Math 228B
Homework 3
Due Thursday 2/21/19

1. Use Jacobi, Gauss-Seidel, and SOR (with optimal ω) to solve

$$\Delta u = -\exp\left(-(x-0.25)^2 - (y-0.6)^2\right)$$

on the unit square $(0,1)\times(0,1)$ with homogeneous Dirichlet boundary conditions. Find the solution for mesh spacings of $h = 2^{-5}$, 2^{-6}, and 2^{-7}. What tolerance did you use? What stopping criteria did you use? What value of ω did you use? Report the number of iterations it took to reach convergence for each method for each mesh.

2. In this problem we compare the speed of SOR to a direct solve using Gaussian elimination. At the end of this assignment is MATLAB code to form the matrix for the 2D discrete Laplacian. The code for the 3D matrix is similar. Note that with 1 GB of memory, you can handle grids up to about 1000×1000 in 2D and $40 \times 40 \times 40$ in 3D with a direct solve. The range of grids you will explore depends on the amount of memory you have.

   (a) Solve the PDE from problem 1 using a direct solve. Put timing commands in your code and report the time to solve for a range of mesh spacings. Use SOR to solve on the same meshes and report the time and number of iterations. Comment on your results. Note that the timing results depend strongly on your implementation. Comment on the efficiency of your program.

   (b) Repeat the previous part in three spatial dimensions for a range of mesh spacings. Change the right side of the equation to be a three dimensional Gaussian. Comment on your results.

3. The optimal choice of ω for SOR can be computed analytically only for simple problems. It can be approximated for more complex problems. However, the speed of convergence is sensitive to the choice of ω as you will see below.

   (a) Show that for the model problem (2D Poisson equation on the unit square with Dirichlet boundary conditions) the eigenvalues of the SOR update matrix, λ, are related to the eigenvalues of the Jacobi update matrix, μ, by

$$\mu = \frac{\lambda + \omega - 1}{\omega \lambda^{1/2}}$$

   Use the same change-of-variables trick demonstrated in class to relate the eigenvalues of the Gauss-Seidel update matrix to those of the Jacobi matrix.

   (b) It can be shown that the spectral radius of the SOR update matrix can be obtained by solving the above equation when $\mu = \rho_J$ is the spectral radius of the Jacobi update matrix. Specifically, the spectral radius of the SOR update is $\rho_{SOR} = |\lambda|$ where λ satisfies

$$\rho_J = \frac{\lambda + \omega - 1}{\omega \lambda^{1/2}}.$$

   Make a plot of ρ_{SOR} as a function of ω for a few grid sizes to demonstrate the sensitivity to the value of ω.

4. Periodic boundary conditions for the one dimensional Poisson equation on $(0,1)$ are $u(0) = u(1)$ and $u_x(0) = u_x(1)$. These boundary conditions are easy to discretize, but lead to a singular system to solve. For example, using the standard discretization, $x_j = jh$ where $h = 1/(N+1)$, the discrete Laplacian at x_0 is $h^{-2}(u_N - 2u_0 + u_1)$.

   (a) Write the discrete Laplacian for periodic boundary conditions in one dimension as a matrix. Show that this matrix is singular, and find the vectors that span the null space. (Note that this matrix is symmetric, and so you have found the null space of the adjoint).

   (b) What is the discrete solvability condition for the discretized Poisson equation with periodic boundary conditions in one dimension? What is the discrete solvability condition in two dimensions?

   (c) Show that v is in the null space of the matrix A, if and only if v is an eigenvector of the iteration matrix $T = M^{-1}N$ with eigenvalue 1, where $A = M - N$. The iteration will converge if the discrete solvability condition is satisfied provided the other eigenvalues are less than 1 in magnitude (true for Gauss-Seidel and SOR, but not for Jacobi).

```
%
% 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);
```