Міністерство освіти і науки України
Національний університет “Львівська політехніка”
Інститут комп’ютерних наук та інформаційних технологій
Кафедра програмного забезпечення

Звіт

Про виконання лабораторної роботи №7

На тему:

«Чисельні методи розв’язування систем нелінійних рівнянь»
з дисципліни «Чисельні методи»

Лектор:
доцент каф. ПЗ
Мельник Н. Б.
Виконав:
ст. гр. ПЗ-11
Морозов О. Р.
Прийняла:
доцент каф. ПЗ
Мельник Н. Б.

« __ » ________ 2022 р.

∑ = _____ _________ .

Львів – 2022

Тема:.чисельні методи розв’язування систем нелінійних рівнянь.
Мета: ознайомлення на практиці з методом ітерацій та методом Ньютона розв’язування систем нелінійних рівнянь.

Теоретичні відомості

Метод простої ітерації - суть методу полягає у перетворенні системи з двох нелінійних рівнянь до вигляду:

$$\begin{cases} x = \varphi_1(x, y), \\ y = \varphi_2(x, y), \end{cases}$$

Після цього ітераційний процес зводиться до такого вигляду:

$$\begin{cases} x_{n+1} = \varphi_1(x_n, y_n), \\ y_{n+1} = \varphi_2(x_n, y_n), \end{cases}$$

Для збіжності ітераційного процесу мають виконуватися такі умови:

$$\left|\frac{\partial \varphi_1}{\partial x}\right| + \left|\frac{\partial \varphi_2}{\partial x}\right| \le q_1 < 1 \ , \qquad \left|\frac{\partial \varphi_1}{\partial y}\right| + \left|\frac{\partial \varphi_2}{\partial y}\right| \le q_2 < 1 .$$

1) функції 1(x, y) та 2(x, y) визначені та неперервно-диференційовані в області D;
2) початкове наближення і всі наступні наближення належать області D;
3) в області D виконуються нерівності:
Ітераційний процес припиняється якщо |xn+1 – xn| + |yn+1 - yn| < eps.

Метод простої ітерації, який застосовують для знаходження розв'язку одного нелінійного рівняння або системи двох нелінійних рівнянь, має перший порядок збіжності (лінійну збіжність).

Метод Ньютона – суть методу полягає у перетворенні системи нелінійних рівнянь до вигляду:

$$\begin{cases} f_1(x_0, y_0) + \dfrac{\partial f_1}{\partial x}\Delta_x + \dfrac{\partial f_1}{\partial y}\Delta_y = 0, \\ f_2(x_0, y_0) + \dfrac{\partial f_2}{\partial x}\Delta_x + \dfrac{\partial f_2}{\partial y}\Delta_y = 0. \end{cases}$$

Після цього ми записуємо якобіан, складений складеної з частинних похідних функцій f1 і f2 в деякій точці:

$$\Delta(x_0, y_0) = \begin{vmatrix} \frac{\partial f_1(x_0, y_0)}{\partial x} & \frac{\partial f_1(x_0, y_0)}{\partial y} \\ \frac{\partial f_2(x_0, y_0)}{\partial x} & \frac{\partial f_2(x_0, y_0)}{\partial y} \end{vmatrix} \neq 0,$$

а поправки x і y визначимо за правилом Крамера із системи:

$$\Delta_x = -\frac{1}{\Delta(x_0, y_0)} \begin{vmatrix} f_1(x_0, y_0) & \frac{\partial f_1(x_0, y_0)}{\partial y} \\ f_2(x_0, y_0) & \frac{\partial f_2(x_0, y_0)}{\partial y} \end{vmatrix}, \qquad \Delta_y = -\frac{1}{\Delta(x_0, y_0)} \begin{vmatrix} \frac{\partial f_1(x_0, y_0)}{\partial x} & f_1(x_0, y_0) \\ \frac{\partial f_2(x_0, y_0)}{\partial x} & f_2(x_0, y_0) \end{vmatrix}.$$

Наступне наближення розв'язку системи отримаємо у вигляді:

$$\begin{cases} x_{n+1} = x_n + \Delta_x, \\ y_{n+1} = y_n + \Delta_y, \end{cases}$$

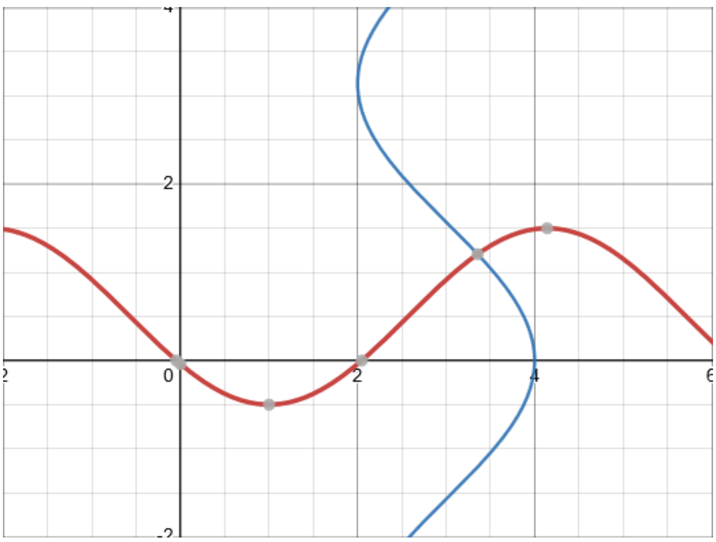
Метод Ньютона – має другий порядок збіжності (квадратичну збіжність).

Індивідуальне завдання

Розв'язати систему нелінійних рівнянь з точністю ε = 10^{-3} методом ітерацій та методом Ньютона.
Варіант 2
cos(x - 1) + y = 0.5
x - cos(y) = 3

Хід роботи

Графік функції:



Код програми:

```
#include <iostream>
#include <iomanip>
#include <cmath>
#define eps 0.001

//return y from 1 equation
double f1_x(double x) {
	return (0.5 - cos(x - 1.0)); // y = 0.5 - cos(x - 1)
}

//return solve from 1 equation
double f1_xy(double x, double y) {
	return (cos(x - 1.0) + y - 0.5); // f(x, y) = cos(x - 1) + y - 0.5
}

double der_f1_x(double x) {
	return (-sin(x - 1.0));
}

double der_f1_y() {
	return 1.0;
}

//return x from 2 equation
double f2_y(double y) {
	return (3.0 + cos(y)); // x = 3 + cos(y)
}

//return solve from 2 equation
double f2_xy(double x, double y) {
	return (x - cos(y) - 3.0); // f(x, y) = x - cos(y) - 3
}

double der_f2_x() {
	return 1.0;
}

double der_f2_y(double y) {
	return sin(y);
}

using namespace std;

double det(double a[2][2]) {
	return (a[0][0] * a[1][1]) - (a[0][1] * a[1][0]);
}

bool isConvergence(double x, double y) {
	return (der_f1_x(x) + der_f2_x() < 1 && der_f1_y() + der_f2_y(y) < 1);
}
```

```
void IterationMethod() {
	cout << "=====[ Simple Iteration Method ]=====" << endl;

	double x = 1, y = 1;
	double prev_x, prev_y;
	int iter = 0;

	if(isConvergence(x, y)) {
		cout << "Error, equation not covergence" << endl;
		return;
	}

	cout << "|Iter|       X       |       Y      |" << endl;
	do {

		++iter;
		prev_x = x;
		prev_y = y;

		y = f1_x(prev_x);
		x = f2_y(prev_y);

		cout << "|" << setw(4) << iter << "|" << setw(15) << x << "|" << 
setw(14) << y << "|" << endl;
	} while((fabs(x - prev_x) + fabs(y - prev_y)) > eps);

	cout << "====================================" << endl << "Result:\n[X] = 
" << x << "\n[Y] = " << y << endl;
}

double Jacobi(double x, double y) {
	double a[2][2];

	a[0][0] = der_f1_x(x);
	a[0][1] = der_f1_y();
	a[1][0] = der_f2_x();
	a[1][1] = der_f2_y(y);

	return det(a);
}

double Dx(double x, double y) {
	double a[2][2];

	a[0][0] = f1_xy(x, y);
	a[0][1] = der_f1_y();
	a[1][0] = f2_xy(x, y);
	a[1][1] = der_f2_y(y);

	return det(a);
```

```
}

double Dy(double x, double y) {
	double a[2][2];

	a[0][0] = der_f1_x(x);
	a[0][1] = f1_xy(x, y);
	a[1][0] = der_f2_x();
	a[1][1] = f2_xy(x, y);

	return det(a);
}

void NewtonMethod() {
	double x0 = 2, y0 = 3;
	if(Jacobi(x0, y0) == 0) {
		cout << "Error: Jacobi determinant = 0" << endl;
		return;
	} else {
		cout << endl << "==========[ Newton Method ]==========" << endl;
		cout << "|Iter|       X       |       Y      |" << endl;

		double x1 = x0 - (Dx(x0, y0) / Jacobi(x0, y0));
		double y1 = y0 - (Dy(x0, y0) / Jacobi(x0, y0));
		int iter = 1;

		cout << "|" << setw(4) << iter << "|" << setw(15) << x1 << "|" << 
setw(14) << y1 << "|" << endl;

		do {
			++iter;
			x0 = x1;
			y0 = y1;
			x1 = x0 - (Dx(x0, y0) / Jacobi(x0, y0));
			y1 = y0 - (Dy(x0, y0) / Jacobi(x0, y0));

			cout << "|" << setw(4) << iter << "|" << setw(15) << x1 << "|" 
<< setw(14) << y1 << "|" << endl;

		} while((fabs(x1 - x0) + fabs(y1 - y0)) > eps);

		cout << "=====================================" << endl << 
"Result:\n[X] = " << x1 << "\n[Y] = " << y1 << endl;
	}
}

int main() {
	IterationMethod();
	NewtonMethod();
	return 0;
}
```

