GaussianEliminationMethodCpp

Software Documentation

Author: matiwa

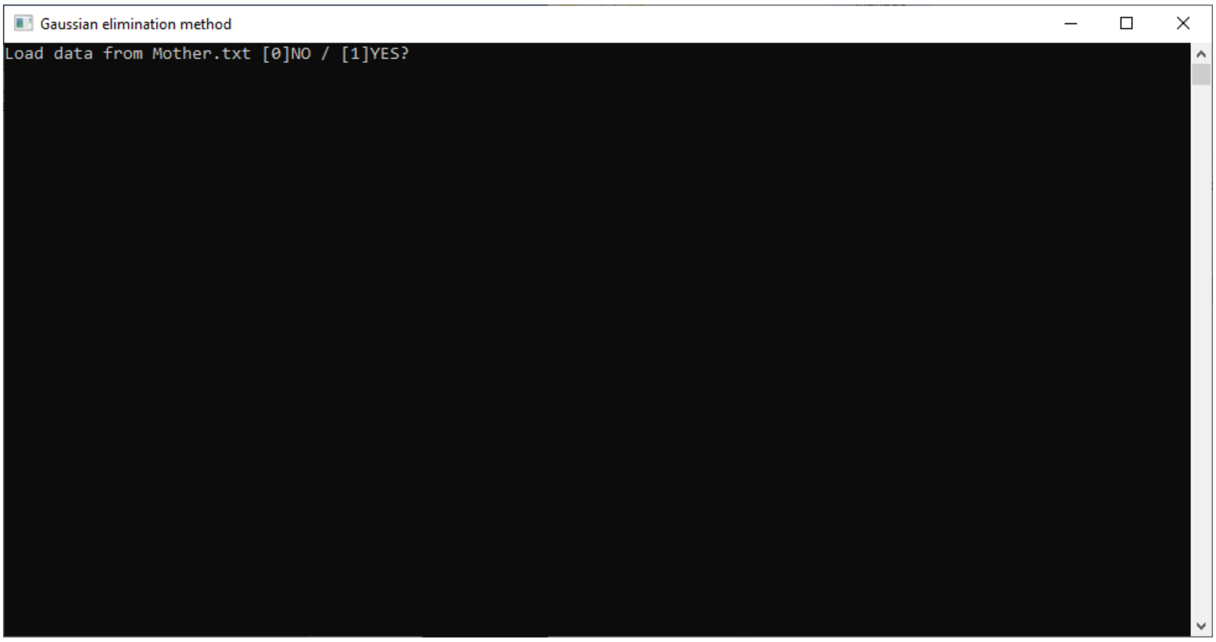
Table of contents

Introduction

This software documentation includes: description of the application's operation, what is needed for use, algorithms used, interface description and source code description. This application is used to calculate a system of n equations with n unknowns Ax = b.

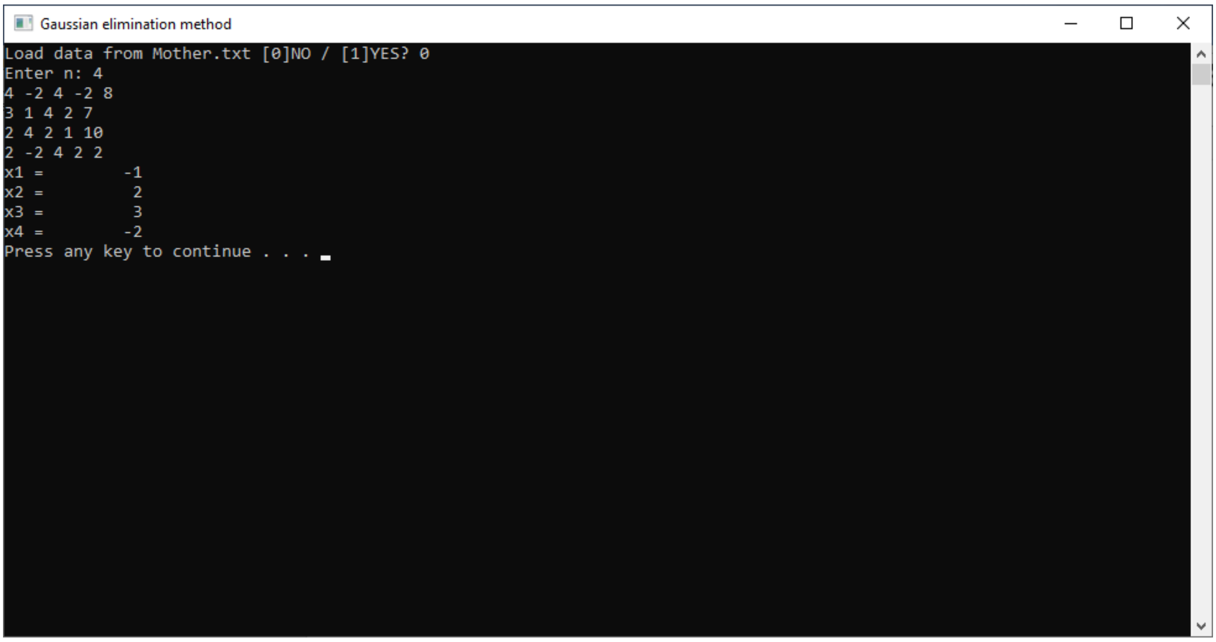
Describing of the application’s operation

