**Ministerul Educației, Culturii și Cercetării**

**Universitatea Tehnică a Moldovei**

**Facultatea Calculatoare, Informatică și Microelectronică**

**Departamentul Ingineria Software și Automatică**

**Raport**

Lucrarea de laborator nr.2

Disciplina: Metode și modele de calcul

Tema: Rezolvarea numerică a sistemelor de ecuații liniare.

**Efectuat**: st.gr.TI-207 Bunescu Gabriel.

**Verificat**: conf. univ. dr. Dohotaru Leonid

Chișinău 2021

**Scopul lucrării:**

1. Să se rezolve sistemul de ecuații liniare Ax=b utilizîndȘ

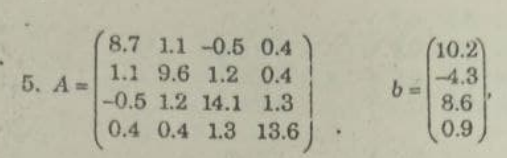
Metoda eliminării a lui Gauss

Metoda iterativă a lui Jacobi cu o eroare E=10-3

Metoda iterativa a lui Gauss-Seidel cu o eroare E=10-3, E=10-5.

1. Să se determine numărul de iterații necesare pentru aproximarea soluției sistemului cu eroarea dată E. Să se compare rezultatele obținute.

