MATH 270A – NUMERICAL LINEAR ALGEBRA – HOMEWORK 1

Due Friday 12th. Handwritten submissions only.

Exercise 1
Compute the inverse the unit lower triangular matrix

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

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.

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.

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?

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.

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.

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 Gaussian elimination?

Exercise 8
Solve the following two problems.

(a) Show that for every matrix A 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 lower triangular matrices and U, U' are upper triangular matrices such that $LU = L'U'$. What is the relation between L and L' and between U and U', respectively?

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.