If the user wants to calculate the solutions of the equation system by Gaussian elimination method, run the * .exe file. After this operation, the following console window will appear on the screen.

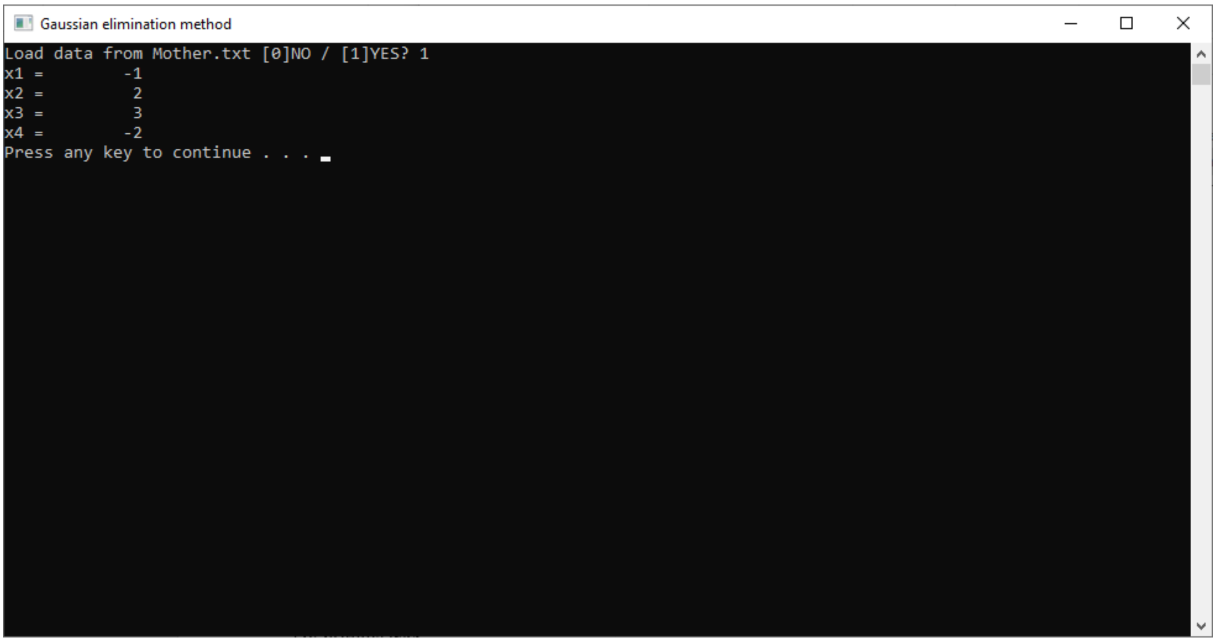


Drawing 1: The beginning of the application's operation [own study]

