

Ministerul Educației al Republicii Moldova
Universitatea Tehnică a Moldovei
Facultatea Calculatoare Informatică și Microelectronică
Departamentul Ingineria Software și Automatică

RAPORT

la lucrarea de laborator nr. 2
la disciplina Metode și modelele de calcul

Tema” Rezolvarea numerică a sistemelor liniare de ecuații “

A efectuat: st. gr. SI-181 Dimoglo Alexandr

A verificat: Șt Buzurniuc

Scopul lucrării:

- 1) Studiarea metodelor numerice pentru rezolvarea sistemelor de ecuații liniare.
- 2) Aplicarea practică a metodelor numerice pentru rezolvarea sistemelor de ecuații liniare

Sarcina lucrării:

Rezolvare sisteme a ecuațiilor liniare

$$(8.7+0.1N)X_1+11X_2-0.5X_3+(0.4+0.2N)X_4=10.2+N$$

$$(1.1+0.01N)X_1+9.6X_2+1.2X_3+(0.4+0.1N)X_4=43+2N$$

$$(-0.5+0.1N)X_1+1.2X_2+14.1X_3+(1.3+0.2N)X_4=8.6+N$$

$$(0.4+0.2N)X_1+0.4X_2+1.3X_3+(13.6+0.2N)X_4=0.9+3N$$

$$N=11.$$

Prin Metoda:

- a) Gauss
- b) Jacoby
- c) Gauss-Zeidel.

a) Cod Metodei Gauss.

```
double * gauss(double **a, double *y, int n)
```

```
{
```

```
    double *x, max;
```

```
    int k, index;
```

```
    const double eps = 0.00001; // точность
```

```
    x = new double[n];
```

```
    k = 0;
```

```
    while (k < n)
```

```
    {
```

```
        // Поиск строки с максимальным a[i][k]
```

```
        max = abs(a[k][k]);
```

```
        index = k;
```

```
        for (int i = k + 1; i < n; i++)
```

```
        {
```

```
            if (abs(a[i][k]) > max)
```

```
            {
```

```
                max = abs(a[i][k]);
```

```
                index = i;
```

```

    }
}

// Перестановка строк

if (max < eps)
{
    // нет ненулевых диагональных элементов

    cout << "Решение получить невозможно из-за нулевого столбца ";

    cout << index << " матрицы A" << endl;

    return 0;
}

for (int j = 0; j < n; j++)
{
    double temp = a[k][j];

    a[k][j] = a[index][j];

    a[index][j] = temp;
}

double temp = y[k];

y[k] = y[index];

y[index] = temp;

// Нормализация уравнений

for (int i = k; i < n; i++)
{
    double temp = a[i][k];

    if (abs(temp) < eps) continue; // для нулевого коэффициента пропустить

    for (int j = 0; j < n; j++)

        a[i][j] = a[i][j] / temp;

    y[i] = y[i] / temp;

    if (i == k) continue; // уравнение не вычитать само из себя

    for (int j = 0; j < n; j++)

        a[i][j] = a[i][j] - a[k][j];

    y[i] = y[i] - y[k];
}

k++;

```

```

    }

    // обратная подстановка

    for (k = n - 1; k >= 0; k--)

    {

        x[k] = y[k];

        for (int i = 0; i < k; i++)

            y[i] = y[i] - a[i][k] * x[k];

    }

    return x;

}

```

Datele de intrate și datele de ieșire:

```

C:\Users\stas1\OneDrive\Рабочий стол\ЭШТХ\Lab2MMC\Gauss.exe
Введите количество уравнений: 4
a[0][0]= 9.8
a[0][1]= 11
a[0][2]= -0.5
a[0][3]= 2.6
a[1][0]= 1.21
a[1][1]= 9.6
a[1][2]= 1.2
a[1][3]= 0
a[2][0]= 0.6
a[2][1]= 1.2
a[2][2]= 14.1
a[2][3]= 3.5
a[3][0]= 2.6
a[3][1]= 0.4
a[3][2]= 1.3
a[3][3]= 15.8
y[0]= 21.2
y[1]= -22.5
y[2]= 19.6
y[3]= 33.9
9.8*x0 + 11*x1 + -0.5*x2 + 2.6*x3 = 21.2
1.21*x0 + 9.6*x1 + 1.2*x2 + 0*x3 = -22.5
0.6*x0 + 1.2*x1 + 14.1*x2 + 3.5*x3 = 19.6
2.6*x0 + 0.4*x1 + 1.3*x2 + 15.8*x3 = 33.9
x[0]=5.27188
x[1]=-3.15751
x[2]=1.19429
x[3]=1.87129

```

b) Cod Metodei Jacoby.

```
void Jacobi ( double** A, double* F, int N, double* X)
```

```

{

    double* TempX = new double[N];

    double* TempY = new double[N];

