wray, Steven, *Alternating Direction Implicity Finite Difference Methods for the Heat Equation on General Domains in Two and Three Dimensions*, Dissertation, Colorado School of Mines, 2016.

Acknowledgements

Prof. Sutanu Sarkar, Jinyuan Liu, Kian Bagheri, Obed Campos, Opal Issan, Josh Krokowski