MATH 578: Numerical Linear Algebra (Homework 4)

Due November 3$^{\text{rd}}$ at $11:59$ pm

Problem 1 For n an even integer let $A = A^T$ denote the following $n \times n$ symmetric, positive-definite sparse matrix with entries

$$A_{i,j} = \begin{cases} 3 & \text{if} \quad j = i, \\ -1 & \text{if} \quad j = i+1 \\ -1 & \text{if} \quad j = i-1 \\ -1 & \text{if} \quad j = n-i+1 \\ 0 & \text{otherwise.} \end{cases} \tag{1}$$

In other words, A has a diagonal band of $3's$, a super-diagonal and sub-diagonal band of $-1's$ and an anti-diagonal band (running from lower-left to top-right) of $-1'$s.

(a) Implement a function

```
function [L,nzl] = Cholesky(A)
```

that computes the Cholesky decomposition for symmetric, positive definite matrices. It should return a lower-triangular matrix L with positive diagonal entries so that $A = LL^T$ as well as an integer nzl that gives the number of non-zero entries in the Cholesky factor. For instance,

$$L = \begin{bmatrix} 1.414213562373095 & 0 & 0 \\ 0.707106781186547 & 1.224744871391589 & 0 \\ 0 & 0.816496580927726 & 1.154700538379251 \end{bmatrix}$$

and $nzl = 5$ when given the matrix

$$A = \begin{bmatrix} 2 & 1 & 0 \\ 1 & 2 & 1 \\ 0 & 1 & 2 \end{bmatrix}$$

Count any entry with $|L_{ij}| > 0$ as a non-zero.

(b) For each value of $n = 10, 20, 100, 200, 1000$ and $n = 2000$, use part (a) to compute the Cholesky decomposition of the corresponding matrix A from (1) above. Use the to results to find an asymptotic estimate for number of nonzero entries $nzl \sim an^2 + bn + c$ in the Cholesky factor L as a function of the problem size. Determine **only** the leading order term. So, if the number of nonzeros scales quadratically then you may report the value of $a > 0$ but ignore the values of b, c. If the number of nonzeros scales linearly then you may report the value of $b > 0$ but ignore the value of c and so forth. Do this experimentally, and so in particular, do **not** attempt to prove anything. The MATLAB command

```
spy(L);
```

may be helpful; it will let you view the pattern of nonzeros in L as an image.

(c) Assume that only nonzero entries of L need to be stored explicitly in memory (so, ignoring the storage of A itself as well as the zero entries in L), that storage of each such nonzero requires 8 bytes and that 8 gigabytes of total memory are available. Use (b) to approximate the largest problem size (i.e. largest value of n) for which Cholesky factorization can be used to solve the system $Ax = b$ with A given by (1) above.

(d) Implement a function

```
function y = Multiply(x)
```

that performs the matrix-multiplication Ax implicitly. That is, simply code the formula

$$\begin{aligned} y_i &= (Ax)_i = 3x_i - x_{i+1} - x_{i-1} - x_{n-i+1} && (i \neq 1, n) \\ y_1 &= (Ax)_1 = 3x_1 - x_2 - x_n && (i = 1) \\ y_n &= (Ax)_n = 3x_n - x_{n-1} - x_1 && (i = n) \end{aligned}$$

rather than forming A explicitly as a matrix.

(e) Implement the Conjugate Gradient algorithm for solving $Ax = b$ using part (d) to perform all Ax operations that are required by the algorithm. Use a stopping condition of

$$\frac{\|Ax - b\|_2}{\|b\|_2} \leq 10^{-6}$$

and keep track of the total number of CG iterations required. Take $b_i = 1/\sqrt{n}$ (i.e. a normalized vector of all ones). For each value of $n = 5,000$, $n = 15,000$ and $n = 30,000$ report the number of CG iterations required for convergence as well as the relative error

$$\frac{\|Ax - b\|_2}{\|b\|_2}$$

of the computed solution at convergence. Does the number of iterations required appear to increase with n, or remain roughly constant with n?

(f) By modifying the multiply function from part (d), repeat part (e) using the matrix (1) with $3.0001'$s on the diagonal instead of $3'$s. Why do the results improve?

(g) The Conjugate Gradient algorithm provides a viable method for solving large-scale systems $Ax = b$ any time the matrix-vector multiplication

$$x \mapsto Ax$$

can be performed efficiently. Suppose A takes the form $A = I + BB^T$ for I the $n \times n$ identity matrix and B the following $n \times 3$ matrix

$$B = \begin{bmatrix} 1 & 0 & 0 \\ 1 & 1 & 0 \\ 1 & 1 & 1 \\ \vdots & \vdots & \vdots \\ 1 & 1 & 1 \end{bmatrix}.$$

Even though A itself is dense, we can still compute Ax quickly via $Ax = x + B(B^T x)$. By modifying the multiply function from part (d), implement the Conjugate Gradient algorithm for solving $Ax = b$. For each $n = 10,000, 50,000$ and $n = 100,000$ report the number of iterations required for convergence. Use

$$\frac{\|Ax - b\|_2}{\|b\|_2} \leq 10^{-12}$$

as a stopping condition and the same right hand side b used in parts (e) and (f). Based on your results, what is the **maximum** number of distinct eigenvalues that A can have, regardless of problem size? (Again, no need to prove anything).