作业反馈3-15

TC第28.1节练习2、3、6、7

TC第28.2节练习1、2、3

TC第28.3节练习1、3

TC第28章问题1

28.1-6

Show that for all $n \geq 1$, there exists a singular $n \times n$ matrix that has an LU decomposition.

- n>1时：

$$\begin{pmatrix} 1 & & & \\ & 1 & & \\ & & 1 & \\ & & & 0 \end{pmatrix} = \begin{pmatrix} 1 & & & \\ & 1 & & \\ & & 1 & \\ & & & 1 \end{pmatrix} \cdot \begin{pmatrix} 1 & & & \\ & 1 & & \\ & & 1 & \\ & & & 0 \end{pmatrix}$$

- n=1时：

$$(0) = (1) \cdot (0)$$

28.1-7

In LU-DECOMPOSITION, is it necessary to perform the outermost **for** loop iteration when $k = n$? How about in LUP-DECOMPOSITION?

- k=n时到底需不需要执行最外层循环？

```
LU-DECOMPOSITION(A)
n = A.rows
let L and U be new n × n matrices
initialize U with 0s below the diagonal
initialize L with 1s on the diagonal and 0s above the diagonal
for k = 1 to n
    u_kk = a_kk
    for i = k + 1 to n
        l_ik = a_ik / u_kk          // l_ik holds v_i
        u_ki = a_ki                 // u_ki holds w_i^T
    for i = k + 1 to n
        for j = k + 1 to n
            a_ij = a_ij − l_ik u_kj
return L and U
```

```
LUP-DECOMPOSITION(A)
n = A.rows
let π[1..n] be a new array
for i = 1 to n
    π[i] = i
for k = 1 to n
    p = 0
    for i = k to n
        if |a_ik| > p
            p = |a_ik|
            k' = i
    if p == 0
        error "singular matrix"
    exchange π[k] with π[k']
    for i = 1 to n
        exchange a_ki with a_k'i
    for i = k + 1 to n
        a_ik = a_ik / a_kk
        for j = k + 1 to n
            a_ij = a_ij − a_ik a_kj
```

28.2-2

Let $M(n)$ be the time to multiply two $n \times n$ matrices, and let $L(n)$ be the time to compute the LUP decomposition of an $n \times n$ matrix. Show that multiplying matrices and computing LUP decompositions of matrices have essentially the same difficulty: an $M(n)$-time matrix-multiplication algorithm implies an $O(M(n))$-time LUP-decomposition algorithm, and an $L(n)$-time LUP-decomposition algorithm implies an $O(L(n))$-time matrix-multiplication algorithm.

已知：

- ***Theorem 28.1 (Multiplication is no harder than inversion)***
- ***Theorem 28.2 (Inversion is no harder than multiplication)***
- **Computing a matrix inverse from an LUP decomposition**

$MM \Leftrightarrow Inv$

$LUP \Rightarrow Inv$

$MM, Inv \Rightarrow LUP?$

[redacted]

对U_1和$A_2P_1^{-1}$分块,使得$U_1=(\overline{U_1}\ \ B)$,$A_2P_1^{-1}=(C\ \ D)$,同时,令$F=D-C\overline{U_1}^{-1}B$,此时

$$A=\begin{pmatrix}L_1 & 0\\ C\overline{U_1}^{-1} & I_{n/2}\end{pmatrix}\begin{pmatrix}\overline{U_1} & B\\ 0 & F\end{pmatrix}P_1$$

再将FLUP分解为$L_2U_2P_2$,此时

$$A=\begin{pmatrix}L_1 & 0\\ C\overline{U_1}^{-1} & L_2\end{pmatrix}\begin{pmatrix}\overline{U_1} & BP_2^{-1}\\ 0 & U_2\end{pmatrix}\begin{pmatrix}I_{n/2} & 0\\ 0 & P_2\end{pmatrix}P_1$$

显然

$$\begin{pmatrix}I_{n/2} & 0\\ 0 & P_2\end{pmatrix}P_1$$

还是一个置换矩阵,因此,这就是新的P,上面的式子就是A的LUP分解,在整个过程中,我们需要计算矩阵求逆和矩阵乘法,并且他们的复杂度是一样的,继而有矩阵求逆⇒LUP分解

