****

Universitatea Tehnică a Moldovei

Catedra Calculatoare

Specialitatea Calculatoare

Raport

Lucrarea de laborator nr.2

Varianta 8

A îndeplinit:st.gr. C-162 Nicolenco Eugeniu

A controlat: conf. universitar Istrati Daniela

Chișinău-2017

Scopul lucrarii:

Sa se rezolve sistemul de ecuatii liniare Ax=b,utilizind:

* Metoda eliminarii Gauss;
* Metoda Cholesky;
* Metoda iterativa Jacobi cu eroarea E=10-3;

Var 3:

A= b=

Metoda Jordan-Gauss:

#include <stdlib.h>

#include <stdio.h>

#include <math.h>

#define N 4

#define M 1

void Gauss(double \*det){

double a[N][N]={{8.7, 1.1, -0.5, 0.4},

{1.1, 9.6, 1.2, 0.4},

{-0.5, 1.2, 14.1, 1.3},

{0.4, 0.4, 1.3, 13.6}};

double b[N][M]={{10.2},{-4.3},{8.6},{0.9}};

double amax,t;

int i,imax,j,k;

int n=N,m=M;

\*det=1.0;

for(k=0;k<n;k++){

amax=0.0;

for(i=k;i<n;i++){

if(amax<fabs(a[i][k])){

amax=fabs(a[i][k]);

imax=i;

}

}

if(amax==0.0){

printf("GaussJordan : Matrice singulara\n");

return;

}

if(imax!=k){

\*det=-(\*det);

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

t=a[imax][j];

a[imax][j]=a[k][j];

a[k][j]=t;

}

for(j=0;j<m;j++){

t=b[imax][j];

b[imax][j]=b[k][j];

b[k][j]=t;

}

}

\*det\*=a[k][k];

t=1.0/a[k][k];

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

for(j=0;j<m;j++)b[k][j]\*=t;

for(i=0;i<n;i++){

if(i!=k){

t=a[i][k];

for(j=0;j<n;j++)a[i][j]-=a[k][j]\*t;

for(j=0;j<m;j++)b[i][j]-=b[k][j]\*t;

}

}

}

printf("Solutia sistemului:\n x=[");

for(i=0;i<n;i++){

for(j=0;j<m;j++) printf("%f,", b[i][j]);

printf("\n");

}

printf("]\n");

}

int main(){

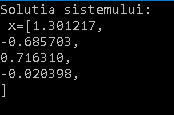
double det;

Gauss(&det);

return 0;

}

Rezultatele afisate:



Metoda Cholesky:

#include <iostream>

#include <conio.h>

#include <stdio.h>

#include <math.h>

#define N 4

#include<time.h>

using namespace std;

clock\_t start,stop;

float A[4][4]={ 8.7,1.1,-0.5,0.4,

1.1,9.6,1.2,0.4,

-0.5,1.2,14.1,1.3,

0.4,0.4,1.3,13.6 };

float B[4]={10.2,-4.3,8.6,0.9};

void cholesky()

{ int i,j,k,w=0,h=0,v=0;

float t,x[N],l[N][N],y[N];

start=clock();

l[0][0]=sqrt(A[0][0]);

for(i=1;i<N;i++)

{l[i][0]=A[i][0]/l[0][0]; }

for(k=1;k<N;k++)

{w++;t=0;

for(j=0;j<k-1;j++)

t+=l[k][j]\*l[k][j];

l[k][k]=sqrt(A[k][k]-t);

for(i=k+1;i<N;i++)

{t=0;w++;

for(j=0;j<k-1;j++)

t+=l[i][j]\*l[k][j];

l[i][k]=(A[i][k]-t)/l[k][k];}}

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

for(j=i+1;j<N;j++)

{l[i][j]=l[j][i]; } }

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

{t=0; w++;

for(j=0;j<i;j++)

t+=l[i][j]\*y[j];

t=B[i]-t;

y[i]=t/l[i][i]; }

for(i=N-1;i>=0;i--)

{t=0;w++;

for(j=N-1;j>i;j--)

t+=l[i][j]\*x[j];

t=y[i]-t;

x[i]=t/l[i][i]; }

printf("Metoda Cholesky:\n");

for(i=0;i<N;i++)printf("y[%d]=%7.3f \n ",i,x[i]);

stop=clock();

h=(stop-start)/CLK\_TCK ;

puts("\nMatricea inversata:");

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

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

printf("%5.2f ",A[i][N-1-j]);

puts(" "); }

}

int main()

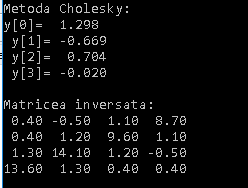
{int i;

cholesky(); printf("\n\n");

system("pause");

return 0;

}

Rezultatele afisate:

Metoda Jacobi:

#include<stdio.h>

#include<conio.h>

#include<math.h>

#include<stdlib.h>

int main()

{

float a[20][20],x[20],e,big,temp,relerror,sum;

int n,i,j,maxit,itr;

char ch;

printf("\n\nIntroduceti marimea ecuatiei(matricei) :: ");

scanf("%d",&n);

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

{

printf("\n\nIntroduceti coeficientii ecuatiei %d si rezultatul ei \n",i);

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

scanf("%f",&a[i][j]);

}

printf("\n\nIntroduceti eroarea (eps) si nr de iteratii :: \n");

scanf("%f%d",&e,&maxit);

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

x[i]=0;

for(itr=1;itr<=maxit;itr++)

{

big=0;

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

{

sum=0;

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

{

if(i!=j

)

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

}

temp=(a[i][n+1]-sum)/a[i][i];

relerror=fabs((x[i]-temp)/temp);

if(relerror>big)

big=relerror;

x[i]=temp;

}

if(big<=e)

{

printf("Converge catre solutia in %d iteratii\n",itr);

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

printf("\n%.4f\t",x[i]);

system("pause");

exit(1);

}

}

printf("ecuatua nu converge in %d iteratii \n",maxit);

getch();

return 0;

}

Rezultatele afisate:

