Федеральное агентство по образованию Российской Федерации

Тамбовский государственный технический университет

Кафедра:

**Отчет** по лабораторной работе №3

**Дисциплина:** Вычислительная математика

Выполнил:. гр.

Проверил:.

Тамбов 20

**Задание**

Решить систему нелинейных уравнение методом простых итераций и методом Ньютона:

**Описание работы программы**

*Метод простых итераций*:

Преобразуем матрицу к следующему виду :

Получим матрицу Якоби:

Определитель:

Т.е. метод простых итераций будет сходиться.

*Метод Ньютона*:

Матрица Якоби для метода Ньютона:

Отсюда:

*Код программы*

#include <stdio.h>

#include <conio.h>

#include <stdlib.h>

#include <math.h>

void vivod(int i,float \*X,float \*Y)

{

int j;

if(i>8)

{

for(j=0;j<4;j++)

printf("\n Step %d x=%f y=%f",j+1,X[j],Y[j]);

printf("\n...");

for(j=4;j>0;j--)

printf("\n Step %d x=%f y=%f",i-j+1,X[8-j],Y[8-j]);

}

else

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

printf("\n Step %d x=%f y=%f",j+1,X[j],Y[j]);

}

int main()

{

float X[8],Y[8];

int j,i=0;

float x,y0,y,x0,e=0;

printf("\tMethod of simple iterations\n");

printf("Enter x0=");

scanf("%f",&x0);

printf("Enter y0=");

scanf("%f",&y0);

printf("Enter accuracy=");

scanf("%f",&e);

for(j=0;j<8;j++)

{

X[j]=0;

Y[j]=0;

}

x=x0;

y=y0;

do

{

i++;

x0=x;

y0=y;

x=1-cos(y0)/2;

y=sin(x0+1)-1.2;

if(i>8)

{

X[4]=X[5]; X[5]=X[6]; X[6]=X[7]; X[7]=x;

Y[4]=Y[5]; Y[5]=Y[6]; Y[6]=Y[7]; Y[7]=y;

}

else

{

X[i-1]=x;

Y[i-1]=y;

}

}

while(sqrt((x-x0)\*(x-x0)+(y-y0)\*(y-y0))>e);

vivod(i,X,Y);

printf("\n\tNewton's method\n");

printf("Enter x0=");

scanf("%f",&x0);

printf("Enter y0=");

scanf("%f",&y0);

printf("Enter accuracy=");

scanf("%f",&e);

for(j=0;j<8;j++)

{

X[j]=0;

Y[j]=0;

}

i=0;

x=x0;

y=y0;

do

{

i++;

x0=x;

y0=y;

x=x0-(-sin(y0)\*(sin(x0+1)-y0-1.2)+(2\*x0+cos(y0)-2))/(2-sin(y0)\*cos(x0+1));

y=y0-(-2\*(sin(x0+1)-y0-1.2)+cos(x0+1)\*(cos(y0)+2\*x0-2))/(2-sin(y0)\*cos(x0+1));

if(i>8)

{

X[4]=X[5]; X[5]=X[6]; X[6]=X[7]; X[7]=x;

Y[4]=Y[5]; Y[5]=Y[6]; Y[6]=Y[7]; Y[7]=y;

}

else

{

X[i-1]=x;

Y[i-1]=y;

}

}

while(sqrt((x-x0)\*(x-x0)+(y-y0)\*(y-y0))>e);

vivod(i,X,Y);

getch();

return 0;

}

Результат:

