Math 224: Scientific Computing

Homework 4: Cholesky factorization and iterative methods

Due: *Thursday, September 25, 2008*

1. (Written) Let $A \in \Re^{n\times n}$ be real, symmetric, and positive definitive. Consider solving $A\vec{x} = \vec{b}$ using Gaussian elimination without pivoting(!). The purpose of this problem is to justify that the pivots will be nonzero.

   (a) Show that all of the diagonal elements satisfy $a_{ii} > 0$. This shows that a_{11} can be used as a pivot element.

   (b) After elimination of a_{1j} for $j = 2, \ldots, n$, let the resulting matrix $A^{(2)}$ be written as

$$A^{(2)} = \begin{bmatrix} a_{11} & a_{12} & \cdot & a_{1n} \\ 0 & & & \\ \vdots & \hat{A}^{(2)} & & \\ 0 & & & \end{bmatrix}$$

   Show that $\hat{A}^{(2)}$ is symmetric and positive definite.

   This approach can be continued inductively to each stage of the elimination process, thus justifying the existence of nonzero pivots at every step. *Hint:* To prove $\hat{A}^{(2)}$ is positive definite, first prove the identity

$$\sum_{i,j=2}^{n} a_{ij}^{(2)} x_i x_j = \sum_{i,j=1}^{n} a_{ij} x_i x_j - a_{11}\left(x_1 + \sum_{j=2}^{n} \frac{a_{j1}}{a_{11}} x_j\right)^2$$

   for any choice of $x_1, x_2, \ldots, x_n$. Then choose x_1 suitably.

2. An X-matrix Let A be an $n \times n$ matrix (n must be an EVEN integer), whose entries are all zeros, except for

$$a_{i,i} = i, \quad a_{i,n-i+1} = i/2, \quad i = 1, 2, \ldots, n$$

   If you "`spy`" this matrix in `matlab`, you will see that the structure of this matrix is an "X." Let $\vec{b} \in \Re^n$ be the vector with all ones, $b_i = 1$ for $i = 1, 2, \ldots, n$. We will consider several approaches to solving $A\vec{x} = \vec{b}$.

   (a) Use the `LUfactor()`, `LUsolve()` routines you wrote earlier to solve this problem. Make some changes to your routines: declare a global integer variable `flops` (flops stands for FLoating-point OPerationS) in your program and each time your LU routine does a floating-point multiplication or division, add one to `flops`. Track the total number of floating point operations needed.

   (b) Write a separate program to solve $A\vec{x} = \vec{b}$ using Jacobi iterations with successive-over-relaxation (SOR) for a given value of the relaxation parameter. Like (a), track the number of `flops` needed. Since A only has $2n$ nonzero entries, calculating the product $A\vec{x}$ should require only $\mathcal{O}(n)$ flops. Write your code to be this efficient. Stop your criterion when $\|\vec{x}_{k+1} - \vec{x}_k\|_\infty < 10^{-12}$.

   (c) Repeat (b) to write a code for Gauss-Seidel iterations. [1]

---

[1] Test (b) and (c) with $n = 6$ and $\omega = 1$; verify that your iteration code is correct; make sure it converges to the solution from (a).

(d) For each of (b) and (c), put your code in a loop over a range of ω values (with ω incrementing by 0.001 or less). Starting from the same initial guess $\vec{x}_0$ (like $\vec{x}_0 = 0$), find the optimal value of ω^* that leads to convergences with the smallest number of iterations. [2]

(e) Save the results (n vs. `flops`) to data files for the following cases:

- LU
- Jacobi (without SOR)
- Gauss-Seidel (without SOR)
- Jacobi-SOR with ω^*
- Gauss-Seidel-SOR with ω^*

for $n = 6, 8, 10, \ldots, 500$. Plot these as curves on a (one) log-log graph.

What is the smallest matrix (n-value) for which iterative solution becomes more computationally efficient than direct solution in this problem?

Extrapolate and make a table predicting the `flops` needed for $n = 1000$ for each method.

---

[2] Test to see if you believe this statement: ω^* is independent of n for this problem.