HW2

2020年3月25日 上午 01:17

MATH3322 Matrix Computation
Homework 2

Due date: 29 March, Sunday

1. Let
$$\boldsymbol{A} = \begin{bmatrix} 1 & 2 & 4 \\ 3 & 8 & 14 \\ 2 & 6 & 13 \end{bmatrix}$$
Find the LU decompsotion with partial pivoting of $\boldsymbol{A}$.

2. Instead of the LU decomposition, we can also use a *UL decomposition* to solve the system of linear equations. In particular, given $\boldsymbol{A} \in \mathbb{R}^{n\times n}$, we decompose $\boldsymbol{A} = \boldsymbol{UL}$, where $\boldsymbol{U} \in \mathbb{R}^{n\times n}$ is *unit upper triangular* and $\boldsymbol{U} \in \mathbb{R}^{n\times n}$ is *lower triangular*. Propose an algorithm for computing the UL decomposition of $\boldsymbol{A}$.

3. Let $\boldsymbol{A} \in \mathbb{R}^{n\times n}$ be a tridiagonal matrix, i.e., $a_{ij} = 0$ if $|i-j| > 1$. We also assume that $\boldsymbol{A}$ is symmetric positive definite (SPD).
   (a) Prove that the Cholesky decompostion $\boldsymbol{A} = \boldsymbol{LL}^T$ satisfies $l_{ij} = 0$ for all $i - j > 1$. In other words, $\boldsymbol{L}$ is bi-diagonal.
   (b) Propose an $O(n)$ algorithm for computing the Cholesky decomposition of $\boldsymbol{A}$. What is the number of operations needed of your algorithm? Your answer should be in the form of $Cn + O(1)$ with explicit constant C.
   (c) Based on the Cholesky decomposition, construct an $O(n)$ algorithm to solve $\boldsymbol{Ax} = \boldsymbol{b}$. Express the number of operations needed in the form of $Cn + O(1)$ with explicit C.

4. We consider a discrete 1-D Laplacian equation $\boldsymbol{Ax} = \boldsymbol{b}$, where
$$\boldsymbol{A} = \begin{bmatrix} 2 & -1 & & & \\ -1 & 2 & -1 & & \\ & \ddots & \ddots & \ddots & \\ & & \ddots & \ddots & -1 \\ & & & -1 & 2 \end{bmatrix} \in \mathbb{R}^{n\times n} \qquad \boldsymbol{b} = \begin{bmatrix} 1 \\ 1 \\ \vdots \\ 1 \end{bmatrix} \in \mathbb{R}^n$$
   (a) Prove that $\boldsymbol{A}$ is SPD.
   (b) Since $\boldsymbol{A}$ is also tridiagonal, the algorithms in Question 3 can be applied. Write a Matlab code to implement your algorithm in Question 3(b)(c) for solving $\boldsymbol{Ax} = \boldsymbol{b}$ where A template file `spdtridiagsolver.m` is provided. Plot the solution you obtained with $n = 500$.

1

1.

$$A = \begin{bmatrix} 1 & 2 & 4 \\ 3 & 8 & 14 \\ 2 & 6 & 13 \end{bmatrix}$$

$$A = \begin{bmatrix} 3 & 8 & 14 \\ 1 & 2 & 4 \\ 2 & 6 & 13 \end{bmatrix}, P = \begin{bmatrix} 0 & 1 & 0 \\ 1 & 0 & 0 \\ 0 & 0 & 1 \end{bmatrix}$$

$$A = \begin{bmatrix} 3 & 8 & 14 \\ \frac{1}{3} & 2 & 4 \\ \frac{2}{3} & 6 & 13 \end{bmatrix}$$

$$A = \begin{bmatrix} 3 & 8 & 14 \\ \frac{1}{3} & \frac{-2}{3} & 4 \\ \frac{2}{3} & \frac{2}{3} & 13 \end{bmatrix}$$

$$A = \begin{bmatrix} 3 & 8 & 14 \\ \frac{1}{3} & \frac{-2}{3} & 4 \\ \frac{2}{3} & -1 & 13 \end{bmatrix}$$

$$A = \begin{bmatrix} 3 & 8 & 14 \\ \frac{1}{3} & \frac{-2}{3} & -\frac{2}{3} \\ \frac{2}{3} & -1 & 13 \end{bmatrix}$$

$$A = \begin{bmatrix} 3 & 8 & 14 \\ \frac{1}{3} & \frac{-2}{3} & -\frac{2}{3} \\ \frac{2}{3} & -1 & 3 \end{bmatrix}$$

Therefore,

$$\begin{bmatrix} 0 & 1 & 0 \\ 1 & 0 & 0 \\ 0 & 0 & 1 \end{bmatrix}\begin{bmatrix} 1 & 2 & 4 \\ 3 & 8 & 14 \\ 2 & 6 & 13 \end{bmatrix} = \begin{bmatrix} 1 & 0 & 0 \\ \frac{1}{3} & 1 & 0 \\ \frac{2}{3} & -1 & 1 \end{bmatrix}\begin{bmatrix} 3 & 8 & 14 \\ 0 & \frac{-2}{3} & -\frac{2}{3} \\ 0 & 0 & 3 \end{bmatrix}$$

2. Algorithm for computing $A = UL$ decompostition:
   a. Let $J = \begin{bmatrix} 0 & 0 & 1 \\ 0 & 1 & 0 \\ 1 & 0 & 0 \end{bmatrix}$, compute JAJ, denote the result by M.
   b. Using LU decomposition algorithm learnt in class to solve M=LU, which is:
      for $k = 1,2,\dots,n$
      $M(k,k:n) = M(k,k:n) - M(k,1:k-1)M(1:k-1,k:n)$
      $M(k+1:n,k) = (M(k+1:n,k) - M(k+1:n,1:k-1)M(1:k-1,k))/M(k,k)$
      end
   c. which L is the lower triangular matrix and U is the upper triangular matrix
   d. Then, Compute $JLJ = U'$ and $JUJ = L'$
   e. Finally, we obtain $A = (JLJ)(JUJ) = U'L'$

Explanation:
Since JLJ is upper triangular matrix and JUJ is lower triangular matrix and $J^2 = I$
$JAJ = JLUJ = JLJ^2UJ = (JLJ)(JUJ) = U'L'$
So, we can obtain the above algorithm.

3. Let $\boldsymbol{A} \in \mathbb{R}^{n\times n}$ be a tridiagonal matrix, i.e., $a_{ij} = 0$ if $|i-j| > 1$. We also assume that $\boldsymbol{A}$ is symmetric positive definite (SPD).
   (a) Prove that the Cholesky decompostion $\boldsymbol{A} = \boldsymbol{LL}^T$ satisfies $l_{ij} = 0$ for all $i - j > 1$. In other words, $\boldsymbol{L}$ is bi-diagonal.

3.(a) Using the Cholesky algorithm learnt in class to decompose A,

for $k = 1,2,\dots,n$

① $L(k,k) = \left(A(k,k) - L(k,1:k-1)(L(k,1:k-1))^T\right)^{\frac{1}{2}}$

② $L(k+1:n,k) = \left(A(k+1:n,k) - L(k+1:n,1:k-1)(L(k,1:k-1))^T\right)/L(k,k)$

end

Assume $A = \begin{bmatrix} d_1 & c_1 & & & \\ c_1 & d_2 & c_2 & & \\ & c_2 & d_3 & c_3 & \\ & & c_3 & \ddots & \ddots \\ & & & c_{n-1} & d_n \end{bmatrix}$ and $c_i, d_i \neq 0$

At step 1, $L(1,1) = \sqrt{A(1,1)} = \sqrt{d_1}$

$L(2:n,1) = \dfrac{A(2:n,1)}{L(1,1)}$

Since $A(3:n,1) = 0$, $A(2:n,1) = A(2,1)$.

$L(2,1) = \dfrac{A(2,1)}{L(1,1)} = \dfrac{c_1}{\sqrt{d_1}}$

So, $L(1:n,1)$, there is only 2 entries with value $\neq 0$.

The bidiagonal property holds at step

$\therefore$ The bi-diagonal property holds at step 1, i.e. $L(i,1) = 0$ for all $i - 1 > 1$.

Assume the bidiagonal property holds for step 1 to step k-1.

i.e. for $i = k-1$, $u_{ij} = 0$ for $j - i > 1$

for $j = k-1$. $l_{ij} = 0$ for $i - j > 1$

$\Rightarrow$ $i = k-1$, $l_{ji} = 0$ for $j - i > 1$.

For step k, (when $i = k$)

① $L(k,k) = \left(A(k,k) - L(k,1:k-1)(L(k,1:k-1))^T\right)^{\frac{1}{2}}$

② $L(k+1:n,k) = \left(A(k+1:n,k) - L(k+1:n,1:k-1)(L(k,1:k-1))^T\right)/L(k,k)$

which is same as ① $L(k,k) = A(k,k)^{\frac{1}{2}}$

② $L(k+1,k) = \dfrac{A(k+1,k)}{L(k,k)}$

$$L(k+2:n,k) = \frac{\left[A(k+2:n,k) - L(k+2:n,1:k-1)L(k+2:n,1:k-1)^T\right]}{L(k,k)}$$

$= \left(0 - 0(L(k,1:k-1)^T\right)/L(k,k)$

$= 0$

$\therefore$ The bidiagonal property holds for step k.

$\therefore$ Using mathematical induction,

The chloesky decomposition $A = LL^T$ satisfies bidiagonal property, $l_{ij} = 0$ for all $i - j > 1$.

3(b). Algorithm:
For $k = 1:n$
$L(k,k) = \sqrt{A(k,k) - L(k,k-1)^2}$
$L(k+1,k) = A(k+1,k)/L(k,k)$
End
After overwriting, algorithm becomes:
For $k = 1:n$
$A(k,k) = \sqrt{A(k,k) - A(k,k-1)^2}$
$A(k+1,k) = A(k+1,k)/A(k,k)$
End

Note that L matrix can be replaced by A.
Operation needed: if taking square root is counted as one operation, 4 operations are needed for first line, except when k=1, 2 operation is needed.
1 operation for second line, except when k=n, no operation is needed.
So it is $(4+1)n - 2 - 1 = 5n - 3$ cost.

3(c).
(c) Based on the Cholesky decomposition, construct an $O(n)$ algorithm to solve $\boldsymbol{Ax} = \boldsymbol{b}$. Express the number of operations needed in the form of $Cn + O(1)$ with explicit C.

First, decompose A in LL^T, where L and L^T are both bi-diagonal matrix using the algorith in (b).
For $k = 1:n$
$A(k,k) = \sqrt{A(k,k) - A(k,k-1)^2}$
$A(k+1,k) = A(k+1,k)/A(k,k)$
End

Then, we use forward substitution to compute $Ly = b$ where L is the lower − bidiaginal matrix.
Pseudo code for forward substitution:
$y_1 = \frac{b_1}{A(1,1)}$//for j=1
for $j = 2:n$
$y_j = \dfrac{b_j - A(j,j-1)y_{j-1}}{A(j,j)}$
end
After overwriting, algorithm becomes:
$b_1 = \dfrac{b_1}{A(1,1)}$
for $j = 2:n$
$b_j = \dfrac{b_j - A(j,j-1)b_{j-1}}{A(j,j)}$
end

Finally, after we obtain y, we use backward substitution to compute $L^Tx = y$
Pseudo Code for backward substitution:
$x_n = \frac{y_n}{A(n,n)}$//for i=n
for $i = n-1:1$
$x_i = \dfrac{y_i - A(i,i+1)y_{i+1}}{A(i,i)}$
end

After overwriting, algorithm becomes:
$b_n = \dfrac{b_n}{A(n,n)}$
for $i = n-1:1$
$b_i = \dfrac{b_i - A(i,i+1)b_{i+1}}{A(i,i)}$
end

Running Cost Analysis:
Using the result of (b), Cholesky decomposition costs $5n - 3$
Forward substitution costs $3(n-1) + 1 = 3n - 2$
Backward substition costs $3(n-1) + 1 = 3n - 2$
Total cost $= 11n - 7$

Q4(a).
4. We consider a discrete 1-D Laplacian equation $\boldsymbol{Ax} = \boldsymbol{b}$, where
$$\boldsymbol{A} = \begin{bmatrix} 2 & -1 & & & \\ -1 & 2 & -1 & & \\ & \ddots & \ddots & \ddots & \\ & & \ddots & \ddots & -1 \\ & & & -1 & 2 \end{bmatrix} \in \mathbb{R}^{n\times n} \qquad \boldsymbol{b} = \begin{bmatrix} 1 \\ 1 \\ \vdots \\ 1 \end{bmatrix} \in \mathbb{R}^n$$
   (a) Prove that $\boldsymbol{A}$ is SPD.
   (b) Since $\boldsymbol{A}$ is also tridiagonal, the algorithms in Question 3 can be applied. Write a Matlab code to implement your algorithm in Question 3(b)(c) for solving $\boldsymbol{Ax} = \boldsymbol{b}$ where A template file `spdtridiagsolver.m` is provided. Plot the solution you obtained with $n = 500$.

4(a). Prove $\forall x, x^TAx > 0$

Let $x = \begin{pmatrix} x_1 \\ x_2 \\ \dots \\ x_n \end{pmatrix} \in \mathbb{R}^n$

$$x^TAx = (x_1 \quad x_2 \quad \dots \quad x_n)\begin{pmatrix} 2 & -1 & & & \\ -1 & 2 & -1 & & \\ & \dots & \dots & \dots & \\ & & \dots & \dots & -1 \\ & & & -1 & 2 \end{pmatrix}\begin{pmatrix} x_1 \\ x_2 \\ \dots \\ x_n \end{pmatrix}$$

$x^TAx = (2x_1 - x_2, -x_1 + 2x_2 - x_3, -x_2 + 2x_3 - x_4, \dots, -x_{n-1} + 2x_n)\begin{pmatrix} x_1 \\ x_2 \\ \dots \\ x_n \end{pmatrix}$

$x^TAx = 2x_1^2 - x_1x_2 - x_1x_2 + 2x_2^2 - x_3x_2 - x_2x_3 + 2x_3^2 - x_4x_3 + \cdots - x_{n-1}x_n + 2x_n^2$

$x^TAx = x_1^2 + (x_1 - x_2)^2 + (x_2 - x_3)^2 + \cdots + (x_{n-1} - x_n)^2 + x_n^2 \geq 0$

And $x^TAx = 0$ only if $x_1 = x_2 = \cdots = x_n = 0$

$\because x \neq 0$

$\therefore x^TAx > 0$

And also we can observe that $A = A^T$

$\therefore$ A is a SPD.

4(b).