



## DIRECT METHODS FOR SOLVING LINEAR SYSTEMS

### 1. Objectives

The goals of the laboratory workshop are as follows:

- to learn fundamental direct methods for solving systems of linear equations, especially sparse linear systems,
- to study the selected algorithms for matrix decompositions,
- to train the skills in coding the selected direct algorithms for solving linear systems in Matlab.

The workshop is scheduled for **3** academic hours.

### 2. Introduction

There are two basic classes of methods for solving a system of linear equations: *direct* and *iterative* methods. Direct methods theoretically give an exact solution in a (predictable) finite number of steps. Unfortunately, this does not have to be true in computational approach due to rounding errors: an error made in one step spreads in all following steps. Classical direct methods usually involve a variety of matrix factorization techniques such as the LU, LDU, LUP, Cholesky, QR, EVD, SVD, and GSVD.

A matrix factorization (or decomposition) decomposes a matrix  $\mathbf{A}$  into a product of other objects (or factors)  $\mathbf{F}_1, \dots, \mathbf{F}_k$ :

$$\mathbf{A} = \mathbf{F}_1 \mathbf{F}_2 \cdots \mathbf{F}_k ,$$

where the number of factor matrices  $k$  depends on the factorization. Most often,  $k = 2$  or  $k = 3$ . There are many different matrix decompositions; each one finds use among a particular class of problems. The most known matrix decompositions or factorizations include: LU, LDU, LUP, Block LU, Cholesky, QR, EVD, SVD, GSVD, Jordan, Jordan-Chevalley, Schur, QZ, Takagi, Polar, Rank factorization, and several forms of NMF.

Direct methods are computationally efficient only for a specific class of problems where the system matrix is sparse and reveals some patterns, e.g. it is a banded matrix. Otherwise, iterative methods are usually more efficient. This tutorial is restricted only to the fundamental direct methods that can be applied both to dense as well as to sparse linear systems.

### 3. Preparation.

The expected time needed for the preparation to this workshop is 9 hours.



### 3.1. Reading

- [1]. A. Bjorck, Numerical Methods for Least Squares Problems, SIAM, Philadelphia, 1996,
- [2]. G. Golub, C. F. Van Loan, Matrix Computations, The John Hopkins University Press, (Third Edition), 1996.
- [3]. J. Stoer R. Bulirsch, Introduction to Numerical Analysis (Second Edition), Springer-Verlag, 1993
- [4]. C. D. Meyer, Matrix Analysis and Applied Linear Algebra, SIAM, 2000,
- [5]. Ch. Zarowski, An Introduction to Numerical Analysis for Electrical and Computer Engineers, Wiley, 2004,
- [6]. G. Strang, Linear Algebra and Its Applications, Harcourt Brace & Company International Edition, 1998.
- [7]. N. W. Keith, Linear Algebra with Applications (Third Edition), PWS PUB Co, 1994 (Electronic Version).
- [8]. G. Lindfield, J. Penny, Numerical Methods Using MATLAB, Ellis Horwood, New York, 1995.

### 3.2. Problems

At the beginning of the laboratory workshop each student should know the answers to the following questions:

- Give examples of the fundamental direct methods for solving systems of linear equations.
- What is the aim of using pivoting in the Gaussian elimination (give examples)?
- How to apply the QR factorization for solving a system of linear equations?
- What is a computational cost of solving a system of size (N by N) with the Gaussian elimination?
- What is the Row Reduced Echelon Form (RREF) and how to obtain an inverse matrix with the RREF?
- What is the LU factorization and how to perform the Gaussian elimination with the LU decomposition?
- What is the difference between LU and LDU?
- What is the interpretation of diagonal entries in the matrix D of the LDU factorization?
- How to calculate a determinant of a matrix A using the LU factorization?
- To which class of matrices the Cholesky factorization can be applied?
- What is the relation of the LU with the Cholesky factorization?
- How can we check whether a matrix is positive-definite?
- How to solve the overdetermined system of linear equations using the QR and SVD factorizations?



### 3.3. Detailed preparation

Each group of students (2 –3 persons) is expected to accomplish the following tasks:

1. solve the selected linear problems analytically and roughly calculate a computational cost,
2. code the selected algorithms for matrix factorizations in Matlab,
3. apply the coded algorithms to the selected problems,
4. compare the results with built-in functions in Matlab,
5. draw the conclusions.

#### • Problems to be analyzed

The following problems should be solved with the coded algorithms.

