Problem 1 **4 points**

Compute the LU-decomposition (without pivoting) of A and use it to solve the system of linear equations $Ax = b$.

E)

$$A = \begin{bmatrix} 3 & 5 & 1 \\ 6 & 8 & 6 \\ 12 & 26 & -7 \end{bmatrix}, \qquad b = \begin{bmatrix} 27 \\ 68 \\ 70 \end{bmatrix}.$$

H)

$$A = \begin{bmatrix} 1 & 2 & 3 & 4 \\ 4 & 13 & 10 & 17 \\ 3 & -4 & 16 & 6 \\ 1 & 27 & -16 & 15 \end{bmatrix}, \qquad b = \begin{bmatrix} 7 \\ 60 \\ -56 \\ 200 \end{bmatrix}.$$

Problem 2 **4 Points**

E a) How many algebraic operations $(+, -, *, /)$ are necessary to solve the system of linear equations $Lx = b$, where $b \in \mathbb{R}^n$ and $L \in \mathbb{R}^{n\times n}$ is a lower triangular matrix with 1's on the diagonal?

E b) How many algebraic operations $(+, -, *, /)$ are necessary to multiply two $n \times n$-matrices?

Note: If the algebraic operations $+, -, *, /$ are performed on the computer, using floating point arithmetic, they are called flops (floating point operations).

H) How many algebraic operations $(+, -, *, /)$ are necessary

- to solve the system of linear equations $Ux = b$, where $U \in \mathbb{R}^{n\times n}$ is an upper triangular matrix whose diagonal elements are not 0?
- to multiply a matrix $A \in \mathbb{R}^{n\times n}$ by a vector $x \in \mathbb{R}^n$?

Hint: $\sum_{k=1}^{n-1} k = 1 + 2 + \ldots + n - 1 = \frac{n(n-1)}{2}$.

Problem 3 **4 Points**

Row operations as matrix multiplication.

H) Let $B = LA$ where $A \in \mathbb{R}^{4\times 4}$ and

$$L = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & -\ell_1 & 1 & 0 \\ 0 & -\ell_2 & 0 & 1 \end{bmatrix}, \qquad \ell_1, \ell_2 \in \mathbb{R}.$$

Show: B is the matrix which we get if we subtract, in the matrix A, from the third row the ℓ_1-multiple of the second row and from the fourth row the ℓ_2-multiple of the second row.

What happens if we multiply L and A in reverse order? Hint: Compute $C = AL$ and interpret the result as column operations.

Problem 4 **4 Points**

Compute the Cholesky-factorization for the following matrices (by hand).

E)

$$A = \begin{bmatrix} 4 & 10 & -4 \\ 10 & 34 & -7 \\ -4 & -7 & 21 \end{bmatrix},$$

H)

$$A = \begin{bmatrix} 16 & -4 & -8 & -4 \\ -4 & 10 & -13 & 10 \\ -8 & -13 & 33 & -19 \\ -4 & 10 & -19 & 20 \end{bmatrix}.$$

Problem 5 **3 Points**

Use the method of completing the square to show which of the following matrices are positive definite.

E)

$$\begin{bmatrix} 25 & 15 \\ 15 & 7 \end{bmatrix}, \qquad \begin{bmatrix} 25 & 15 \\ 15 & 12 \end{bmatrix}, \qquad \begin{bmatrix} 25 & 15 \\ 15 & 9 \end{bmatrix}.$$

H)

$$\begin{bmatrix} 16 & 12 \\ 12 & 9 \end{bmatrix}, \qquad \begin{bmatrix} 16 & -12 \\ -12 & 10 \end{bmatrix}, \qquad \begin{bmatrix} 16 & 20 \\ 20 & 0 \end{bmatrix}.$$

Problem 6 **5 Points**

E) We saw in the lecture that the diagonal elements of a positive definite matrix are positive. We can generalize this further: Let $A \in \mathbb{R}^{n\times n}$ be positive definite, and let $i = [i_1, i_2, \ldots, i_p]$ be a vector of pairwise different indices ($i_k \in \mathbb{N}, i_k \leq n$). Then also the corresponding sub-matrix of A,

$$\widetilde{A} = \begin{bmatrix} a_{i_1 i_1} & a_{i_1 i_2} & \ldots & a_{i_1 i_p} \\ a_{i_2 i_1} & a_{i_2 i_2} & \ldots & a_{i_2 i_p} \\ \vdots & & & \vdots \\ a_{i_p i_1} & \ldots & \ldots & a_{i_p i_p} \end{bmatrix} \in \mathbb{R}^{p\times p},$$

is positive definite. Explain this statement and illustrate it with examples. We can use this fact to show that a matrix is *not* positive definite.