```

```

double *z = new double[N];

double norm; // норма, определяемая как наибольшая разность компонент столбца иксов соседних итераций.

for (int i=0; i<N; i++){

    z[i]=A[i][i];

}

//for (int g=0; g<4; g++){

for (int i=0; i<N; i++){

    //cout<<"Temp "<<i<<"=";

    for (int j=0; j<N; j++){

        if(i!=j){

            TempX[i]=TempX[i]+(-A[i][j]/z[i]*X[j]);

            cout<<-A[i][j]<<"/"<<z[i]<<"* "<<X[j]<<endl;

        }

    }

    // cout<<"\n";

    TempX[i]=TempX[i]+X[i];

}

for (int i=0; i<N; i++){

    X[i]=TempX[i];

}

}

```

Datele de intrare:

```

C:\Users\stas1\OneDrive\Рабочий стол\еСюб\1зштхЕ\Lab2MMC\jacoby.exe
Введите количество уравнений: 4
a[0][0]= 9.8
a[0][1]= 11
a[0][2]= -0.5
a[0][3]= 2.6
a[1][0]= 1.21
a[1][1]= 9.6
a[1][2]= 1.2
a[1][3]= 0
a[2][0]= 0.6
a[2][1]= 1.2
a[2][2]= 14.1
a[2][3]= 3.5
a[3][0]= 2.6
a[3][1]= 0.4
a[3][2]= 1.3
a[3][3]= 15.8
y[0]= 21.2
y[1]= -22.5
y[2]= 19.6
y[3]= 33.9
9.8*x0 + 11*x1 + -0.5*x2 + 2.6*x3 = 21.2
1.21*x0 + 9.6*x1 + 1.2*x2 + 0*x3 = -22.5
0.6*x0 + 1.2*x1 + 14.1*x2 + 3.5*x3 = 19.6
2.6*x0 + 0.4*x1 + 1.3*x2 + 15.8*x3 = 33.9

```

Datele de ieşire

```

x[0]=-4.29569
x[1]=-2.79017
x[2]=0.964897
x[3]=1.73455

```

c) Cod Metodei Gauss-Zeidel:

```
void GaussZeidel( double** A, double* F, int N, double* X)

{

    double* TempX = new double[N];

    double* TempY = new double[N];

    double *z = new double[N];

    double norm;

    for (int i=0; i<N; i++){

        z[i]=A[i][i];

    }

    for (int i=0; i<N; i++){

        for (int j=0; j<N; j++){

            if(i!=j){

                TempX[i]=TempX[i]+(-A[i][j]/z[i]*X[j]);

            }

        }

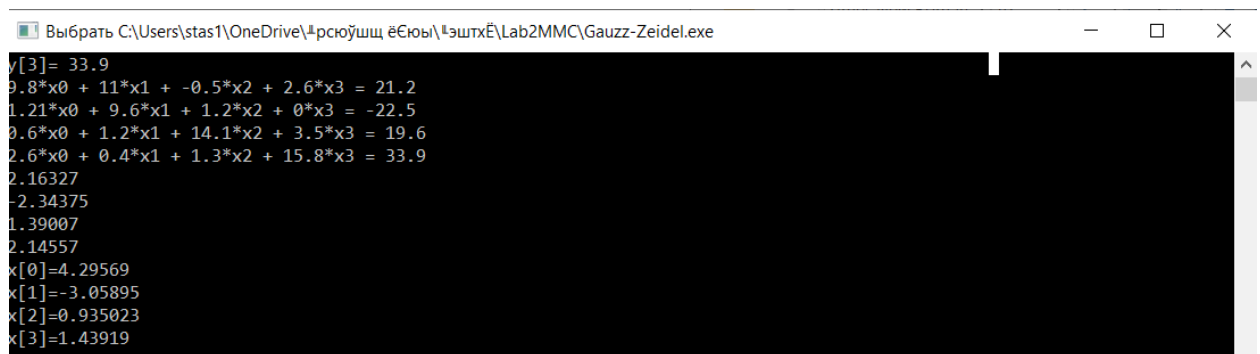
        TempX[i]=TempX[i]+X[i];

        X[i]=TempX[i];

    }

}
```

Datele de intrare si datele de iesire



```
Выбрать C:\Users\stas1\OneDrive\Рабочий стол\ёСюы\ЛэштхЕ\Lab2MMC\Gauzz-Zeidel.exe
y[3]= 33.9
9.8*x0 + 11*x1 + -0.5*x2 + 2.6*x3 = 21.2
1.21*x0 + 9.6*x1 + 1.2*x2 + 0*x3 = -22.5
0.6*x0 + 1.2*x1 + 14.1*x2 + 3.5*x3 = 19.6
2.6*x0 + 0.4*x1 + 1.3*x2 + 15.8*x3 = 33.9
2.16327
-2.34375
1.39007
2.14557
x[0]=4.29569
x[1]=-3.05895
x[2]=0.935023
x[3]=1.43919
```

Concluzei:

În urma lucrării de laborator, am aflat despre metode numerice pentru rezolvarea sistemelor de ecuații liniare. În practică, am aplicat metoda Gauss, Jacobi și Gauss-Seidel. Am dezvoltat un algoritm de soluție pentru fiecare metoda, am scris cod C++ și am găsit rădăcinile ecuației.