Scientific Computing - Homework 2

蔡子若 018071910011

1 Page 94, 2.45(Scientific Computing: An Introductory Survey)

Since A is nonsingular, computing the product $A^{-1}Bc$ is equivalent to solve the equations $Ax = Bc$.

First compute Bc. Then compute the LU factorization, $A = LU$.

Solve the equations $Ly = Bc$ and $Ux = y$ respectively.

Hence $x = U^{-1}y = U^{-1}L^{-1}Bc = A^{-1}Bc$.

2 Page 95, 2.48

$A^T = (LU)^T = U^T L^T$, where U^T is a lower triangular matrix and L^T is an upper triangular matrix.

Solving the equations $A^T x = b$ is equivalent to solving $U^T L^T x = b$.

Hence we first solve $U^T y = b$ then solve $L^T x = y$.

3 Page 95, 2.49

(a) Solve $Ly = b$, then solve $Px = y$.

(b) Solve $Py = b$, then solve $Lx = y$.

4 Page 95, 2.51

It is possible.

Example: $x = (3,\ 3)$, $y = (4,\ 1)$.

$\|x\|_1 = 6, \|y\|_1 = 5,\ \|x\|_1 > \|y\|_1$.

But $\|x\|_\infty = 3, \|y\|_\infty = 4,\ \|x\|_\infty < \|y\|_\infty$.

5 Page 95, 2.52

$\|A\|_1$ is easier to compute.

To compute $\|A\|_1$ we only need to get the absolute sum of each column and choose the biggest sum.

To compute $\|A\|_2$ we need to compute the singular-value decomposition or get the biggest eigenvalue of $A^T A$, which is absolutely harder.

6 Page 95, 2.57

(a) $\text{cond}(A) = \|A\|_1 \cdot \|A^{-1}\|_1 = 6 \times \dfrac{1}{2} = 3$

(b) $\text{cond}(A) = \|A\|_\infty \cdot \|A^{-1}\|_\infty = 6 \times \dfrac{1}{2} = 3$. The answer does not differ.

7 Page 95, 2.61

(a) $\text{cond}(A) = \|A\|_2 \cdot \|A^{-1}\|_2 = 10^{10} \times 10^{10} = 10^{20} \gg 1$, the matrix is ill-conditioned.

(b) $\text{cond}(A) = \|A\|_2 \cdot \|A^{-1}\|_2 = 10^{10} \times 10^{-10} = 1$, the matrix is well-conditioned.

(c) $\text{cond}(A) = \|A\|_2 \cdot \|A^{-1}\|_2 = 10^{-10} \times 10^{10} = 1$, the matrix is well-conditioned.

(d) A is singular, hence $\text{cond}(A) = \infty$. The matrix is ill-conditioned.

8 Page 96, 2.77

$$\begin{bmatrix} 4 & 2 \\ 2 & 2 \end{bmatrix} = \begin{bmatrix} 2 & 0 \\ 1 & 1 \end{bmatrix} \begin{bmatrix} 2 & 1 \\ 0 & 1 \end{bmatrix}$$

9 Page 97, 2.10

(a) Assume $P \in \mathbb{R}^{n\times n}$, $P^T = (a_{ij})$, $P^T P = (b_{ij})$.

Then $b_{ii} = \sum\limits_{j=1}^{n} a_{ij}^2$, $b_{ij} = \sum\limits_{k=1}^{n} a_{ik}a_{jk}, i \neq j$.

Since P is a permutation matrix, then

$\forall i, 1 \leqslant i \leqslant n$, $\exists j, 1 \leqslant j \leqslant n$, s.t. $a_{ij} = 1, a_{ik} = 0, k = 1, 2, \cdots j-1, j+1, \cdots n$, $a_{lj} = 0, l = 1, 2 \cdots i-1, i+1, \cdots n$.

Hence $b_{ii} = \sum\limits_{j=1}^{n} a_{ij}^2 = 1 + 0 \times (n-1) = 1$,

For $i \neq j$, $b_{ij} = \sum\limits_{k=1}^{n} a_{ik}a_{jk} = 0 \times n = 0$.

Then $P^T P = I$, $P^{-1} = P^T$.

(b) Assume $P = (a_{ij})$, follow the algorithm:

(i) Set $k = 1$.

(ii) If $k = n$, stop.

For $k < n$:

If $a_{kk} = 1$, then set $k \leftarrow k+1$ and return to (ii).

If $a_{kk} = 0$, then find $a_{kl}, k < l \leqslant n$ s.t. $a_{kl} = 1$. Multiply the row-interchanging elementary matrix M_{kl} to P from the left, which interchange k-th and l-th row of matrix P. Set $k \leftarrow k+1$ and return to (ii).

Now prove: This algorithm will finally interchange the rows of P such that it becomes the identity matrix I.

First, everytime P multiply a row-interchanging elementary matrix to the left, it is still a permutation matrix.

Second, when the algorithm (ii) is on the k-th step, $\forall i, 1 \leqslant i < k$, $a_{ii} = 1$. This can be proved by mathematical induction.

Finally, when $k = n$, from above we get $a_{ii} = 1, i = 1, 2, \cdots n-1$. Then a_{nn} must be 1 since the matrix is a permutation matrix. Proof finished.

Since the inverse matrix of a row-interchanging elementary matrix is itself, P can be expressed as a product of pairwise interchanges.

10 Page 98, 2.31

Since A is symmetric definite matrix, $\exists B$ s.t. $A = B^T B$, B is nonsingular.

(1) $\|x\|_A = 0 \Leftrightarrow (x^T Ax)^{1/2} = 0 \Leftrightarrow x^T B^T Bx = 0 \Leftrightarrow (Bx)^T Bx = 0 \Leftrightarrow Bx = 0 \Leftrightarrow x = 0$.

(2) For any scalar γ, $\|\gamma x\|_A = (\gamma^2 x^T Ax)^{1/2} = |\gamma|(x^T Ax)^{1/2} = |\gamma| \cdot \|x\|_A$.

(3) $\|x+y\|_A \leqslant \|x\|_A + \|y\|_A \Leftrightarrow \sqrt{(x+y)^T A(x+y)} \leqslant \sqrt{x^T Ax} + \sqrt{y^T Ay} \Leftrightarrow x^T Ay \leqslant \sqrt{x^T Axy^T Ay}$
$\Leftrightarrow (Bx)^T By \leqslant \sqrt{(Bx)^T Bx(By)^T By}$.

For each $a, b \in \mathbb{R}^{n\times 1}$, $a^T b = \sum_{i=1}^{n} a_i b_i \leqslant \sqrt{\left(\sum_{i=1}^{n} a_i^2\right)\left(\sum_{i=1}^{n} b_i^2\right)} = \sqrt{a^T ab^T b}$, which is exactly the Cauchy–Schwarz inequality.

Hence $(Bx)^T By \leqslant \sqrt{(Bx)^T Bx(By)^T By}$, which leads to $\|x+y\|_A \leqslant \|x\|_A + \|y\|_A$.

11 Computer problem (in C or C++)

Using Gaussian elimination to achieve the LU decomposition with and without a column pivoting; Using the two LU decomposition algorithm to solve linear systems in which the coefficient matrix is (1) general nonsingular matrix; (2) positive definite matrix; (3) diagonally dominant matrix. Compare the numerical accuracy for the two algorithms. The size of the matrices should be greater than 1000.

For general nonsingular matrix, the numerical accuracy with pivoting is quite better than the method without pivoting. The Frobenius norm of the residual $(A \cdot x - b)$ without pivoting is 1.46536×10^{-9}. With pivoting the Frobenius norm of the residual is 3.14365×10^{-13}.

For positive definite matrix, the numerical accuray with pivoting is almost the same as that without pivoting. The Frobenius norm of the residual without pivoting is 4.64723×10^{-9}, and the norm of the residual with pivoting is 5.26639×10^{-9}, which is even a little bigger.

For diagonally dominant matrix, there is no chance to make a pivoting.