MATH 270A – NUMERICAL LINEAR ALGEBRA – HOMEWORK 1

Due Friday, October 12th. Handwritten submissions only.

Exercise 1
Compute the inverse the unit lower triangular matrix

$$L = \begin{pmatrix} 1 & 0 & 0 & 0 \\ -4 & 1 & 0 & 0 \\ -2 & 4 & 1 & 0 \\ 0 & 3 & -1 & 1 \end{pmatrix}.$$

Solution 1
The inverse is

$$L = \begin{pmatrix} 1 & 0 & 0 & 0 \\ 4 & 1 & 0 & 0 \\ -14 & -4 & 1 & 0 \\ -26 & -7 & 1 & 1 \end{pmatrix}.$$

Exercise 2
Describe an algorithm (in pseudocode) that computes the matrix-vector product $y = Ax$ of an $n \times n$ matrix A and an n-dimensional vector x.

Solution 2
- FOR $i = 1, \dots, n$ DO
- $y_i = 0$
- FOR $j = 1, \dots, n$ DO $y_i = y_i + a_{ij}x_j$ END FOR
- END FOR

Exercise 3
Describe an *in place* algorithm (in pseudocode) that computes the matrix-vector product Lx of a lower triangular $n \times n$ matrix L and an n-dimensional vector x and writes the result into x. Here, *in place* means the algorithm uses an amount of auxiliary memory that does not depend on the matrix dimension.

Solution 3
- FOR $j = n, \dots, 1$ DO
- $x_j = a_{jj}x_j$
- FOR $k = 1, \dots, k-1$ DO $x_j = x_j - a_{jk}x_k$ END FOR
- END FOR

Exercise 4
Suppose that A is an invertible $n \times n$ matrix and let $b^{(1)}, b^{(2)}, \dots, b^{(M)}$ be M vectors of dimension n. We want to solve the linear systems of equations

$$Ax^{(1)} = b^{(1)}, \quad Ax^{(2)} = b^{(2)}, \quad \dots, \quad Ax^{(M)} = b^{(M)}.$$

(1) How many divisions and multiplications are performed if Gaussian elimination is used for all M systems?

(2) How many divisions and multiplications are performed if first the LU decomposition of A is calculated and then the systems are solved with successive triangular substitutions?
(3) For which M and n will which approach use less divisions and multiplications?

Solution 4
For the sake of brevity, we introduce the notation

$$A_n = \sum_{k=1}^{n-1} k, \quad S_n = \sum_{k=1}^{n-1} k^2.$$

We count the operations as follows.

(1) Using Gaussian elimination uses in its first phase $M \cdot A_n$ divisions and $M \cdot S_n$ multiplications on the matrix entries, and further $M \cdot A_n$ multiplications on the right-hand side. In the backsubstitution phase, it uses further $n \cdot M$ divisions and $M \cdot A_n$ multiplications. Lumping division and multiplication together, we get a total count of

$$X_n = M \cdot n + 3M \cdot A_n + M \cdot S_n.$$

(2) The decomposition phase of the LU decomposition uses A_n divisions and S_n multiplications. The substitution phase of the LU decomposition uses Mn divisions (for the backward substitution) and $2M \cdot A_n$ multiplications. Together we get a count of

$$Y_n = Mn + (2M+1)A_n + S_n.$$

(3) The question is for which $n \geq 1$ the difference

$$\begin{aligned} &\Delta_n \\ &\quad = X_n - Y_n \\ &\quad = Mn + 3MA_n + MS_n - Mn - (2M+1)A_n - S_n \\ &\quad = (M-1)S_n + (M-1)A_n \end{aligned}$$

is positive. This is the case for $M \geq 2$ and $n \geq 2$.

Exercise 5
Consider a unit lower triangular matrix L a upper triangular matrix U. Suppose that L and U are saved in the memory of a matrix A.

$$L = \begin{pmatrix} 1 & 0 & \dots & 0 \\ l_{21} & 1 & \dots & 0 \\ \vdots & & \ddots & \vdots \\ l_{n1} & l_{n2} & \dots & 1 \end{pmatrix}, \quad U = \begin{pmatrix} u_{11} & u_{12} & \dots & u_{1n} \\ 0 & u_{22} & \dots & u_{2n} \\ \vdots & & \ddots & \vdots \\ 0 & \dots & 0 & u_{nn} \end{pmatrix}, \quad A = \begin{pmatrix} u_{11} & u_{12} & \dots & u_{1n} \\ l_{21} & u_{22} & \dots & u_{2n} \\ \vdots & & \ddots & \vdots \\ l_{n1} & l_{n2} & \dots & u_{nn} \end{pmatrix}.$$

Describe an *in place* algorithm (in pseudocode) that computes the matrix-matrix product LU and writes the result into the memory of A. Here, *in place* means the algorithm uses an amount of auxiliary memory that does not depend on the matrix dimension.

Solution 5
- FOR $i = n, \dots, 1$ DO
- FOR $j = n, \dots, 1$ DO
- $F = 0$
- FOR $k = 1, \dots, \min(i, j)$ DO $F = F + a_{ik}a_{kj}$ END FOR
- $a_{ij} = F$
- END FOR
- END FOR

Exercise 6
Consider the matrix A and the vector b given by

$$A = \begin{pmatrix} 2 & 8 & 1 \\ 4 & 4 & -1 \\ -1 & 2 & 12 \end{pmatrix}, \quad b = \begin{pmatrix} 32 \\ 16 \\ 52 \end{pmatrix}.$$

(1) Compute the LU decomposition of A.
(2) Solve the linear system $Ax = b$ by successive substitution. Double check your result.
(3) Compute the inverse of A^{-1}. Double check your result.

Solution 6
The LU decomposition of A is given by the matrices

$$L = \begin{pmatrix} 1 & 0 & 0 \\ 2 & 1 & 0 \\ -0.5 & -0.5 & 1 \end{pmatrix}, \quad U = \begin{pmatrix} 2 & 8 & 1 \\ 0 & -12 & -3 \\ 0 & 0 & 11 \end{pmatrix}$$

Successive substitution gives the following values:

$$y = \begin{pmatrix} 32 \\ -48 \\ 44 \end{pmatrix}, \quad x = \begin{pmatrix} 2 \\ 3 \\ 4 \end{pmatrix}$$

The inverse matrix is given by

$$A^{-1} = \begin{pmatrix} -25/132 & 47/132 & 1/22 \\ 47/264 & -25/264 & -1/44 \\ -1/22 & 1/22 & 1/11 \end{pmatrix}.$$

Exercise 7
Gaussian elimination (or LU decomposition) is a possible method to compute the determinat of a matrix.

(1) Suppose that $LU = A$ is the LU decomposition of an $n \times n$ matrix A. Prove that $\det(A) = \det(U)$.
(2) How many multiplications and divisions are needed to compute $\det(A)$ using the Laplace expansion?
(3) For which value of n does the Laplace expansion use more multiplications and divisions than the method using Gaussian elimination?

Solution 7
First, the identity $\det(A) = \det(U)$ follows from the product formula for the determinant and the fact that $\det(L) = 1$.

The Laplace expansion ranges over $n!$ different permutations, and for each permutation, we compute $n-1$ different products. There are no divisions, hence the operation count is $(n-1)(n!)$.

Computing the U component in the LU decomposition of the matrix A requires $n(n-1)/2$ divisions and $\sum_{k=1}^{n-1} k^2 = n(n-1)(2n-1)/6$ multiplications. Furthermore, computing the determinant eventually needs $n-1$ additional multiplications. Using equivalence transformations, we get

$$\begin{aligned} & \frac{1}{6}n(n-1)(2n-1) + \frac{1}{2}n(n-1) + (n-1) < (n!)(n-1) \\ \equiv\ & \frac{n}{6}(2n-1) + \frac{n}{2} + 1 < (n!) \\ \equiv\ & \frac{1}{6}(2n^2 - n + 3n + 6) < (n!) \\ \equiv\ & \frac{1}{6}(2n^2 + 2n + 6) < (n!). \end{aligned}$$

This inequality holds certainly for, say, $n \geq 5$. Manually checking shows that it holds for $n \geq 3$.

Exercise 8

Solve the following two problems.

(a) Let A be a matrix for which the LU decomposition exists without pivoting. Show that there exists a unique lower triangular matrix L and a unique unit upper triangular matrix U such that $A = LU$.

(b) Suppose that L, L' are invertible lower triangular matrices and U, U' are invertible upper triangular matrices such that $LU = L'U'$. What is the relation between L and L' and between U and U', respectively?

Solution 8(a) For the first part of the exercise, we recall that the LU decomposition $A = LU$ with L having unit diagonal entries is unique. There exists a unique diagonal matrix D and a unique unit upper triangular matrix R such that $U = DR$. We define $L' = LD$, which shows the existence of the decomposition $A = L'R$ as in the statement of the exercise. The uniqueness of the decomposition can be shown as in the lecture.

(b) For the second part of the exercise, let us define diagonal matrices D_R, D'_R and unit upper triangular matrices R, R' such that $U = DR$ and $U' = D'R'$. Furthermore, we define diagonal matrices D_L, D'_L and unit upper triangular matrices B, B' such that $L = BD_L$ and $L' = B'D'_L$.

We then have $A = BD_LD_RR = B'D'_LD'_RR'$. The uniqueness of the LU decomposition with unit lower triangular matrix implies that $B = B'$ and $D_LD_RR = D'_LD'_RR'$. From the latter we get $D_LD_R = D'_LD'_R$ and $R = R'$.

Consequently, $L = BD_L = B'D_L = L'(D')_L^{-1}D_L$ and $U = U_RR = D_RR' = D_R(D')_R^{-1}U'$. Finally, $(D')_L^{-1}D_LD_R(D')_R^{-1} = (D')_L^{-1}D'_LD'_R(D')_R^{-1} = I$.

Our overall conclusion is that there exists a diagonal matrix T such that $L = L'T$ and $U = T^{-1}U'$.

[(a)]

Remark 1

Counting the number of floating-point operations gives a rough idea how much run-time an algorithm will need. However, observed run-times are influenced by a multitude of factors in the software and the hardware.