ASSIGNMENT A: SOLVING LINEAR ALGEBRAIC EQUATIONS

Numerical Methods (ENUME 2019) - Project

Ernest Pokropek
293076

Under the supervision of Andrzej Miękina, PhD

Table of contents

I. Introduction
II. Methodology
III. Results
IV. Conclusions
V. Bibliography

I. Introduction

Depending on the values that matrix consists of, its properties differ. Thus, various operations on it, may result in varied numerical errors, especially when solving sets of equations. In the first section of the following text, dependence of determinant (**det**) and condition number of inversion (**cond**) of the matrix is to be analysed, taken into consideration matrix $\boldsymbol{A}_{N,x}$ of the following form:

$$\mathbf{A}_{N,x} = \begin{bmatrix} x^2 & -\frac{3x}{2} & \frac{3x}{2} & -\frac{3x}{2} & \cdots & \frac{3x}{(-1)^{N-2}\cdot 2} & \frac{3x}{(-1)^{N-1}\cdot 2} \\ -\frac{3x}{2} & \frac{18}{4} & -\frac{18}{4} & \frac{18}{4} & \cdots & \frac{18}{(-1)^{N-3}\cdot 4} & \frac{18}{(-1)^{N-2}\cdot 4} \\ \frac{3x}{2} & -\frac{18}{4} & \frac{27}{4} & -\frac{27}{4} & \cdots & \frac{27}{(-1)^{N-4}\cdot 4} & \frac{27}{(-1)^{N-3}\cdot 4} \\ -\frac{3x}{2} & \frac{18}{4} & -\frac{27}{4} & \frac{36}{4} & \cdots & \frac{36}{(-1)^{N-5}\cdot 4} & \frac{36}{(-1)^{N-4}\cdot 4} \\ \frac{3x}{2} & -\frac{18}{4} & \frac{27}{4} & -\frac{36}{4} & \cdots & \vdots & \vdots \\ \vdots & \vdots & \vdots & \vdots & & \frac{(N-1)\cdot 9}{4} & -\frac{(N-1)\cdot 9}{4} \\ \frac{3x}{(-1)^{N-1}\cdot 2} & \frac{18}{(-1)^{N-2}\cdot 4} & \frac{27}{(-1)^{N-3}\cdot 4} & \frac{36}{(-1)^{N-4}\cdot 4} & \cdots & -\frac{(N-1)\cdot 9}{4} & \frac{N\cdot 9}{4} \end{bmatrix}$$

Fig. 1). Scheme of the considered matrix

In further analysis, two methods of inverting the matrix $A_{N,x}$ are presented:

a) Based on the LU factorisation:

"lower-upper" decomposition factoring a matrix as the product of a lower triangular matrix and an upper triangular one. Introduced by mathematician Tadeusz Banachiewicz in 1938. [1]

b) Based on the LLT factorisation (Cholesky decomposition):

a decomposition of a Hermitian, positive-definite matrix into the product of a lower triangular matrix and its conjugate transpose (as only the real entries of matrix A are considered here, the conjugate will not be referred). When it is applicable, the Cholesky decomposition is roughly twice as efficient as the LU decomposition for solving systems of linear equations.[2]

II. Methodology

The determinants and condition numbers were calculated for interval $\alpha \in [\alpha_N - 0.01,\ \alpha_N + 0.01\ ]$ consisting of 2000 points, where α_N is the smallest positive value of α which yields:

$$det(A_{N,x}) = 0 \ for\ x = \sqrt{\alpha^2 + \frac{1}{2}} - 1 \tag{1.1}$$

As the determinant is the algebraic sum of products of the $n \times n$ matrix and deriving from fact that each of those products consists at least one element either from first row or column, zeroing the whole first column and row will result in determinant equal to 0. Thus, the equation follows:

$$\sqrt{{\alpha_N}^2 + \frac{1}{2}} - 1 = 0 \tag{1.2}$$

Solving this equation using MATLAB's ***fzero*** function, the smallest positive value of α (α_N) is equal to $\frac{\sqrt{2}}{2}$.

Calculating the inverse of the matrix A, the following equation is used:

$$A \cdot A^{-1} = I \tag{2.1}$$

Denoting A^{-1} matrix's columns as y_N and identity matrix's columns as e_N, where N is the total number of columns in consecutive matrices, systems of N linear equations derived from equation 1.1 can be described as:

$$A \cdot y_N = e_N \tag{2.2}$$

And from the LU factorisation follows:

$$L \cdot U \cdot y_N = e_N \tag{2.3}$$

The LLT decomposition holds as follows:

$$A = L \cdot L^T \tag{3.1}$$

Using the equation 2.2 describing the set of linear equations in inversed matrix, it may be rewritten in the following form:

$$L \cdot L^T \cdot y_N = e_N \tag{3.2}$$

Correctness of implement inversion algorithms were tested by comparing them to MATLAB's operator ***inv***.