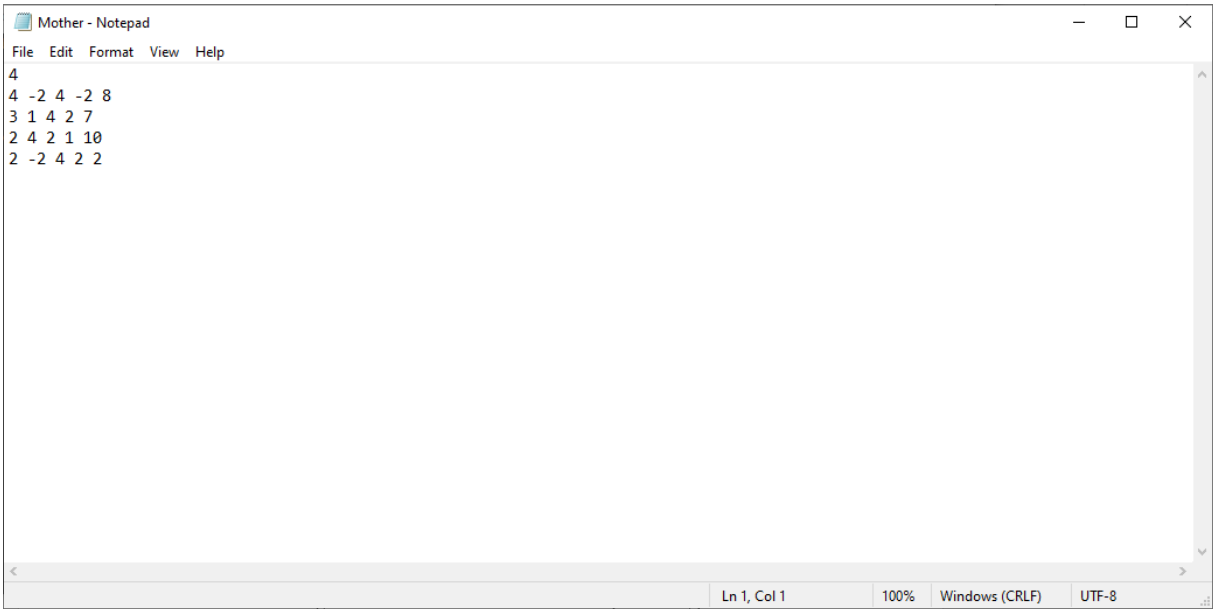
First, the user selects the program operation option. If he chooses 0, he will have to enter the number of unknowns in the console window first, then enter the values of matrix A and intercepts b. Then confirm with Enter. Finally, the searched values appear. When the action cannot be performed correctly, the message "The divisor of zero!" Appears.



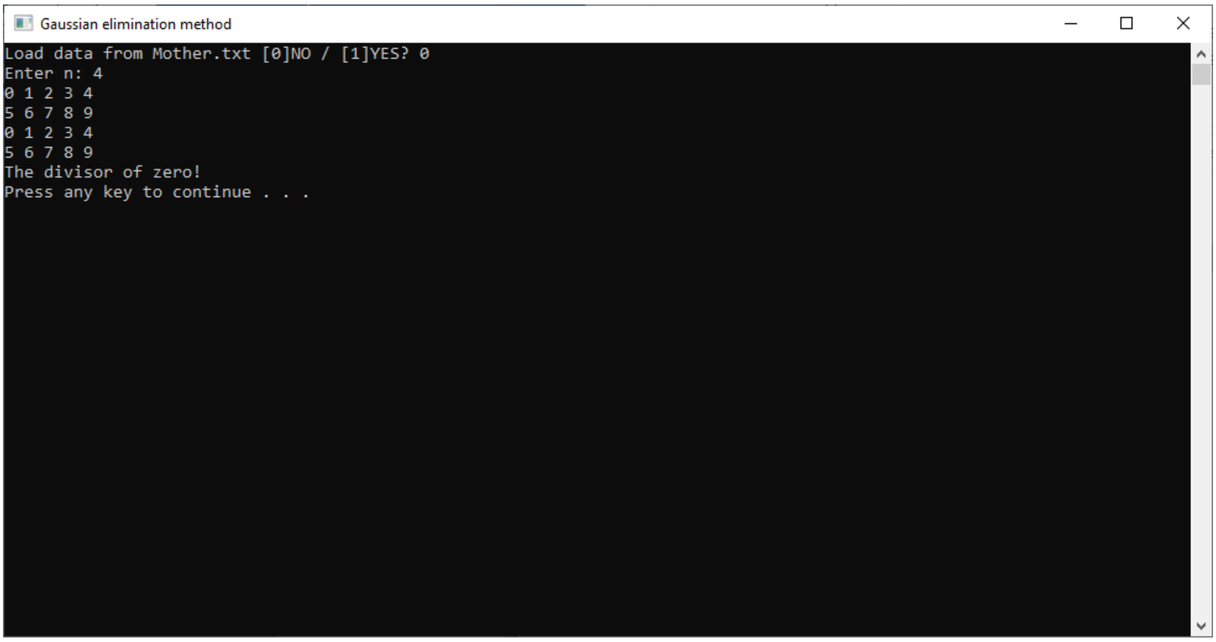
Drawing 2: The effect of the program for 0 option [own study]