Notes

Python: create the matrix $\widetilde{A}$ with the command `A[numpy.ix_(i,i)]`, where `i` is the index vector.
Matlab: create the matrix $\widetilde{A}$ with the command `A(i,i)`, where `i` is the index vector.

H) Show, using a method of your choice (e.g. completing the square, determinants, eigenvalues, etc.), which of the following matrices are positive definite and which are not.

$$A_1 = \begin{bmatrix} 1 & 1 \\ 1 & 1 \end{bmatrix}, \quad A_2 = \begin{bmatrix} 1 & -3 & 1 \\ -3 & 5 & -2 \\ 1 & -2 & 1 \end{bmatrix}, \quad A_3 = \begin{bmatrix} 7 & 1 \\ 1 & 7 \end{bmatrix}, \quad A_4 = \begin{bmatrix} \mathbf{0} & A_3 \\ A_3 & \mathbf{0} \end{bmatrix}, \quad A_5 = \begin{bmatrix} 1 & 4 & -5 & 6 \\ 4 & -2 & -6 & 2 \\ -5 & -6 & 1 & 3 \\ 6 & 2 & 3 & 100 \end{bmatrix}.$$

Programming Assignment

(a) Write a Python or Matlab function `x=forback(L,U,b)`, which solves, for given lower triangular matrix $L \in \mathbb{R}^{n\times n}$ and upper triangular matrix $R \in \mathbb{R}^{n\times n}$, the system of linear equations $LU\,x = b$ using forward substitution and backward substitution. Test the program with

$$L = \begin{bmatrix} 1 & 0 & 0 \\ -2 & 1 & 0 \\ 4 & 5 & 1 \end{bmatrix}, \qquad U = \begin{bmatrix} 2 & -1 & 6 \\ 0 & 3 & 9 \\ 0 & 0 & -2 \end{bmatrix}, \qquad b = \begin{bmatrix} 18 \\ -3 \\ 231 \end{bmatrix}.$$

The solution of $LU\,x = b$ is then $x = [1\ 2\ 3]^\top$.

(b) Write your own function `[L,U]=mylu(A)` in Python or Matlab which computes the LU-decomposition of $A \in \mathbb{R}^{n\times n}$ (without pivoting). Do not use Python commands such as `numpy.linalg.solve(A,b)` or Matlab comands such as `A\b` to solve the system of linear equations.
You can test your program with the matrices from question (a), i.e. set $A = LU$. The function should take the matrix A as input argument and give back the matrices L, U as output arguments.

(c) Create a matrix $A = [a_{ij}]$ and a block matrix $B = [b_{ij}]$, s.t.

$$A = x * y^T \quad \text{und} \quad B = \left(\begin{array}{c|c} \begin{pmatrix} a_{11} & a_{13} & a_{15} & a_{17} \\ a_{41} & a_{43} & a_{45} & a_{47} \\ a_{71} & a_{73} & a_{75} & a_{77} \end{pmatrix} & c * c^T \\ \hline 1.9 * d^T & 1\ 2\ 3 \end{array} \right)$$

with $x = \begin{pmatrix} 0.2 & 0.4 & 0.5 & 0.6 & 0.7 & 0.8 & 0.9 \end{pmatrix}^T$, $y = \begin{pmatrix} 3 & 4 & 6 & 8 & 10 & 12 & 14 & 16 \end{pmatrix}^T$, $c = \begin{pmatrix} 3 & 4.3 & 7 \end{pmatrix}^T$, and $d = \begin{pmatrix} 1 & \pi & 2 & 3 \end{pmatrix}^T$.

(d) Write a program, in Python or Matlab, to create the following plot including axis, labels, and title.
Python: Check out the *Subplots, axes, and figures* example at matplotlib.org/gallery.html.
Matlab: Use the commands `subplot` and `hold on`.