In order to test reliability of both methods of inversion, following norms were calculated for matrices of form $A_{N,x}$ and their inverse estimates $\widehat{A}_{N,x}^{-1}$, with $N \in \{3, 10, 20\}$ and $x = \frac{2^k}{300}, k \in \{0, 1, 2, \ldots, 21\}$:

- $\delta_2 = \left\| A_{N,x} \cdot \hat{A}_{N,x}^{-1} - I_N \right\|_2$ $(the\ root - mean - square\ error\ (RMSE))$ (4.1)
- $\delta_\infty = \left\| A_{N,x} \cdot \hat{A}_{N,x}^{-1} - I_N \right\|_\infty$ $(the\ maximum\ error\ (ME))$ (4.2)

Finally, the created norms were tested by comparing them to operator ***norm*** available in MATLAB.

III. Results

Sample generated matrix $A_{7,5}$

```
>> A = generate_matrix(7, 5)

A =

   25.0000   -7.5000    7.5000   -7.5000    7.5000   -7.5000    7.5000
   -7.5000    4.5000   -4.5000    4.5000   -4.5000    4.5000   -4.5000
    7.5000   -4.5000    6.7500   -6.7500    6.7500   -6.7500    6.7500
   -7.5000    4.5000   -6.7500    9.0000   -9.0000    9.0000   -9.0000
    7.5000   -4.5000    6.7500   -9.0000   11.2500  -11.2500   11.2500
   -7.5000    4.5000   -6.7500    9.0000  -11.2500   13.5000  -13.5000
    7.5000   -4.5000    6.7500   -9.0000   11.2500  -13.5000   15.7500
```

Fig. 2). Sample generated matrix for N = 7 and x = 5

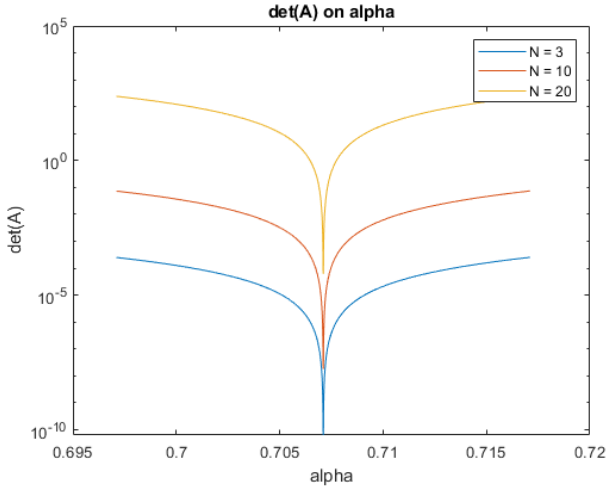

Fig. 3.1). Relation of determinant of matrix $A_{N,x}$ on values of alpha

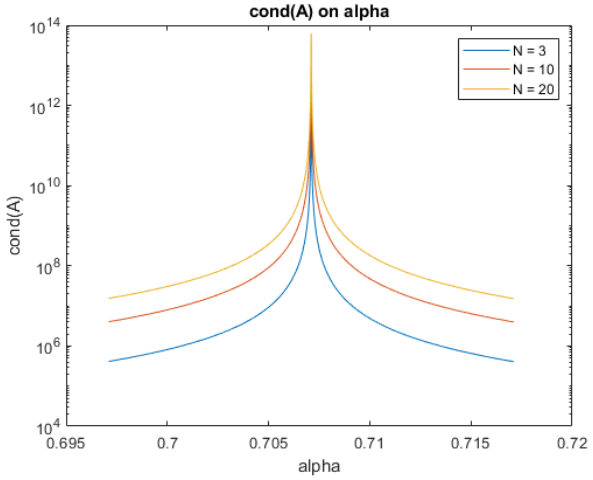

Fig. 3.2). Relation of condition number of inversion of matrix $A_{N,x}$ on values of alpha

From figure 3.1, the determinant of the created matrix can be depicted to be 0 indeed around the approximation of $\frac{\sqrt{2}}{2}$ (≈ 0.707). The value of the determinant decreases exponentially around the found α_N (tends to zero), as every element of first row and column of $A_{N,x}$ approaches 0 itself. For bigger sizes (N) of the matrix, the determinant has higher values, and its 'zeroing' is less precise, due to more non-zero elements in products. It is worthy to notice, that the condition number of inversion changes in the similar exponential manner, yet it increases. The differences between the examined parameter in terms of values of N is less remarkable, and though the lines does not intersect, they are much closer than for the determinant to alpha relation (Fig. 3.1). Around the α_N, the condition number of inversion tends to infinity, as columns and rows are beginning to form linear combinations (it also explains the similarity in behaviour of Fig. 3.1 and Fig. 3.2, as the determinant equal to 0 means linear dependence of the elements consisting the matrix). The largest value of the examined parameter is for the largest matrix, as there are more linearly dependent elements.

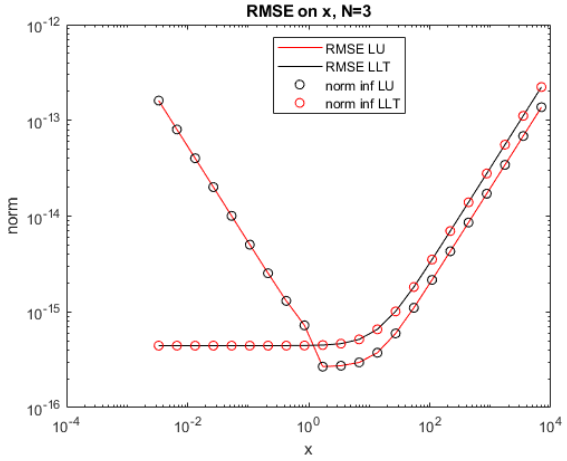

Fig. 4.1). Relation of root mean square error of matrix $A_{3,x}$ and MATLAB's second norm on values of x

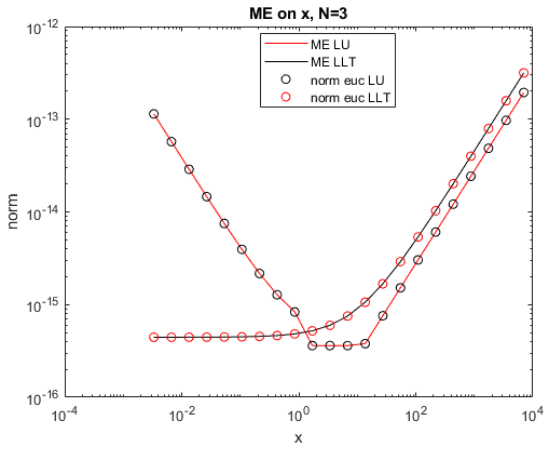

Fig. 4.2). Relation of maximal error of matrix $A_{3,x}$ and MATLAB's infinite norm on values of x

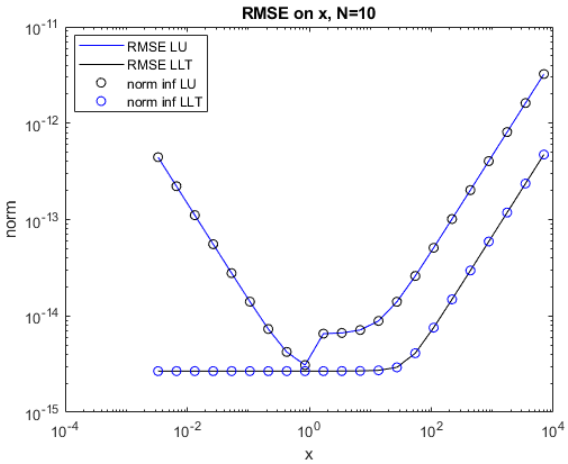

Fig. 4.3). Relation of root mean square error of matrix $A_{10,x}$ and MATLAB's second norm on values of x

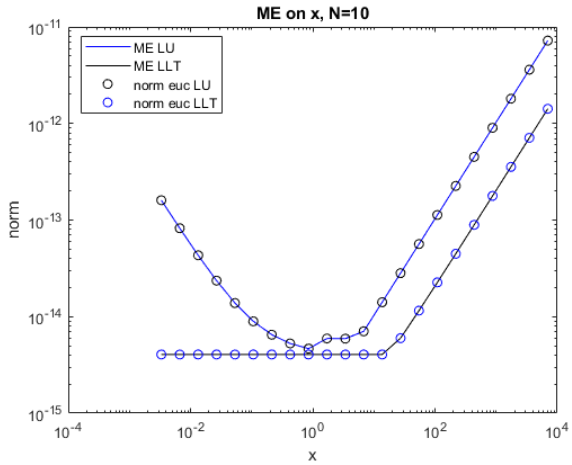

Fig. 4.4). Relation of maximal error of matrix $A_{10,x}$ and MATLAB's infinite norm on values of x