Drawing 3: The effect of the program for 1 option [own study]



Drawing 4: The contents of the Mother.txt file [own study]



Drawing 5: Unable to calculate solutions [own study]

What is needed for use?

The application does not require installation. It only needs the Windows operating system.

Algorithms used

The algorithm used in the application is based on the Gaussian elimination method.

The Gaussian elimination method allows us to calculate a system of n equations with n unknowns Ax = b. We transform the matrix of coefficients A into the upper triangular matrix R, from which we then calculate the final solution - i.e. the vector x. The upper triangular matrix R is obtained as follows: intercepts b)

1. In the matrix (A, b) we look for an element ar1 other than zero and we go to the next point. If, however, there is no such element, it means that the matrix is singular and we cannot solve this system.
2. We replace the r-th and the first row.
3. We subtract from the i-th row of the matrix l the multiplicity of the i-th and first rows. This can be represented by the following formulas: (i = 2,3, ..., n j = 2,3, ..., n)

$$a_{ij} = a_{ij} - l_i \cdot a$$

$$b_i = b_i - l_i \cdot b_i$$

$$\text{gdzie } \; l_i = \frac{a_{i1}}{a_{11}}$$

Drawing 6: Formula [1]

4. Then we call this procedure from the first point recursively for the matrix (A ', b') - that is, the matrix (A, b) minus the first column and the first row.

$$(A, b) = \left| \begin{array}{c|cccc|} a_{11} & a_{12} & \dots & a_{1n} & b_n \\ a_{21} & a_{22} & \dots & a_{2n} & b_2 \\ \dots & \dots & \dots & \dots & \dots \\ a_{n1} & a_{n2} & \dots & a_{nn} & b_n \end{array} \right|$$

$$(A', b') = \left| \begin{array}{cccc} a_{22} & \dots & a_{2n} & b_2 \\ \dots & \dots & \dots & \dots \\ a_{n2} & \dots & a_{nn} & b_n \end{array} \right|$$

Drawing 7: Formula [1]

We can obtain a partial selection of the basic element by selecting the element, and in the first point, | ar1 | = max | ai1 | i = 1,2, ..., n

Full selection of the base element with | ars | = max | aij | i = 1,2, ..., n j = 1,2, ..., n We have to rearrange the r-th and first row and the s-th and first column (changes in columns should be remembered because it causes the replacement of unknowns!).

Once we transform the matrix of coefficients A to the upper triangular matrix R, and vector b to vector c, we can find the final solution from the formula: (i = n, n-1, ..., 1)

$$x_i = \frac{c_i - \sum_{k=i+1}^{n} r_{ik} x_k}{r_{ii}}$$

Drawing 8: Formula [1]

For example, let the matrix A and the vector b be given, from which we create the matrix (A, b)

$$(A, b) = \begin{vmatrix} 3 & 1 & 0 & -3 \\ 1 & 2 & -1 & 0 \\ 0 & -1 & 3 & 0 \end{vmatrix}$$

Drawing 9: Matrix (A,b) [1]

after transformations:

$$\begin{vmatrix} 3 & 1 & 0 & -3 \\ 0 & \frac{5}{3} & -1 & 1 \\ 0 & -1 & 3 & 0 \end{vmatrix}$$

Drawing 10: Matrix A [1]

then we call the procedure recursively for:

$$\begin{vmatrix} \frac{5}{3} & -1 & 1 \\ -1 & 3 & 0 \end{vmatrix}$$

Drawing 11: Example solution [1]

after transformations:

$$\begin{vmatrix} \frac{5}{3} & -1 & 1 \\ 0 & \frac{12}{5} & \frac{3}{5} \end{vmatrix}$$

Drawing 12: Example solution [1]

Since the last matrix can no longer be recursively called (after cutting off the top row and the right column, the vector will remain), so we have obtained the matrix R and vector c we are looking for, from which we determine the final solution

$$(R,\, c) = \begin{vmatrix} 3 & 1 & 0 & -3 \\ 0 & \frac{5}{3} & -1 & 1 \\ 0 & 0 & \frac{12}{5} & \frac{3}{5} \end{vmatrix}$$

Drawing 13: Matrix (R,c) [1]

x [3] = 0.25 x [2] = 0.75 x [1] = - 1.25 [1]

An example from Mother.txt shows the following solution:

Let us solve the above given Gaussian elimination method the following system of equations:

$$\begin{aligned} 4x_1 - 2x_2 + 4x_3 - 2x_4 &= 8 \\ 3x_1 + 1x_2 + 4x_3 + 2x_4 &= 7 \\ 2x_1 + 4x_2 + 2x_3 + 1x_4 &= 10 \\ 2x_1 - 2x_2 + 4x_3 + 2x_4 &= 2 \end{aligned}$$

Drawing 14: System of equations [2]

We begin the variable elimination stage:

- from equation 2 we subtract equation 1 multiplied by 3/4 = 0.75

- from equation 3 we subtract equation 1 multiplied by 2/4 = 0.5

- from equation 4 we subtract equation 1 multiplied by 2/4 = 0.5.

$$3x_1 + x_2 + 4x_3 + 2x_4 - 0{,}75 \cdot (4x_1 - 2x_2 + 4x_3 - 2x_4) = 7 - 0{,}75 \cdot 8$$
$$2x_1 + 4x_2 + 2x_3 + x_4 - 0{,}5 \cdot (4x_1 - 2x_2 + 4x_3 - 2x_4) = 10 - 0{,}5 \cdot 8$$
$$2x_1 - 2x_2 + 4x_3 + 2x_4 - 0{,}5 \cdot (4x_1 - 2x_2 + 4x_3 - 2x_4) = 2 - 0{,}5 \cdot 8$$

$$4x_1 - 2x_2 + 4x_3 - 2x_4 = 8$$
$$3x_1 + x_2 + 4x_3 + 2x_4 - 3x_1 + 1{,}5x_2 - 3x_3 + 1{,}5x_4 = 1$$
$$2x_1 + 4x_2 + 2x_3 + x_4 - 2x_1 + x_2 - 2x_3 + x_4 = 6$$
$$2x_1 - 2x_2 + 4x_3 + 2x_4 - 2x_1 + x_2 - 2x_3 + x_4 = -2$$

$$4x_1 - 2x_2 + 4x_3 - 2x_4 = 8$$
$$3x_1 - 3x_1 + x_2 + 1{,}5x_2 + 4x_3 - 3x_3 + 2x_4 = 1{,}5x_4 = 1$$
$$2x_1 - 2x_1 + 4x_2 + x_2 + 2x_3 - 2x_3 + x_4 + x_4 = 6$$
$$2x_1 - 2x_1 - 2x_2 + x_2 + 4x_3 - 2x_3 + 2x_4 + x_4 = -2$$

$$4x_1 - 2x_2 + 4x_3 - 2x_4 = 8$$

$$\begin{array}{rrrrl}
4{,}0x_1 - & 2{,}0x_2 + & 4{,}0x_3 - & 2{,}0x_4 & = 8 \\
 & 2{,}5x_2 + & x_3 + & 3{,}5x_4 & = 1 \\
 & 5{,}0x_2 + & & 2{,}0x_4 & = 6 \\
 & -x_2 + & 2{,}0x_3 + & 3{,}0x_4 & = -2
\end{array}$$

Drawing 15: Variable elimination stage [2]

Note that from Equations 2, 3, and 4, the unknown x 1 has disappeared because the coefficient for it has been reduced to zero. We eliminate the next unknowns:

- from equation 3 we subtract equation 2 multiplied by 5 / 2.5 = 2

- from equation 4 we subtract equation 2 multiplied by -1 / 2.5 = -0.4.

