Math 228C
Homework 2
Due Friday, 5/3

1. Consider the forward time, centered space discretization

$$\frac{u_j^{n+1} - u_j^n}{\Delta t} + a\frac{u_{j+1}^n - u_{j-1}^n}{2\Delta x} = b\frac{u_{j-1}^n - 2u_j^n + u_{j+1}^n}{\Delta x^2},$$

to the convection-diffusion equation,

$$u_t + au_x = bu_{xx}, \quad b > 0.$$

(a) Let $\nu = a\Delta t/\Delta x$ and $\mu = b\Delta t/\Delta x^2$. Since the solution to the PDE does not grow in time, it seems reasonable to require that the numerical solution not grow in time. Use von Neumann analysis to show that the numerical solution does not grow (in 2-norm) if and only if $\nu^2 \leq 2\mu \leq 1$.

(b) Suppose that we use the mixed implicit-explicit scheme

$$\frac{u_j^{n+1} - u_j^n}{\Delta t} + a\frac{u_{j+1}^n - u_{j-1}^n}{2\Delta x} = b\frac{u_{j-1}^{n+1} - 2u_j^{n+1} + u_{j+1}^{n+1}}{\Delta x^2}.$$

Use von Neumann analysis to derive a stability restriction on the time step.

2. Write programs to solve

$$u_t = \Delta u \text{ on } \Omega = (0,1) \times (0,1)$$
$$u = 0 \text{ on } \partial\Omega$$
$$u(x,y,0) = \exp\left(-100((x-0.3)^2 + (y-0.4)^2)\right)$$

to time $t = 1$ using forward Euler and Crank-Nicolson. For forward Euler, use a time step just below the stability limit. For Crank-Nicolson use a direct solver for sparse systems. At the end of this assignment is MATLAB code to form the matrix for the 2D discrete Laplacian. The code is posted on canvas.

(a) For Crank Nicolson, fix the time step at $\Delta t = 0.01$ and time your codes for different grid sizes and compare the time to solve using forward Euler and Crank Nicolson.

(b) How did the time scale with the grid size in your code?

(c) For this problem we could use an FFT-based Poisson solver which will perform the direct solve in $\mathcal{O}(N\log(N))$, where N is the total number of grid points. We could also use multigrid and perform the solve in $\mathcal{O}(N)$ time. How should the time scale as the grid is refined for Crank-Nicolson if we used an $\mathcal{O}(N)$ solver?

On canvas, I posted an FFT-based solver to experiment with.

3. The motion of a thin elastic filament in a viscous fluid is determined by the balance of the elastic force and the drag force. Assuming inertia is negligible, the drag is proportional to the velocity, and the deflection is small, the vertical displacement, $u(s,t)$, satisfies

$$\xi u_t = -k u_{ssss},$$

where ξ is the drag coefficient and k is the elastic stiffness coefficient. This equation requires two boundary conditions at each end. Consider the case where one end ($s = 0$) is fixed in space but rotated so that the boundary conditions are

$$u(0,t) = 0, \qquad u_s(0,t) = f(t) = \sin(2\pi t).$$

The other end ($s = 1$) is free (no force, no torque), and the corresponding boundary conditions are

$$u_{ss}(1,t) = 0, \qquad u_{sss}(1,t) = 0.$$

Assume the beam is discretized into the equally spaced points $s_j = j\Delta s$ for $j = 0, \ldots, N$ and $\Delta s = 1/N$. A second-order discretization of the fourth-derivative operator that accounts for the boundary conditions is

$$\begin{aligned}
&\frac{7u_1 - 4u_2 + u_3}{\Delta s^4} + \frac{2f(t)}{\Delta s^3}, && \text{for } j = 1 \\
&\frac{u_{j-2} - 4u_{j-1} + 6u_j - 4u_{j+1} + u_{j+2}}{\Delta s^4}, && \text{for } j = 2 \ldots N-2 \\
&\frac{u_{N-3} - 4u_{N-2} + 5u_{N-1} - 2u_N}{\Delta s^4}, && \text{for } j = N-1 \\
&\frac{2u_{N-2} - 4u_{N-1} + 2u_N}{\Delta s^4}, && \text{for } j = N
\end{aligned}$$

(a) Use von Neumann analysis to predict a stability restriction on forward Euler for advancing the system in time.

(b) Numerically compute the eigenvalues of the spatially discrete system for a few different grid sizes to determine the stability restriction for forward Euler for advancing the method in time. How do the results compare with the von Neumann analysis?

(c) Using the parameters $\xi = 1$, $k = 0.05$, and $u(s,0) = 0$ run a simulation using $N = 100$ using forward Euler. How long would it take to generate the solution up to time $t = 10$? You may take a smaller number of time steps, and extrapolate to calculate the time.

(d) Trapezoidal rule time stepping is unconditionally stable for this problem. Using $\Delta t = 0.01$, how long does it take to generate a numerical solution to time $t = 10$ using Trapezoidal rule for time stepping?

(e) Make a few plots of the solution at different time points on the same axes. Note the solution quickly approaches a periodic state.

```
%
% lap2d.m
%
%   form the (scaled) matrix for the 2D Laplacian for Dirichlet boundary
%   conditions on a rectangular node-centered nx by ny grid
%
%   input:  nx -- number of grid points in x-direction (no bdy pts)
%           ny -- number of grid points in y-direction
%
%   output: L2 -- (nx*ny) x (nx*ny) sparse matrix for discrete Laplacian
%
function L2 = lap2d(nx,ny);

    % make 1D Laplacians
    %
    Lx = lap1d(nx);
    Ly = lap1d(ny);

    % make 1D identities
    %
    Ix = speye(nx);
    Iy = speye(ny);

    % form 2D matrix from kron
    %
    L2 = kron(Iy,Lx) + kron(Ly,Ix);

%
% function: lap1d -- form the (scaled) 1D Laplacian for Dirichlet
%                    boundary conditions on a node-centered grid
%
%   input:  n -- number of grid points (no bdy pts)
%
%   output: L -- n x n sparse matrix for discrete Laplacian
%
function L = lap1d(n)
    e = ones(n,1);
    L = spdiags([ e -2*e e], [-1 0 1],n,n);
```