Результат:

Microsoft Visual Studio Debug...

```
=====[ Simple Iteration Method ]=====
|Iter|       X       |       Y       |
|   1|        3.5403|          -0.5|
|   2|       3.87758|       1.32461|
|   3|       3.24371|       1.46535|
|   4|       3.10525|       1.12327|
|   5|       3.43274|       1.00937|
|   6|       3.53239|       1.25911|
|   7|       3.30667|       1.32011|
|   8|       3.24807|       1.17123|
|   9|       3.38902|       1.12667|
|  10|       3.42967|       1.22993|
|  11|        3.3343|       1.25711|
|  12|       3.30857|       1.19146|
|  13|       3.37031|       1.17264|
|  14|       3.38772|       1.21701|
|  15|       3.34645|       1.22904|
|  16|       3.33514|       1.20018|
|  17|       3.36219|       1.19206|
|  18|       3.36974|       1.21133|
|  19|       3.35177|       1.21662|
|  20|       3.34682|       1.20397|
|  21|       3.35865|       1.20044|
|  22|       3.36194|       1.20884|
|  23|        3.3541|       1.21116|
|  24|       3.35193|       1.20563|
|  25|       3.35711|       1.20409|
|  26|       3.35855|       1.20775|
|  27|       3.35512|       1.20877|
|  28|       3.35417|       1.20635|
|  29|       3.35643|       1.20568|
|  30|       3.35706|       1.20728|
|  31|       3.35557|       1.20772|
|  32|       3.35515|       1.20666|
|  33|       3.35614|       1.20637|
|  34|       3.35641|       1.20707|
=====================================
Result:
[X] = 3.35641
[Y] = 1.20707

==========[ Newton Method ]==========
|Iter|       X       |       Y       |
|   1|       2.39245|      0.289935|
|   2|       3.60716|       1.51804|
|   3|        3.3439|       1.22647|
|   4|       3.35592|       1.20697|
|   5|       3.35591|       1.20691|
=====================================
Result:
[X] = 3.35591
[Y] = 1.20691

D:\University\2_semester\Numerical Methods\Lab 7\x64\Debug\
```

Висновок

Виконуючи лабораторну роботу №6, я навчився програмувати розв’язки систем нелінійних рівнянь методами простої ітерації та Ньютона.