$$4x_1 - 2x_2 + 4x_3 - 2x_4 = 8$$
$$2{,}5x_2 + x_3 + 3{,}5x_4 = 1$$
$$5x_2 + 2x_4 - 2 \cdot (2{,}5x_2 + x_3 + 3{,}5x_4) = 6 - 2$$
$$-x_2 + 2x_3 + 3x_4 + 0{,}4 \cdot (2{,}5x_2 + x_3 + 3{,}5x_4) = -2 + 0{,}4$$

$$4x_1 - 2x_2 + 4x_3 - 2x_4 = 8$$
$$2{,}5x_2 + x_3 + 3{,}5x_4 = 1$$
$$5x_2 + 2x_4 - 5x_2 - 2x_3 - 7x_4 = 4$$
$$-x_2 + 2x_3 + 3x_4 + x_2 + 0{,}4x_3 + 1{,}4x_4 = -1{,}6$$

$$4x_1 - 2x_2 + 4x_3 - 2x_4 = 8$$
$$2{,}5x_2 + x_3 + 3{,}5x_4 = 1$$
$$5x_2 - 5x_2 - 2x_3 + 2x_4 - 7x_4 = 4$$
$$-x_2 + x_2 + 2x_3 + 0{,}4x_3 + 3x_4 + 1{,}4x_4 = -1{,}6$$

$$\begin{aligned}
4{,}0x_1 - 2{,}0x_2 + 4{,}0x_3 - 2{,}0x_4 &= 8{,}0 \\
2{,}5x_2 + 1{,}0x_3 + 3{,}5x_4 &= 1{,}0 \\
-2{,}0x_3 - 5{,}0x_4 &= 4{,}0 \\
2{,}4x_3 + 4{,}4x_4 &= -1{,}6
\end{aligned}$$

Drawing 16: Elimination of further unknowns [2]

Final Elimination:

- from equation 4 we subtract equation 3 multiplied by 2.4 / -2 = -1.2 with both sides:

$$
\begin{aligned}
&4x_1 - 2x_2 + 4x_3 - 2x_4 = 8\\
&2{,}5x_2 + x_3 + 3{,}5x_4 = 1\\
&-2x_3 - 5x_4 = 4\\
&2{,}4x_3 + 4{,}4x_4 + 1{,}2 \cdot (-2x_3 - 5x_4) = -1{,}6 + 1{,}2 \cdot 4
\end{aligned}
$$

$$
\begin{aligned}
&4x_1 - 2x_2 + 4x_3 - 2x_4 = 8\\
&2{,}5x_2 + x_3 + 3{,}5x_4 = 1\\
&-2x_3 - 5x_4 = 4\\
&2{,}4x_3 + 4{,}4x_4 - 2{,}4x_3 - 6x_4 = 3{,}2
\end{aligned}
$$

$$
\begin{aligned}
&4x_1 - 2x_2 + 4x_3 - 2x_4 = 8\\
&2{,}5x_2 + x_3 + 3{,}5x_4 = 1\\
&-2x_3 - 5x_4 = 4\\
&2{,}4x_3 - 2{,}4x_3 + 4{,}4x_4 - 6x_4 = 3{,}2
\end{aligned}
$$

$$
\begin{array}{rrrrl}
4{,}0x_1 & -2{,}0x_2 + & 4{,}0x_3 - & 2{,}0x_4 & = 8{,}0\\
 & 2{,}5x_2 + & 1{,}0x_3 + & 3{,}5x_4 & = 1{,}0\\
 & & -2{,}0x_3 - & 5{,}0x_4 & = 4{,}0\\
 & & & -1{,}6x_4 & = 3{,}2
\end{array}
$$

Drawing 17: Final elimination [2]

We start the reverse procedure. Going from the end, we set the next unknowns:

$$x_4 = \frac{3{,}2}{-1{,}6} = \mathbf{-2}$$

Drawing 18: Determining the unknowns [2]

Given x 4 we can calculate x 3 from equation 3:

$$x_3 = \frac{4 + 5x_4}{-2} = \frac{4 + 5 \cdot (-2)}{-2} = \frac{4 - 10}{-2} = \frac{-6}{-2} = \mathbf{3}$$

Drawing 19: Determining the unknowns [2]

We calculate x 2 from equation 2:

$$x_2 = \frac{1 - 3{,}5x_4 - x_3}{2{,}5} = \frac{1 - 3{,}5 \cdot (-2) - 3}{2{,}5} = \frac{1 + 7 - 3}{2{,}5} = \frac{5}{2{,}5} = \mathbf{2}$$

Drawing 20: Determining the unknowns [2]

And the last unknown x 1 we get from equation 1:

$$x_1 = \frac{8 + 2x_4 - 4x_3 + 2x_2}{4} = \frac{8 + 2 \cdot (-2) - 4 \cdot 3 + 2 \cdot 2}{4} = \frac{8 - 4 - 12 + 4}{4} = \frac{-4}{4} = \mathbf{-1}$$

Drawing 21: Determining the unknowns [2]

and finally we can save the solution: [2]

$$\begin{aligned} x_1 &= -1 \\ x_2 &= 2 \\ x_3 &= 3 \\ x_4 &= -2 \end{aligned}$$

Drawing 22: Solution [2]

Interface description

The interface is a console pane. The operation of the program is based on user communication. He gives the needed values on the input. The course of the process and possible operating errors are described in the chapter „Describing of the application's operation".

Source code description

The project was made in the C++ programming language, in the Dev-C++ programming environment. All work was done on the Windows 10 operating system. The application's source code looks like this.

```
#include <iostream>
#include <iomanip>
#include <cmath>
#include<windows.h>
#include<fstream>
```

```
using namespace std;

const double eps = 1e-12;
double **AB, *X;
int n, i, j;
int option;

int wczytaj(char *nplik)
{
	ifstream plik(nplik);
	plik >> n;
	AB = new double * [n];
	X  = new double [n];

	for( i = 0; i < n; i++ ) AB[i] = new double[n+1];

	for( i = 0; i < n; i++ )
		for( j = 0; j <= n; j++ ) plik >> AB [ i ][ j ];
	plik.close();
	return 0;
}

void pokaz(){
	for(int a=0;i<n;a++){
		for(int b=0;b<=n;b++){
			cout<<AB[i][j];
			if(b==n) cout<<endl;
			else if(b<n) cout<<" ";
		}
	}
}

bool gauss ( int n, double **AB, double *X)
{
	int i, j, k;
	double m, s;

	for( i = 0; i < n - 1; i++ ) {
	for( j = i + 1; j < n; j++ ) {
		if( fabs ( AB [ i ][ i ] ) < eps ) return false;
		m = -AB [ j ][ i ] / AB [ i ][ i ];
		for( k = i + 1; k <= n; k++ )
		AB [ j ][ k ] += m * AB [ i ][ k ];
	}
	}

	for(i = n-1; i>=0; i--) {
	s = AB [ i ][ n ];
```

```
	for( j = n - 1; j >= i + 1; j-- )
			s -= AB [ i ][ j ] * X [ j ];
	if( fabs ( AB [ i ][ i ] ) < eps ) return false;
	X [ i ] = s / AB [ i ][ i ];
	}
	return true;
}

int main( )
{
	SetConsoleTitleA("Gaussian elimination method");

	cout<<"Load data from Mother.txt [0]NO / [1]YES? ";
	cin>>option;

	if(option==0){
		cout<<"Enter n: ";
		cin>>n;

		AB = new double * [n];
		X  = new double [n];

		for( i = 0; i < n; i++ ) AB [ i ] = new double [ n + 1 ];

		for( i = 0; i < n; i++ )
			for( j = 0; j <= n; j++ ) cin >> AB [ i ][ j ];

	}else{
		wczytaj("Mother.txt");
	}

	pokaz();

	if(gauss(n,AB,X)){
		for( i = 0; i < n; i++ )
			cout<<"x"<<i+1<<" = "<<setw(9)<<X[i]<<endl;
	}
	else cout<<"The divisor of zero!\r\n";

	for( i = 0; i < n; i++ ) delete [ ] AB [ i ];
	delete [ ] AB;
	delete [ ] X;

	system("pause");
	return 0;
}
```

Listing 1: Source code [own study]

List of drawings

List of listings

Bibliography

[1] http://www.algorytm.org/procedury-numeryczne/metoda-eliminacji-gaussa.html
[2] https://eduinf.waw.pl/inf/alg/001_search/0076.php