Звіт з предмету Чисельні методи
Пустовіта Дмитра
КА-32

1. Умова Задачі (Варіант 16)

Розв'язати систему нелінійних рівнянь

$\begin{cases} \cos(x+\alpha) + by = c \\ x + \sin(y+\beta) = d \end{cases}$ методом простої ітерації, та розв'язати систему

$\begin{cases} \sin(x+y) + cx = d \\ x^2 + y^2 = 1 \end{cases}$ методом Ньютона. При цьому вважати, що

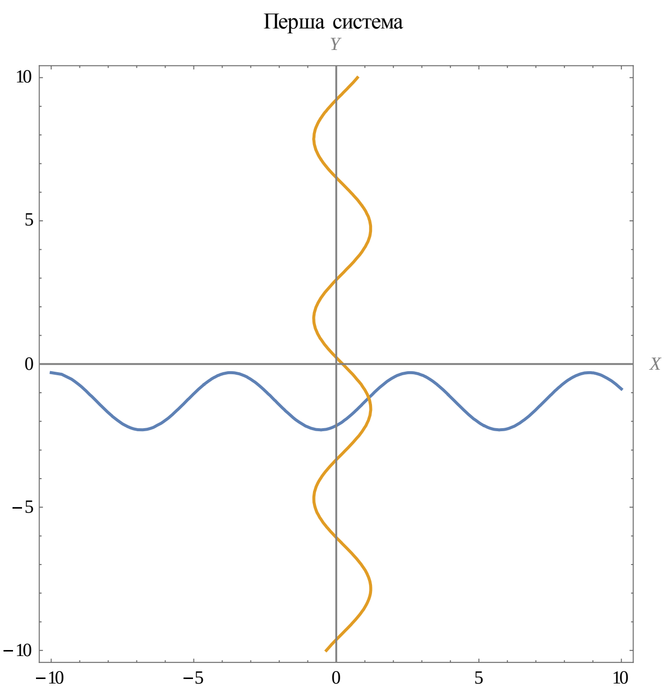
$\alpha = 0{,}553; \quad \beta = -0{,}015; \quad a = 0.923; \quad b = 1.001; \quad c = -1.3; \quad d = 0.209;$

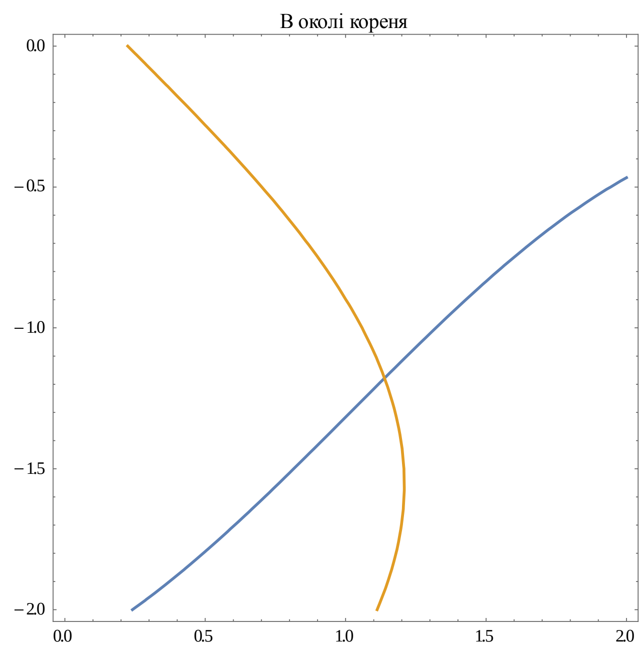
2. Математичне розв'язання задачі

Для використання методу простої ітерації спершу потрібно рівняння F(X)=0 звести до вигляду Ф(X)=X. тобто отримаємо:

$$\begin{cases} \cos(x+\alpha) + by = c \\ x + \sin(y+\beta) = d \end{cases} \sim \begin{cases} y = \frac{c}{b} - \cos(x+\alpha) \\ x = d - \sin(y+\beta) \end{cases}$$

Для того, аби знайти початкове наближення побудуємо схематичні графіки правих частин:





Звідси можна взяти початкове наближення:

$$X = \begin{pmatrix} x \\ y \end{pmatrix} \approx \begin{pmatrix} 1 \\ -1 \end{pmatrix}$$