**Problem 1:** Solve the system and find the pivots when:

$$\begin{cases} 2u - v = 0 \\ -u + 2v - w = 0 \\ -v + 2w - z = 0 \\ -w + 2z = 5 \end{cases}.$$

**Problem 2:** Solve the following system of linear equations using the Gaussian elimination with pivoting:

$$\begin{cases} x_1 + x_2 + x_3 = 1 \\ x_1 + x_2 + 2x_3 = 2 \\ x_1 + 2x_2 + 2x_3 = 1 \end{cases}.$$

Explain why the Gaussian elimination without pivoting does not work.

**Problem 3:** Solve the system:

$$\begin{cases} 0.0001x_1 + x_2 = 1, \\ x_1 + x_2 = 2, \end{cases}$$

with the Gaussian elimination with and without pivoting at the roundoff error limited to 3 significant digits. Compute the condition number of the system matrix.

**Problem 4:** Solve the system of linear equations:

$$\begin{cases} 0.835x_1 + 0.667x_2 = 0.168 \\ 0.333x_1 + 0.266x_2 = 0.067 \end{cases}$$

Then slightly perturb  $b_2$  from 0.067 to 0.066 and compute the solution to the perturbed system. Explain the change in the solution by computing the condition number of the system matrix.



**Problem 5:** Find  $\mathbf{A}^{-1}$  to

$$\mathbf{A} = \begin{bmatrix} 2 & 1 & 2 \\ 1 & 2 & 3 \\ 4 & 1 & 2 \end{bmatrix}$$

solving the system  $\mathbf{AX} = \mathbf{I}_3$ .

**Problem 6:** Apply the LU factorization to the matrix

$$\mathbf{A} = \begin{bmatrix} 1 & 2 & 3 & 4 \\ -1 & 1 & 2 & 1 \\ 0 & 2 & 1 & 3 \\ 0 & 0 & 1 & 1 \end{bmatrix}.$$

Then calculate  $\det(\mathbf{A})$  using the matrix  $\mathbf{U}$ . Finally solve  $\mathbf{Ax} = \mathbf{b}$  for  $\mathbf{b} = [1 \ \dots \ 1]^T$ .

**Problem 7:** Let  $\mathbf{A} = [a_{ij}] \in \mathbb{R}^{N \times N}$ , with  $a_{ij} = \frac{1}{i+j-1}$  (Hilbert matrix). For  $N = 5$ , perform the LU factorization of the matrix  $\mathbf{A}$ . Then, compute  $\det(\mathbf{A})$ .

**Problem 8:** Let

$$\mathbf{A} = \begin{bmatrix} 1 & -1 & 0 & 0 \\ -1 & 2 & -1 & 0 \\ 0 & -1 & 2 & -1 \\ 0 & 0 & -1 & 2 \end{bmatrix}$$

Check whether the symmetric matrix  $\mathbf{A}$  is positive-definite. If so, apply the Cholesky factorization. Then, compute its inverse.

**Problem 9:** Let  $\mathbf{A}$  be the Pascal matrix of order 100 (*pascal* function in Matlab). Check whether the matrix is positive definite. If so, apply the Cholesky factorization and give interpretation of the obtained factor.

**Problem 10:** Transform the following matrix to the RREF, determine  $\text{rank}(\mathbf{A})$  and identify the columns corresponding to the basic and free variables.

$$\mathbf{A} = \begin{bmatrix} 1 & 2 & 2 & 3 & 1 \\ 2 & 4 & 4 & 6 & 2 \\ 3 & 6 & 6 & 9 & 6 \\ 1 & 2 & 4 & 5 & 3 \end{bmatrix}$$



## THE DEVELOPMENT OF THE POTENTIAL AND ACADEMIC PROGRAMMES OF WROCŁAW UNIVERSITY OF TECHNOLOGY

**Problem 11:** Compute the LU factorization for:

$$\mathbf{A} = \begin{bmatrix} 1 & 3 & 3 & 2 \\ 2 & 6 & 9 & 5 \\ -1 & -3 & 3 & 0 \end{bmatrix}.$$

Determine a set of basic variables and a set of free variables, and find a homogeneous solution to  $\mathbf{Ax} = \mathbf{0}$ . What is the rank of  $\mathbf{A}$ ?

**Problem 12:** Compute the QR factorization of the matrix:

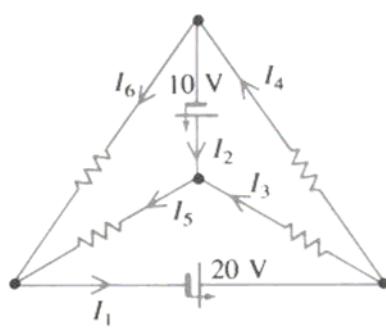
$$\mathbf{A} = \begin{bmatrix} 0 & -1 & -3 \\ 0 & 0 & -2 \\ 0 & -2 & -1 \end{bmatrix}.$$

How many flops (multiplications/divisions, additions/subtractions) are needed to perform the QR factorization with the Householder transformations and Givens rotations?