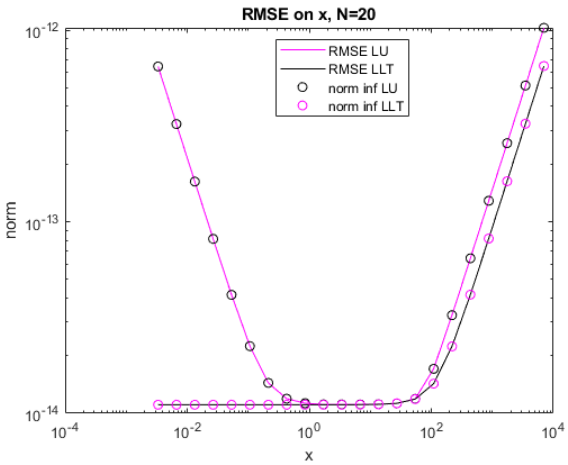

Fig. 4.5). Relation of root mean square error of matrix $A_{20,x}$ and MATLAB's second norm on values of x

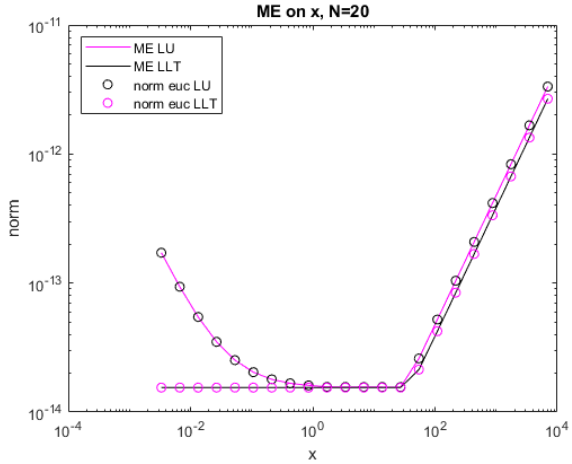

Fig. 4.6). Relation of maximal error of matrix $A_{20,x}$ and MATLAB's infinite norm on values of x

For each $x = \frac{2^k}{300}, k \in \{0, 1, 2, \ldots, 21\}$ the matrix $A_{N,x}$ has been generated, considering $N \in \{3, 10, 20\}$, and the norms were computed using $\hat{A}_{N,x}^{-1}$, inverted either using LU or LLT decomposition. The results are presented on figures indexed from 4.1 up to 4.6, consecutively.

Noticeably, for each size of the matrix, the value of error (either RMSE or ME) for given algorithm of decomposition is almost identical to the corresponding MATLAB's norm. Generally, RMSE for any matrix has steeper form, and for bigger matrices its value for small x is significantly larger. Also, there is a bigger difference between LU and LLT inversing algorithms in terms of RMSE, than for ME. For small matrices (N = 3), the differences are almost negligible.

Considering LU factorization, the value of ME for matrices of different sizes differ in much less extent than RMSE, for $x < 0.1$, due to rarely occurring outstanding errors. Meanwhile, for the same values of x, norms for both LU and LLT decompositions are of the same magnitude, and constant.

For smallest values of x, the norms concerning inverse using LLT factorization are smaller than those using LU, and are constant, while those using LU decomposition are decreasing. The smaller parameter N, the steeper and longer is the decrease. For matrices of smallest size (N=3), The inversion using LLT decomposition is remarkably better than one using LU, yet it changes for x larger than 1, as the lines intersect. After few iterations, the difference between inversion using LU and LLT is negligible, with LU being slightly better. The biggest disparity of norms is visible for matrices of size N=10, as LU algorithm is constantly being worse numerically. The algorithm's norms does not intersect, and after reaching its optimal value, the difference between LU and LLT using inversions is of the biggest value. For the biggest matrices (N=20), the two norms begin to unify inter values for $x \sim 1$, and proceeds to be almost of the same magnitude for one third's of the graph. As the norms begin to increase, the difference between LU and LLT decomposition algorithms is the smallest of the all, with inversion using LLT factorization is being constantly better.

The smallest value of norms is noted for smallest matrices ($N = 3$), as it required the least number of operations to achieve inversion. Also, for those matrices, the condition number is of less magnitude than for bigger ones, which results in more stable inversion. There is no visible correlation between size and norm value for the matrix, as its magnitude for inversion algorithm using LU decomposition for N=10 is greater than both inversions for N=20.