Тепер побудуємо ітераційний процес:

$$\begin{cases} X_0 = \begin{pmatrix} 1 \\ -1 \end{pmatrix} \\ X_{k+1} = \Phi(X_{k+1}) \end{cases}$$

Критерієм завершення ітераційного процесу будуть умови:

$$\|X_{k+1} - X_k\| < eps$$
$$\|F(X_{k+1})\| < eps$$

Метод збігатиметься до розв’язку, якщо Ф – є стиснутим зображенням, тобто:

$$\|\Phi(X) - \Phi(Y)\| < L\|x - y\|, \qquad \text{де } L < 1$$

Для розв’язання системи $\begin{cases} \sin(x+y) + cx = d \\ x^2 + y^2 = 1 \end{cases}$ методом Ньютона

побудуємо матрицю $F'(X) = \begin{pmatrix} \frac{\partial f_1}{\partial x} & \frac{\partial f_1}{\partial y} \\ \frac{\partial f_2}{\partial x} & \frac{\partial f_2}{\partial y} \end{pmatrix}$

Маємо:

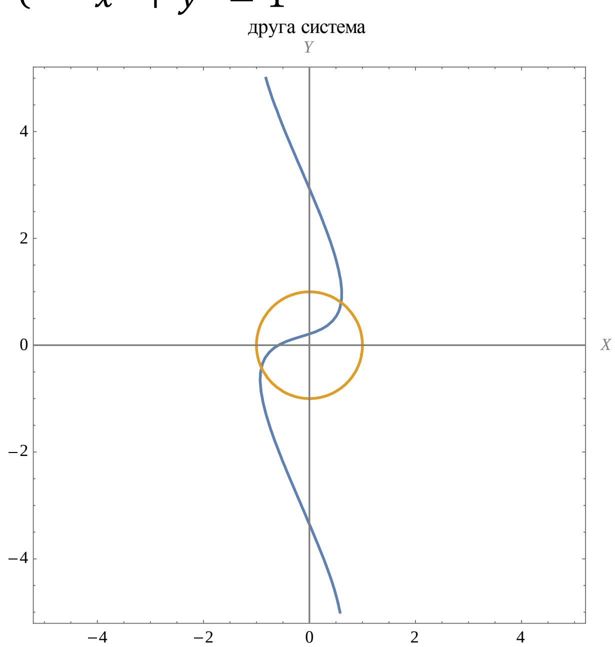
$$F'(X) = \begin{pmatrix} \cos(x+y) + c & \cos(x+y) \\ 2x & 2y \end{pmatrix}$$

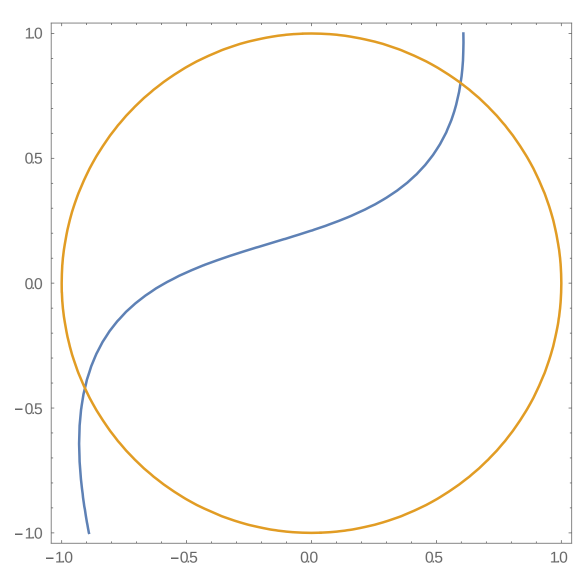
З формули Тейлора отримаємо таке співвідношення:

$$X_{k+1} = X_k - (F'(X_k))^{-1} \times F(X_k)$$

Аби знайти початкове наближення зобразимо схематично графіки функцій

$$\begin{cases} \sin(x+y) + cx = d \\ x^2 + y^2 = 1 \end{cases}$$





Таким чином отримали перше наближення для точок X_1 та X_2:

$$\begin{cases} X_1 = \begin{pmatrix} 0.5 \\ 0.7 \end{pmatrix} \\ X_2 = \begin{pmatrix} -0.8 \\ -0.3 \end{pmatrix} \end{cases}$$