**Problem 13:** Let  $\mathbf{Ax} = \mathbf{b}$ , where  $\mathbf{A} = \mathbf{I}_N \otimes \mathbf{C}^T \mathbf{C}$ , the symbol  $\otimes$  denotes the Kronecker product,  $\mathbf{I}_N \in \mathbb{R}^{N \times N}$  is an identity matrix,  $\mathbf{C} \in \mathbb{R}^{M \times M}$  is a random matrix with a uniform distribution,  $M = 100$ , and  $N = 50$ , and  $\mathbf{x} \sim N(\mathbf{0}, \mathbf{I}_{MN})$ . Find the direct method that solves the above system of linear equations with the lowest computational cost. Estimate the cost with a roughly calculated number of flops and with the elapsed time.

**Problem 14:** Solve  $\mathbf{Ax} = \mathbf{b}$ , where  $\mathbf{A} \in \mathbb{R}^{10 \times 10}$  is the Hilbert matrix, and  $\mathbf{x} \sim N(\mathbf{0}, \mathbf{I}_{10})$ , with the Gaussian elimination in Matlab ( $\mathbf{x} := \mathbf{A}\backslash\mathbf{b}$ ). Then make a small change in an entry of  $\mathbf{A}$  or  $\mathbf{b}$ , and compare the solutions.

**Problem 15:** Find the currents in the circuit, assuming all resistances are 10 Ohm.



### • Algorithms to be coded

Students are expected to code the selected algorithms and apply them to the above-mentioned problems.



## THE DEVELOPMENT OF THE POTENTIAL AND ACADEMIC PROGRAMMES OF WROCŁAW UNIVERSITY OF TECHNOLOGY

**Algorithm 1:** Gaussian elimination, e.g. [2] Section 3.2.6. Algorithm 3.2.1., pp. 98 - 99; [6] Section 1, pp. 1 – 60; [7] Section 1.2, pp. 13 – 22.

**Algorithm 2:** Gaussian elimination with Complete Pivoting: [2] Section 3.4.8, Algorithm 3.4.2. pp. 117 – 120.

**Algorithm 3:** Forward Substitution: [2] Section 3.1, Algorithms 3.1.1. and 3.1.3, pp. 88 – 90.

**Algorithm 4:** Back Substitution: [2] Section 3.1, Algorithms 3.1.2. and 3.1.4, pp. 88 – 90.

**Algorithm 5:** The Gauss-Jordan elimination algorithm: [2] Section 1.3, pp. 23 – 27.

**Algorithm 6:** The RREF algorithm: [2] Section 1.4, pp. 32 – 38.

**Algorithm 7:** The LU factorization without pivoting: [2] Section 3.2. pp. 94 – 102.

**Algorithm 8:** The LU factorization with partial pivoting: [2] Section 3.4.5. pp. 114 – 115.

**Algorithm 9:** A family of the Cholesky factorization algorithms (column-wise, row-wise, outer product), e.g. in [1] Chapter 2, Section 2.2.2, pp. 44-48, Algorithms 2.2.1 – 2.2.3.

**Algorithm 10:** The Cholesky factorization: [2] Section 4.2.4., pp. 143 – 146.

**Algorithm 11:** The QR algorithm by the Householder transformation, e.g. in [1] Chapter 2, Sections 2.3-2.4, pp. 51-63, Algorithms 2.4.1.

**Algorithm 12:** The QR algorithm by the Givens rotations, e.g. in [1] Chapter 2, Sections 2.3-2.4, pp. 51-63, Algorithms 2.4.2.

**Algorithm 13:** The QR algorithm by the Gram-Schmidt orthogonalization, e.g. in [1] Chapter 2, Sections 2.4.2, pp. 60-63, Algorithms 2.4.5.

## 4. Content of report

The report should contain:

- introductory page,
- detailed mathematical description of the analyzed problems,
- basic description of the coded algorithms,
- the Matlab code (together with the detailed end-line comments) of the analyzed algorithm,
- the results obtained with the coded algorithms,
- the results obtained with the corresponding Matlab in-built functions,
- conclusions

The section “Results” should present final solutions, a convergence analysis, a computational cost with respect to flops and elapsed time, and the comparison to the results obtained with the Matlab built-in functions.