**Varianta:5**

****

**Codul sursă:**

**Metoda eliminării a lui Gauss**

**#include <iostream>**

**#include<conio.h>**

**#include<stdlib.h>**

**using namespace std;**

**float a[10][10],b[10],x[10],dp;**

**int i,j,n,iv,t,k,l;**

**float temp;**

**int main()**

**{**

**cout<<"Dati numarul de ecuatii ";cin>>n;**

**cout<<"Dati matricea A "<<endl;**

**for(i=1;i<=n;i++)**

**for(j=1;j<=n;j++)**

**{**

**cout<<"a["<<i<<"]["<<j<<"]= ";**

**cin>>a[i][j];**

**}**

**cout<<"Dati termenul liber"<<endl;**

**for(i=1;i<=n;i++)**

**{**

**cout<<"b["<<i<<"]= ";**

**cin>>b[i];**

**}**

**for(j=1;j<=n-1;j++)**

**{**

**iv=j;**

**t=1;**

**while( (iv<=n) && (t==1) )**

**if(a[iv][j] == 0) iv=iv+1;**

**else t=0;**

**if(t==1)**

**{**

**cout<<"Determinantul pricipal este nul ";**

**exit(0);**

**}**

**if(j!=iv)**

**{**

**for(k=j;k<=n;k++)**

**{**

**temp=a[j][k];**

**a[j][k]=a[iv][k];**

**a[iv][k]=temp;**

**}**

**// schimbam si elementul corespunzator liniei k al termenului liber**

**temp=b[j];**

**b[j]=b[iv];**

**b[iv]=temp;**

**}**

**for(l=j+1;l<=n;l++)**

**{**

**for(k=j+1;k<=n;k++)**

**a[l][k]=a[l][k] - a[j][k] \* a[l][j] / a[j][j];**

**// calcul pt. termenul liber**

**b[l]=b[l] - b[j] \* a[l][j] / a[j][j];**

**}**

**}**

**if(a[n][n]==0)**

**{**

**cout<<"Deterninantul principal este nul";**

**exit(0);**

**}**

**// calculul lui xn**

**x[n]=b[n] / a[n][n];**

**for(i=n-1;i>=1;i--)**

**{**

**temp = b[i];**

**for(j=i+1;j<=n;j++) temp = temp - a[i][j] \* x[j];**

**x[i] = temp / a[i][i];**

**}**

**for(i=1;i<=n;i++)**

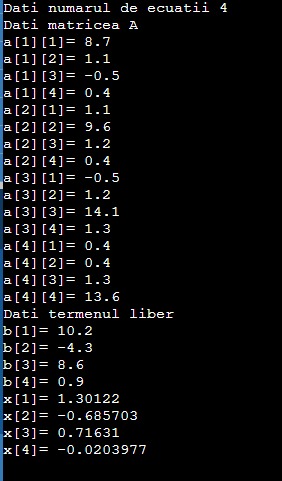
**cout<<"x["<<i<<"]= "<<x[i]<<endl;**

**getch();**

**return 0;**

**}**

**Rezultatul obținut:**

****

**Codul sursă:**

**Metoda iterativă a lui Jacobi cu o eroare E=10-3**

**#include <iostream>**

**#include<stdlib.h>**

**#include<conio.h>**

**#include<math.h>**

**using namespace std;**

**double a[10][10],x[10],b[10],temp[10];**

**// a este matricea sistemului**

**// x vectorul de necunoscute ale sistemului**

**// b vectorul de termeni liberi**

**// temp un vector pentru calcul al unor valori temporare double q,eps,nb,s;**

**// q numar egal cu norma matricei I-A ( adica q = |I - A| )**

**// eps eroarea de calcul aproximativ al solutiilor sistemului**

**// nb numar egal cu norma vectorului B ( adica nb = |B| )**

**int n,m,j,i,iteratii;**

**int main()**

**{**

**cout<<"Intoduceti nr. de linii si de coloane ale matricei sistemului ";**

**cin>>n;**

**for(i=1;i<=n;i++)**

**for(j=1;j<=n;j++)**

**{**

**cout<<"a["<<i<<","<<j<<"]= ";**

**cin>>a[i][j];**

**}**

**cout<<"Introduceti elementele termenului liber "<<endl;**

**for(i=1;i<=n;i++)**

**{**

**cout<<"b["<<i<<"]= ";**

**cin>>b[i];**

**}**

**cout<<"Sistemul initial este :"<<endl;**

**for(i=1;i<=n;i++)**

**{**

**cout<<a[i][1]<<" x"<<1;**

**for(j=2;j<=n;j++)**

**cout<<" + "<<a[i][j]<<" x"<<j;**

**cout<<" = "<<b[i]<<endl;**

**}**

**cout<<"are solutiile :"<<endl;**

**double eps=0.0001;**

**double nb=1.1;**

**double q=0.7;**

**m=(int)(log(eps \* (1-q) / nb ) / log(q) + 1);**

**for(i=1;i<=n;i++) x[i]=0.0;**

**for(iteratii=1;iteratii<=m;iteratii++)**

**{**

**for(i=1;i<=n;i++)**

**{**

**double s=0.0;**

**for(j=1;j<=n;j++)**

**s+=a[i][j] \* x[j];**

**temp[i]=x[i] - s + b[i];**

**}**

**for(i=1;i<=n;i++) x[i]=temp[i];**

**}**

**for(i=1;i<=n;i++)**

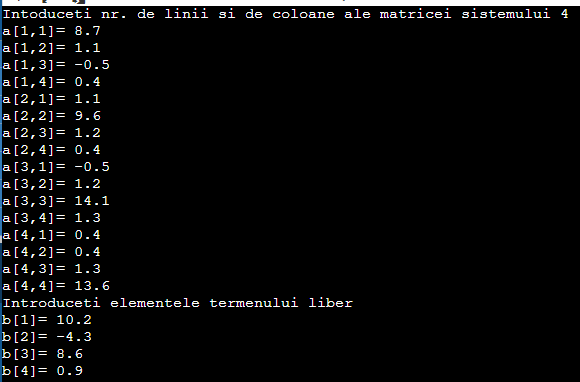
**cout<<"x["<<i<<"]= "<<x[i]<<endl;**

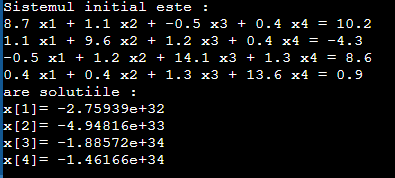
**getch();**

**return 0;**

**}**

Rezultatele obținute:





**Codul sursă:**

**Metoda iterativa a lui Gauss-Seidel**

**#include <iostream>**

**#include<stdlib.h>**

**#include<conio.h>**

**#include<math.h>**

**using namespace std;**

**double a[10][10],x[10],b[10];**

**// a este matricea sistemului**

**// x vectorul de necunoscute ale sistemului**

**// b vectorul de termeni liberi**

**int n,m,j,i,eroare;**

**void gauss\_seidel(double a[10][10], double x[10], int n)**

**{**

**int imax=20;**

**int eps=0.001;**

**int i,j,k;**

**double t,tt[10];**

**// se genereaza elementele rezultate prin impartirea fiecarui elem. de pe linie cu**

**// elem. de pe diagonala principala**

**for(i=1;i<=n;i++)**

**{**

**t=-1.0/a[i][i];**

**tt[i]=b[i]/a[i][i];**

**for(j=1;j<=n;j++)**

**a[i][j]=a[i][j] \* t;**

**}**

**k=0;**

**for(i=1;i<=n;i++) b[i]=tt[i];**

**do{**

**k++;**

**eroare=0.0;**

**for(i=1;i<=n;i++)**

**{**

**t = tt[i];**

**for(j=1;j<=n;j++) t = t + a[i][j] \* b[j];**

**b[i] = b[i] + t;**

**if(b[i] != 0.0) t = t / b[i];**

**if(fabs(t)>eroare) eroare = fabs(t);**

**}**

**}while((eroare<eps) || (k==imax) );**

**return;**

**}**

**int main()**

**{**

**cout<<"Introduceti nr. de linii si de coloane ale matricei sistemului ";**

**cin>>n;**

**for(i=1;i<=n;i++)**

**for(j=1;j<=n;j++)**

**{**

**cout<<"a["<<i<<","<<j<<"]= ";**

**cin>>a[i][j];**

**}**

**cout<<"Introduceti elementele termenului liber "<<endl;**

**for(i=1;i<=n;i++)**

**{**

**cout<<"b["<<i<<"]= ";**

**cin>>b[i];**

**}**

**cout<<"Sistemul initial este :"<<endl;**

**for(i=1;i<=n;i++)**

**{**

**cout<<a[i][1]<<" x"<<1;**

**for(j=2;j<=n;j++)**

**cout<<" + "<<a[i][j]<<" x"<<j;**

**cout<<" = "<<b[i]<<endl;**

**}**

**cout<<"are solutiile :"<<endl;**

**gauss\_seidel(a,b,n);**

**for(i=1;i<=n;i++)**

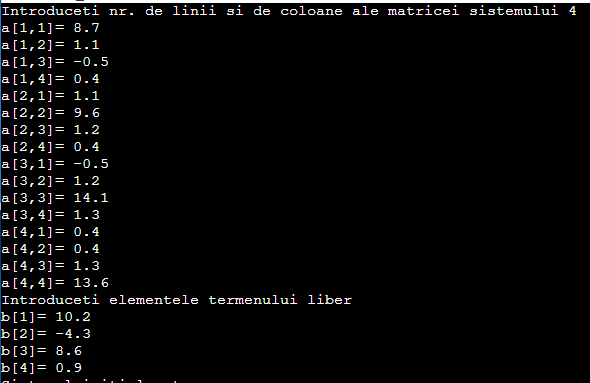
**cout<<"x["<<i<<"]= "<<b[i]<<endl;**

**cout<<"Eroarea relativa maxima este "<<eroare<<endl;**

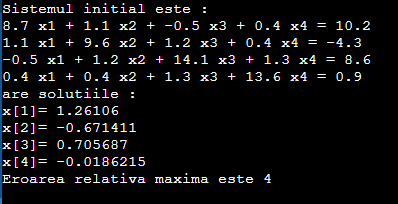
**getch();**

**}**

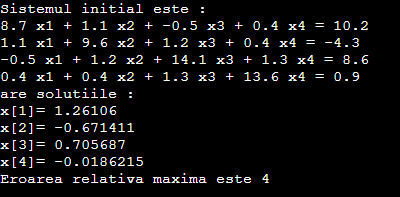
**Rezultatele obținute:**



**Eroare E=10-3**



**Eroare E=10-5**



**Concluzie:**

În cadrul laboratorului am studiat rezolvarea numerică a sistemului de ecuații liniare. Am folosit 3 metode de rezolvare: Gauss, Jacobi și Gauss – Seidel. Analizând datele obținute și iterațiile obținute de la fiecare metodă observăm că din metodele iterative Gauss

– Seidel este cea mai eficientă pentru că în cadrul algoritmului folosim rădăcina obținută în cadrul aceleiași iterații dar nu în următoarea ca în cazul metodei Jacobi.