Applied Numerical Methods I, 3450:427/527, Kreider, Summer 2017

Homework Set 6 (30 points)
Due date: Thursday 20 July

We're going to do several things in this project. First, we'll introduce a special version of Gauss Elimination to use with tridiagonal matrices. The expensive inner loop is replaced by a simple calculation that speeds up the solution process tremendously (line 18 is the row reduction, and line 23 is the back substitution). Mathematicians call this special version the Crout Algorithm, and engineers call it the Thomas Algorithm. Second, we'll get a preview of some topics in Applied Numerical Methods II – specifically, how to approximate derivatives and how to solve linear boundary value problems numerically.

Part 1. The algorithm we will use is tridiag.m, on my web site. It is taken from Numerical Recipes, by WH Press et al, Cambridge University Press, 1992. The parameters are:

AL is the left subdiagonal (remember, AL(1) is not used, as discussed in class)
AM is the main diagonal
AR is the right superdiagonal
r is the right hand side
u is the output vector

To use the algorithm on Au=r, script files typically have this structure:

```
{build x, spatial domain}
{build AL, AM, AR for i=1}
{build AL, AM, AR for i=2,...,N-1}
{build AL, AM, AR for i=N}
{build r}
u = tridiag(AL,AM,AR,r)
plot(x,u)
```

Download the routine from my web site. You don't need to submit anything for Part 1.

Part 2. Whenever you get an algorithm from an outside source, you should test it on a simple problem whose answer you know. Here is a system in augmented matrix form that you can use to test your algorithm. The solution to the system is $u = (1, 1, 1, 1)'$ so it is easy to see whether you're using the function properly.

$$\left(\begin{array}{rrrr|r} 2 & -1 & 0 & 0 & 1 \\ -1 & 2 & -1 & 0 & 0 \\ 0 & -1 & 2 & -1 & 0 \\ 0 & 0 & -1 & 2 & 1 \end{array}\right)$$

Turn in a copy of the script file you used to test the algorithm, and a print out of the results.

Part 3. We are now going to solve a linear boundary value problem. The problem is

$$\begin{aligned} y''(x) + y(x) &= \sin(x^2+1) \\ y(0) &= 2 \\ y(30) &= 5 \end{aligned}$$

Think of $y(x)$ as the steady state temperature of a thin wire that sits in $0 \le x \le 30$. We'll solve for $y(x)$ at a grid of points x_i $(i = 1, \ldots, N)$. The notation is $x_1 = 0$, $x_i = (i-1)h$, $x_N = 30$, with $h = 30/(N-1)$. This will give us a set of unknown function values $y_1, y_2, \ldots, y_N$ at each of the grid points. We'll let $h = .001$, so that $N = 30001$. At each of the interior points $i = 2, \ldots N-1$, we'll need to approximate the equation by discretizing the equations. At the 2 endpoints $i = 1$ and $i = N$, we'll use the boundary conditions: $y_1 = 2$ and $y_N = 5$.

To discretize the equation, we need an approximation for the derivative y''. The standard form developed in 428/528 ANM2 is

$$y''(x) = \frac{y(x-h) - 2y(x) + y(x+h)}{h^2}$$

If we let $x = x_i$, then $x + h = x_{i+1}$ and $x - h = x_{i-1}$, so we get

$$y''(x_i) = \frac{y_{i-1} - 2y_i + y_{i+1}}{h^2}$$

When we discretize the equation at a point x_i, the result is

$$\frac{y_{i-1} - 2y_i + y_{i+1}}{h^2} + y_i = \sin(x_i^2 + 1)$$

We need to combine terms to set this up as a tridiagonal system:

$$\left(\frac{1}{h^2}\right) y_{i-1} + \left(1 - \frac{2}{h^2}\right) y_i + \left(\frac{1}{h^2}\right) y_{i+1} = \sin(x_i^2 + 1)$$

For simplicity, denote the coefficients of y_{i-1}, y_i and y_{i+1} as A, B and C. This equation applies to x_i for $i = 2, \ldots, N-1$, which will form the interior rows of the matrix system. With the left boundary condition in the first row, and the right in the last row, we get a tridiagonal matrix:

$$\begin{bmatrix} 1 & 0 & 0 & 0 & \ldots & 0 \\ A & B & C & 0 & \ldots & 0 \\ 0 & A & B & C & 0 & 0 \\ 0 & 0 & \ddots & \ddots & \ddots & 0 \\ 0 & 0 & \ldots & A & B & C \\ 0 & 0 & \ldots & 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} y_1 \\ y_2 \\ y_3 \\ \vdots \\ y_{N-1} \\ y_N \end{bmatrix} = \begin{bmatrix} 2 \\ \sin(x_2^2+1) \\ \sin(x_3^2+1) \\ \vdots \\ \sin(x_{N-1}^2+1) \\ 5 \end{bmatrix}$$

We'll talk in class about how to build this system in MATLAB and how to use the tridiagonal solver. You will solve the system and plot the solution for Part 3.

Part 4. Change the right boundary condition to $y(30) = 16$ and rerun the code. Submit the new plot (take a moment to compare with the previous plot), but do not submit a second copy of the code.

WHAT TO TURN IN. You do not have to write a summary. Make sure that what you turn in is easily readable, but don't go to extreme lengths to make it look pretty.

Part 2. A copy of the script file you used to test the algorithm, and a printout of the results. (1 or 2 pages)

Part 3. A copy of the script file you used to build and solve the system and plot of the solution. (1 or 2 pages)

Part 4. The updated plot. (1 page)