Project 2 MAT 362

Due Sunday October 15th midnight

The goal of this project is to use and appreciate the importance of tridiagonal matrices. First, we will code, test, and compare three tridiagonal solvers, and then apply one to generate cubic splines. We will a tridiagonal solver to experiment with our favourite tridiagonal matrix $-D_2$, that is, the one discretizing the second derivative operator with zero Dirichlet boundary conditions on [0,1] via the 3-point central second difference formulae. Finally, we will see that the eigenvalues of $-D_2$ approximate solutions to an important 2nd order linear BVP.

I. Code and test an $m \times m$ tridigagonal linear system solver based on Gaussian elimination. Implement in a function that can be called by other functions for various applications, i.e., by passing in $m \times 1$ column vectors $\{L, D, U, b\}$ in some format, and returning the solution x.
   (a) Test the solver on a few small systems. Compare solutions to exact, known answers.
   (b) Test with a very large m. Compare against the built in matlab solver such as '\', using the sparse matrix Matlab command to build the matrix. In Matlab Help, look up 'mldivide', the proper name for '\'. Determine what algorithm Matlab uses when this overloaded operator is called on a sparse tridiagonal matrix system.

II. Repeat some of your experiments by implementing the iterative Gauss-Sidel method for tridiagonal systems.

III. Use a trigiagonal solver to compute a cubic spline for an interesting set of data. After initial testing on small examples, perhaps one for which you know the exact solution, take a more finely digitized piece of a curve from some photo or drawing and compute one or more cubic splines fitting some and/or all of the data points. Plot the splines along with the original graphic.

IV. Let A be our favourite tridiagonal matrix:

$$D_2 = \frac{1}{(dx)^2} \begin{pmatrix} -2 & 1 & & & & \\ 1 & -2 & 1 & & & \\ & \ddots & \ddots & \ddots & & \\ & & \ddots & \ddots & \ddots & \\ & & & 1 & -2 & 1 \\ & & & & 1 & -2 \end{pmatrix} \in M^{(n-1)\times(n-1)},$$

i.e., $(D_2 * \mathbf{y})_i \approx y''(v_i)$

   (a) Let $f(x) = x(1-x)$ define a function on $[0, 1]$. Discretize f into a vector b of length $m = n-1$, discounting endpoints where f satisfies the zero Dirichlet boundary condition. Compute $v = Ab$ and $w = A^{-1}b$, where for the latter a tridiagonal solver is used to solve $Aw = b$. Compare your results to the exact, known computation of $-f''(x)$ and solution to $-y''(x) = f(x)$ with this boundary condition. Present graphical and tabular results indicating the closeness of your fit.
   (b) Repeat the previous experiment with $f(x) = e^x * \sin(\pi x)$.

(c) Look up the error term for the 3-point central second difference formula (Burden and Faires has it). Use the formula to explain your results from IV a) and b).

V. Compute some eigenvalues and eigenvectors of -D2 via a Matlab call like "[V,D]= eigs(-D2, 6, 'sm')". Plot the resulting vectors, comparing them against known exact solutions to $-y'' = \lambda y$ with 0-D BC. Compare the eigenvalues too. How do the 6 biggest eigenvectors look? Investigate convergence to the true, smallest few eigenvalues for larger sizes of D2.

Final Project idea: Repeat Problem I for the D_2 block tridiagonal matrix approximating the Laplacian on the square.