The Schur complement arises as the result of performing a block Gaussian elimination by multiplying the matrix M from the right with the "block lower triangular" matrix

$$L = \begin{bmatrix} I_p & 0 \\ -D^{-1}C & I_q \end{bmatrix}.$$

Here I_p denotes a $p \times p$ identity matrix. After multiplication with the matrix L the Schur complement appears in the upper $p \times p$ block. The product matrix is

$$ML = \begin{bmatrix} A & B \\ C & D \end{bmatrix} \begin{bmatrix} I_p & 0 \\ -D^{-1}C & I_q \end{bmatrix} = \begin{bmatrix} A - BD^{-1}C & B \\ 0 & D \end{bmatrix}$$
$$= \begin{bmatrix} I_p & BD^{-1} \\ 0 & I_q \end{bmatrix} \begin{bmatrix} A - BD^{-1}C & 0 \\ 0 & D \end{bmatrix}.$$

This is analogous to an LDU decomposition. That is, we have shown that

[redacted]

and inverse of M thus may be expressed involving D^{-1} and the inverse of Schur's complement (if it exists) only as

[redacted]

$$= \begin{bmatrix} (A - BD^{-1}C)^{-1} & -(A - BD^{-1}C)^{-1}BD^{-1} \\ -D^{-1}C(A - BD^{-1}C)^{-1} & D^{-1} + D^{-1}C(A - BD^{-1}C)^{-1}BD^{-1} \end{bmatrix}.$$

是否有灵感?

遵循TC上的LU方法，可以更加直接：

$$M = \begin{bmatrix} A & B \\ C & D \end{bmatrix}, U' = \begin{bmatrix} I_p & -A^{-1}B \\ 0 & I_q \end{bmatrix}$$

则：

$$MU' = \begin{bmatrix} A & 0 \\ C & D - CA^{-1}B \end{bmatrix}$$

所以：

$$M = MU'U'^{-1} = \begin{bmatrix} A & 0 \\ C & D - CA^{-1}B \end{bmatrix} * \begin{bmatrix} I_p & -A^{-1}B \\ 0 & I_q \end{bmatrix}^{-1}$$

$$= \begin{bmatrix} A & 0 \\ C & D - CA^{-1}B \end{bmatrix} * \begin{bmatrix} I_p & A^{-1}B \\ 0 & I_q \end{bmatrix}$$

$$= \begin{bmatrix} I_p & 0 \\ CA^{-1} & I_q \end{bmatrix} * \begin{bmatrix} A & 0 \\ 0 & D - CA^{-1}B \end{bmatrix} * \begin{bmatrix} I_p & A^{-1}B \\ 0 & I_q \end{bmatrix}$$

复杂度待进一步确认

28.2-3

Let $M(n)$ be the time to multiply two $n \times n$ matrices, and let $D(n)$ denote the time required to find the determinant of an $n \times n$ matrix. Show that multiplying matrices and computing the determinant have essentially the same difficulty: an $M(n)$-time matrix-multiplication algorithm implies an $O(M(n))$-time determinant algorithm, and a $D(n)$-time determinant algorithm implies an $O(D(n))$-time matrix-multiplication algorithm.

Given the LUP decomposition $A = P^{-1}LU$ of a square matrix A, the determinant of A can be computed straightforwardly as

$$\det(A) = \det(P^{-1})\det(L)\det(U) = (-1)^S \left(\prod_{i=1}^{n} l_{ii}\right)\left(\prod_{i=1}^{n} u_{ii}\right).$$

- **矩阵乘法➔求行列式**
  - 求行列式 ≤ LUP分解
  - LUP分解 ≤ 矩阵乘法　练习28.2-2
- **求行列式➔矩阵乘法**
  - 矩阵乘法 ≤ 求逆矩阵　定理28.1
  - 求逆矩阵 ≤ 求行列式+求伴随矩阵
  - 求伴随矩阵 ≤ 求行列式

$$\mathbf{A}^{-1} = \frac{1}{\det(\mathbf{A})}\operatorname{adj}(\mathbf{A})$$

?