Your Homework Assignment
CMSC/AMSC 661 Spring 2005
Using Sparse Solvers for Linear Systems
Problem 1: Due April 12
Problems 2,3: Due April 19

In this homework, we study problems related to the elliptic operator $-\bigtriangledown \cdot (\bigtriangledown u)$ with Dirichlet boundary conditions.

Problem 1. In this problem, we study the elliptic eigenvalue problem $-\bigtriangledown \cdot (\bigtriangledown u) = \lambda u$ on the square $(-1, 1) \times (-1, 1)$ with zero boundary conditions. The true eigenvalues can be found in Chapter 6, so we can see how well the discrete approximation performs.
1a. (5) Use Pdetool to plot the first 5 eigenfunctions.

- Describe in words the shape of each of these eigenfunctions v_i, $i = 1, \ldots, 5$. How does the shape change as i increases?
- Theory tells us that we have good approximations with a coarse grid only for small i. How does the shape of the eigenfunctions make this result easier to understand?

1b. (10) Create 5 plots, for eigenvalues 1, 6, 11, 16, and 21, of the error in the approximate eigenvalue vs. the square root of the size of the matrix. The horizontal axis should be the square root of the size of the finite element matrix (use at least 4 different matrix sizes, with the largest obtained using at least 2000 triangles). The vertical axis should be the log of the absolute value of the difference between the true eigenvalue and the computed one. (If you find a plotting scheme that is more revealing, you may use it.) Label the plots so that they are easy to understand. Discuss:

- What convergence rate do you observe for each eigenvalue?
- How does it compare with the theoretical convergence rate?
- Explain any discrepancy.
- Are all of the eigenvalues well-approximated by coarse meshes?

Submit: For 1a, submit only your discussion, no programs or plots. For 1b, submit your 5 graphs and your discussion, followed by the listings of your documented programs.

Tools:
For 1a, you might use the Pdetool to compute eigenfunctions. Use the same actions you used before, but

- Under “pde → pdespecification”, choose “eigenmodes”.
- After “solve”, the toolbox displays the eigenfunction corresponding to the smallest eigenvalue. A message on the bottom of the window says how many eigenvalues were computed.
- To see other eigenfunctions, choose “plot → parameters”, find the “eigenvalue” box in the lower right corner, select another one, push “done”, and then push the “plot” button.

For 1b, you may want to make use of `initmesh`, `refinemesh`, `pdeeig`, `squareg`, and `squareb1`.

Now for something completely different – a problem in which we use and evaluate some of our sparse solvers.

Problem 2. (10) You have been hired as a consultant by a start-up company, LaplaceIsUs.com, to advise us on the use of sparse iterative solvers. Our business is to solve the Laplace equation in 2 and 3 dimensions over various domains, and we have developed two types of problems that we believe typical of those that our customers will provide. The first domain is a sector of a circle (like pacman) using an adaptive finite element grid, and the second is a 3-dimensional box discretized using finite differences. Use `slit2.m` and `laplace3d.m` (with $n = 15$) to generate three linear systems. Solve the linear systems using these algorithms:

- Cholesky on the original matrix.
- Cholesky using the reverse Cuthill-McKee ordering (`symrcm`).
- Cholesky using the minimum degree ordering (`symmmd`).
- GMRES, with restart $m = 20$ (`gmres`).
- GMRES, with restart $m = 100$.
- CG (`pcg`).
- CG with diagonal preconditioning.
- CG with incomplete Cholesky preconditioning (`cholinc`).
- CG using the minimum degree ordering, with incomplete Cholesky preconditioning (`cholinc`).

For the iterative methods, start with initial guess of zeros and use `tol` $= .01h^2$. For incomplete Cholesky, use a drop tolerance of .05.
Your Matlab program should make a table reporting, for each method,

- time to solve the system (For Cholesky, include reordering, factorization, forward and back substitution. For the iterative methods, include initialization of the preconditioners and the time taken by the iteration.)
- storage (estimated). Include the storage for any matrix factors and the storage used by the iterative methods. (Document how you made this estimate.)
- number of iterations (1 for the direct methods, otherwise the number of GMRES or CG iterations, or, equivalently, the number of matrix-vector products involving A).
- the final relative residual $\|b - Ax_{computed}\|_2 / \|b\|_2$.

Considering the 2d and 3d problems separately, report to the CEO the performance of the various methods. Make a recommendation of the method-of-choice for the 2-d problems and justify it. For the 3d problem, what are the 4 most promising methods and why?

Problem 3. (15) Take the 4 most promising methods for the 3d problem and study them further. Graph time (vertical axis) vs storage (horizontal axis) for various choices of n, and report to the CEO on your conclusions on what algorithms should be used for various problem sizes and why. (Use a wide range of values of n.) Incomplete Cholesky preconditioning should be one of your 4 methods; recommend a good choice for the drop tolerance and justify your recommendation.

It would be nice to include experiments with multigrid in this homework. Maybe next time I teach the course....

Submission instructions:

- Problem 2: Submit your table, your summary of the performance of the methods, and your recommendations with justification.
- Problem 3: Submit your graph and your recommendation with justification.
- At the **end of your submission**, attach your documented programs for Problems 2 and 3.