Критерієм завершення ітераційного процесу будуть умови:

$$\|X_{k+1} - X_k\| < eps$$
$$\|F(X_{k+1})\| < eps$$

3. **Результати роботи** (метод простої ітерації)

Початкове наближення:
1
-1

Iteration #1

```
Solution:
1.05848
-1.3165
||X[k+1]-X[k]||= 0.321854
||F(x)= ||0.135316
```

Iteration #2

```
Solution:
1.1805
-1.25803
||X[k+1]-X[k]||= 0.135308
||F(x)= ||0.122265
```

Iteration #3

```
Solution:
1.16499
-1.13671
||X[k+1]-X[k]||= 0.122305
||F(x)= ||0.0452641
.
.
.
```

Iteration #10

```
Solution:
1.13934
-1.17954
||X[k+1]-X[k]||= 0.00224946
||F(x)= ||0.00199032
```

Iteration #11

```
Solution:
1.13905
-1.17746
||X[k+1]-X[k]||= 0.00210825
||F(x)= ||0.000872613
```

Iteration #12

```
Solution:
1.13828
-1.17775
||X[k+1]-X[k]||= 0.000822325
||F(x)= ||0.000891001
.
.
.
```

Iteration #18

```
Solution:
1.13858
-1.17825
||X[k+1]-X[k]||= 4.03218e-005
||F(x)= ||8.31817e-005
```

Iteration #19

```
Solution:
1.13857
-1.17821
||X[k+1]-X[k]||= 3.78034e-005
||F(x)= ||0.000126467
```

Iteration #20

```
Solution:
1.13856
-1.17822
||X[k+1]-X[k]||= 1.47565e-005
||F(x)= ||0.000134197
```

Iteration #21

```
Solution:
1.13856
-1.17823
||X[k+1]-X[k]||= 1.3835e-005
||F(x)= ||0.000118664
```

Iteration #22

```
Solution:
1.13857
-1.17823
||X[k+1]-X[k]||= 5.40058e-006
||F(x)= ||0.000115459
```

(Метод Нютона)

```
Початкове наближення:
0.5
0.7
```

Iteration #1

```
Solution:
0.61729
0.801936
||X[k+1]-X[k]||= 0.155396
||F(x)||= 0.270064
```

Iteration #2

```
Solution:
0.597361
0.80222
||X[k+1]-X[k]||= 0.0199315
||F(x)||= 0.0333085
```

Iteration #3

```
Solution:
0.597176
0.802111
||X[k+1]-X[k]||= 0.000215262
||F(x)||= 0.000440598
```

Iteration #4

```
Solution:
0.597175
0.802111
||X[k+1]-X[k]||= 3.80315e-008
||F(x)||= 6.31116e-008
```

```
Початкове наближення:
-0.8
-0.3
Iteration #1
Solution:
-0.928563
-0.407164
||X[k+1]-X[k]||= 0.16737
||F(x)||= 0.276631

Iteration #2
Solution:
-0.907512
-0.420774
||X[k+1]-X[k]||= 0.0250683
||F(x)||= 0.0379713

Iteration #3
Solution:
-0.907381
-0.42031
||X[k+1]-X[k]||= 0.000482856
||F(x)||= 0.000628993

Iteration #4
Solution:
-0.907381
-0.42031
||X[k+1]-X[k]||= 1.59126e-007
||F(x)||= 2.89809e-007
```

4.Код програми

```
// NonLineSys.cpp : Defines the entry point for the console application.
//

#include "stdafx.h"
#include "fstream"
#include "math.h"

#define alfa 0.553
#define beta -0.015
#define a 0.923
#define b 1.001
#define c -1.3
#define d 0.209
#define eps 0.00001
#define n 2

typedef double vec[n];
typedef double Mat[n][n];
using namespace std;

void system1(vec x, vec* X){
        (*X)[1] = c / b - cos(x[0] + alfa);
        (*X)[0] = d - sin(x[1] + beta);
}

double normVec(vec* x){
        double s = 0;
        for (int i = 0; i < n; i++)
                s = s + pow((*x)[i], 2);
        return pow(s, 0.5);
}

double newazka1(vec* x){
        vec X;
        X[0] = b*(*x)[1] +cos((*x)[0] + alfa)-c;
        X[1] = (*x)[0]+sin((*x)[1] + beta)-d;
        return normVec(&X);
}

double normvecs(vec* x,vec* y){
        double s = 0;
        for (int i = 0; i < n; i++)
                s = s + pow((*x)[i] - (*y)[i], 2);
        return pow(s, 0.5);
}

void outVec(ofstream &stm, vec* x){
        int i;
        for (i = 0; i < n; i++)
                stm << (*x)[i] << endl;
}

void simpleIter(vec* x0){
        ofstream fout("simpIter.txt");
        int i,k = 0;
        vec x;
        double p,q;
        fout << "Початкове наближення:" << endl;
        outVec(fout, x0);
        do {
                k++;
                system1((*x0), &x);
                p = fabs(normvecs(&x, x0));
                q = newazka1(&x);
                for (i = 0; i < n; i++) (*x0)[i] = x[i];
                fout << "Iteration #" << k << endl << "Solution:"<<endl;
                outVec(fout, &x);
                fout << "||X[k+1]-X[k]||= " << p << endl;
                fout << "||F(x)= ||" << q <<endl;
        } while ((q>eps)&&(p > eps));
        fout.close();
}


double system2(vec x, vec* X){
        (*X)[0] = sin(x[0] + x[1]) + c*x[0] - d;
        (*X)[1] = pow(x[1], 2) + pow(x[0], 2) - 1;
        return normVec(X);
}

void dsystem2(vec x, Mat* A){
        (*A)[0][0] = cos(x[0] + x[1]) + c;
        (*A)[0][1] = cos(x[0] + x[1]);
        (*A)[1][0] = 2*x[0];
        (*A)[1][1] = 2*x[1];
}

void obern2x2(Mat* A, Mat* B){
```

```
	double L = (*A)[0][0] * (*A)[1][1] -
(*A)[0][1] * (*A)[1][0];
	(*B)[0][0] = (*A)[1][1]/L;
	(*B)[0][1] = -(*A)[0][1]/L;
	(*B)[1][0] = -(*A)[1][0]/L;
	(*B)[1][1] = (*A)[0][0]/L;
}

void product(Mat*  A, vec* x, vec* Ax){
	int i, j;
	double S;
	for (i = 0; i < n; i++){
		S = 0;
		for (j = 0; j < n; j++)
			S = S + (*A)[i][j] * (*x)[j];
		(*Ax)[i] = S;
	}
}

void Newton(ofstream &fout,vec* x0){
	int i,k = 0;
	vec x,Fx,FFx;
	Mat A,B;
	double p,q;
	fout << "Початкове наближення:" << endl;
	outVec(fout, x0);
	do{
		k++;
		q=system2((*x0), &Fx);
		dsystem2((*x0), &A);
		obern2x2(&A, &B);
		product(&B,&Fx,&FFx);
		for (i = 0; i < n; i++) x[i] =
(*x0)[i] - FFx[i];
	    p = fabs(normvecs(&x, x0));
		for (i = 0; i < n; i++) (*x0)[i] =
x[i];
		fout << "Iteration #" << k << endl <<
"Solution:" << endl;
		outVec(fout, &x);
		fout << "||X[k+1]-X[k]||= " << p <<
endl;
		fout << "||F(x)||= " << q << endl <<
endl;
	} while ((q>eps) && (p > eps));
}

int _tmain(int argc, _TCHAR* argv[])
{
	vec u, v, t;
	int i;
	ifstream fin("in.txt");
	ofstream fout("Newton.txt");
	for (i = 0; i < n; i++) fin >> u[i];
	for (i = 0; i < n; i++) fin >> v[i];
	for (i = 0; i < n; i++) fin >> t[i];
	simpleIter(&u);
	Newton(fout,&v);
	Newton(fout,&t);
	fin.close();
	fout.close();
	return 0;
}
```

5. Висновок

Виконавши нескладні перетворення було розв’язано першу систему методом простої ітерації. Другу систему було розв’язано методом Ньютона, таким чином як видно з малюнків було знайдено всі корені заданих систем.