Homework 8

MATH 6204 (8204) - 090

Fall 2018

100 Points

Reading Assignment: Lectures 9 &10

Due: Thursday, November 15, 11:59pm Student Name: ____________________

Instruction. On the due date, you should email your completed homework with a single zipped folder named “hw8_YourName. The folder should at least include a driver file named “main.py”.

Problem. Consider the following heat equation,

$$\frac{\partial y(x,t)}{\partial t} = \frac{\partial^2 y(x,t)}{\partial x^2}, \tag{1}$$

$$y(x,0) = \sin \pi x, \ \ 0 < x < 1, \tag{2}$$

$$y(0,t) = y(1,t) = 0, \ t > 0, \tag{3}$$

where equations (2)-(3) are the boundary conditions. Using either the explicit scheme or the implicit scheme, one can solve the heat equation numerically. It is known that using each of these finite-difference methods, the heat equation can be discretized into a linear system of difference equations for each time level, as shown below:

- The explicit scheme:

$$w_{j,i+1} = \lambda w_{j-1,i} + (1-2\lambda) w_{j,i} + \lambda w_{j+1,i}.$$

In matrix form,

$$w^{(i+1)} = A_R \cdot w^{(i)}, \ or$$

$$\begin{bmatrix} w_{1,i+1} \\ \vdots \\ \vdots \\ \vdots \\ \vdots \\ w_{N-1,i+1} \end{bmatrix} = \underbrace{\begin{bmatrix} 1-2\lambda & \lambda & 0 & \cdots & \cdots & 0 \\ \lambda & 1-2\lambda & \lambda & \ddots & \ddots & \vdots \\ 0 & \ddots & \ddots & \ddots & \ddots & \vdots \\ \vdots & \ddots & \ddots & \ddots & \ddots & 0 \\ \vdots & \ddots & \ddots & \lambda & 1-2\lambda & \lambda \\ 0 & \cdots & \cdots & 0 & \lambda & 1-2\lambda \end{bmatrix}}_{A_R, \ \ (N-1)\times(N-1)} \begin{bmatrix} w_{1,i} \\ \vdots \\ \vdots \\ \vdots \\ \vdots \\ w_{N-1,i} \end{bmatrix}$$

- The implicit Scheme:

$$\frac{w_{j,i+1} - w_{j,i}}{\triangle t} = \frac{w_{j-1,i+1} - 2w_{j,i+1} + w_{j+1,i+1}}{\triangle x^2},$$

In matrix form,

$$A_L \cdot w^{(i+1)} = w^{(i)},$$

$$\underbrace{\begin{bmatrix} 1+2\lambda & -\lambda & 0 & \cdots & \cdots & 0 \\ -\lambda & 1+2\lambda & -\lambda & \ddots & \ddots & \vdots \\ 0 & \ddots & \ddots & \ddots & \ddots & \vdots \\ \vdots & \ddots & \ddots & \ddots & \ddots & 0 \\ \vdots & \ddots & \ddots & -\lambda & 1+2\lambda & -\lambda \\ 0 & \cdots & \cdots & 0 & -\lambda & 1+2\lambda \end{bmatrix}}_{A_L} \underbrace{\begin{bmatrix} w_{1,i+1} \\ w_{2,i+1} \\ \vdots \\ \vdots \\ w_{N-2,i+1} \\ w_{N-1,i+1} \end{bmatrix}}_{w^{(i+1)}} = \underbrace{\begin{bmatrix} w_{1,i} \\ w_{2,i} \\ \vdots \\ \vdots \\ w_{N-2,i} \\ w_{N-1,i} \end{bmatrix}}_{w^{(i)}}$$

where $\lambda = \frac{\triangle t}{(\triangle x)^2}$, indexes i and j are used to define points in a two-dimensional mesh, $A_R \equiv I + \lambda G$, $A_L \equiv I - \lambda G$. As defined in Lecture 10, G is a symmetric tridiagonal matrix:

$$G = \underbrace{\begin{bmatrix} 2 & -1 & & & 0 \\ -1 & \ddots & \ddots & & \\ & \ddots & \ddots & \ddots & \\ & & \ddots & \ddots & \ddots \\ 0 & & & \ddots & \ddots \end{bmatrix}}_{(N-1)\times(N-1)}.$$

You are asked to do eigenvalue-based stability analysis numerically (not analytically) for both the explicit and implicit schemes. That is, you should do the numerical analysis by calculating the eigenvalues of matrix A_R for the explicit scheme and those of matrix A_L^{-1} for the implicit scheme while providing your comments. Note that these numerical results and comments must be able to show up on the console once your Python is executed. It is required that your numerical results (that is, eigenvalues) be tabulated and plotted to facilitate the analysis.

There are two ways available to calculate eigenvalues for a symmetric matrix:

1. Using the analytical formula on page 14, Lecture 10, or
2. Using *numpy.linalg.eigvals*

You should do both for your numerical analysis, each using four sets of algorithmic parameters as shown below:

- $dx = 0.050$, $dt = 0.0010$, $N = 100$
- $dx = 0.050$, $dt = 0.0015$, $N = 100$
- $dx = 0.040$, $dt = 0.0010$, $N = 100$
- $dx = 0.040$, $dt = 0.0015$, $N = 100$

As mentioned earlier, your calculated eigenvalues must be tabulated and plotted. Certainly, you are free to determine how to display your numerical results. But be sure that your Python program should be well structured and include detailed remarks, including, but not limited to, the homework's purpose, algorithms, and author (i.e., your name) .