Given matrix $A_{7,5}$:

```
A =

   25.0000   -7.5000    7.5000   -7.5000    7.5000   -7.5000    7.5000
   -7.5000    4.5000   -4.5000    4.5000   -4.5000    4.5000   -4.5000
    7.5000   -4.5000    6.7500   -6.7500    6.7500   -6.7500    6.7500
   -7.5000    4.5000   -6.7500    9.0000   -9.0000    9.0000   -9.0000
    7.5000   -4.5000    6.7500   -9.0000   11.2500  -11.2500   11.2500
   -7.5000    4.5000   -6.7500    9.0000  -11.2500   13.5000  -13.5000
    7.5000   -4.5000    6.7500   -9.0000   11.2500  -13.5000   15.7500
```

```
>> inv(A)

ans =

    0.0800    0.1333         0         0         0         0         0
    0.1333    0.8889    0.4444         0         0         0         0
         0    0.4444    0.8889    0.4444         0         0         0
         0         0    0.4444    0.8889    0.4444         0         0
         0         0         0    0.4444    0.8889    0.4444         0
         0         0         0         0    0.4444    0.8889    0.4444
         0         0         0         0         0    0.4444    0.4444

>> invertLU(A)

ans =

    0.0800    0.1333         0         0         0         0         0
    0.1333    0.8889    0.4444         0         0         0         0
         0    0.4444    0.8889    0.4444         0         0         0
         0         0    0.4444    0.8889    0.4444         0         0
         0         0         0    0.4444    0.8889    0.4444         0
         0         0         0         0    0.4444    0.8889    0.4444
         0         0         0         0         0    0.4444    0.4444

>> invertLLT(A)

ans =

    0.0800    0.1333         0         0         0         0         0
    0.1333    0.8889    0.4444         0         0         0         0
         0    0.4444    0.8889    0.4444         0         0         0
         0         0    0.4444    0.8889    0.4444         0         0
         0         0         0    0.4444    0.8889    0.4444         0
         0         0         0         0    0.4444    0.8889    0.4444
         0         0         0         0         0    0.4444    0.4444
```

Fig. 5). Comparing different inversion algorithms

Comparing every matrix from previous set to one inverted by MATLAB's inv, they were almost identical. The best results were obtained inverting matrices using LU algorithm, however it is worth to notice, that the obtained average difference is smaller for LU decomposition using algorithm, as MATLAB's operator **inv** uses LU factorisation by itself[3]. The average difference gets smaller with the bigger size of the inverted matrix.

Superiority of inversion using LLT algorithm prove norms, as for this decomposition are significantly smaller than for the algorithm using LU factorisation. This applies to bigger matrices, as for those of size 3x3, the norms are almost of the same magnitude. It is worth to notice, that for bigger matrices (N=20), the Euclidean (second) norm is more than twice the magnitude of its infinite norm for inversion using LLT, possibly due to fact that its norms are constant for some period of iterations, and then starts to increase.

IV. Conclusions

Condition number of inversion of certain matrix is strictly related to its determinant, as the two parameters are connected to linear dependence of the elements (row, columns) of the matrix. Either row or column linear dependence results in determinant tending to 0, and condition number of inversion to infinity, which is visible on figures 3.1 and 3.2. Considering the matrix consisting of linearly dependent elements (or being close to), its condition number of inversion, as well as determinant, will be larger for bigger sizes of this matrix.

When inversing given matrix A, using LU or LLT decomposition results in easier set of linear equations to solve. The LU using algorithm performs better for matrices of small sizes (N=3), when those matrices are far from linearly dependant. Contrary to this decomposition, algorithm using LLT factorization does not improve with changes of the parameter x (only eventually decreases in its performance), thus, it is more numerically stable. In overall performance, the LLT factorization is better choice for inversion of a matrix, however the matrix has to be symmetric positive definite, while LU decomposition will perform fine for broader range of matrices.

V. Bibliography

[1] Schwarzenberg-Czerny, A. (1995). "On matrix factorization and efficient least squares solution". Astronomy and Astrophysics Supplement. 110: 405

[2] Press, William H.; Saul A. Teukolsky; William T. Vetterling; Brian P. Flannery (1992). Numerical Recipes in C: The Art of Scientific Computing (second ed.). Cambridge University England EPress. p. 994.

[3] https://www.mathworks.com/help/matlab/ref/inv.html