МІНІСТЕРСТВО ОСВІТИ І НАУКИ УКРАЇНИ

НАЦІОНАЛЬНИЙ УНІВЕРСИТЕТ «ЛЬВІВСЬКА ПОЛІТЕХНІКА»

******

Лабораторна робота №3

З дисципліни

“Чисельні методи ”

*Виконав:*

*студент групи ІР-25*

*Трощук Тарас*

*Прийняла/в:*

*Дзелендзяк У.Ю.*

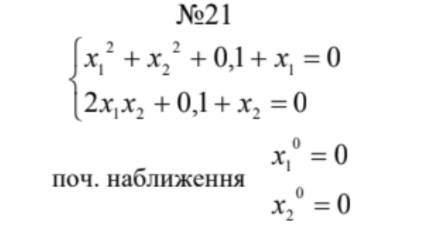
*Львів - 2024*

**Лабораторна робота № 3**

Варіант 21

**Завдання:**

Розв’язати систему нелінійних рівнянь методом січних (оберт.матриці методом Гауса з вибор. гол. ел-тів по стовпцю )



**Розв’язок:**

#include <iostream>

#include <cmath>

using namespace std;

const double e = 1e-6;

const double h = 1e-5;

const int n = 2;

void computeFunctions(const double x[], double f[]) {

f[0] = x[0] \* x[0] + x[1] \* x[1] + 0.1 + x[0];

f[1] = 2 \* x[0] \* x[1] + 0.1 + x[1];

}

void computeJacobian(const double x[], double J[n][n]) {

double x\_h[n], f1[n], f2[n];

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

x\_h[i] = x[i];

x\_h[0] += h;

computeFunctions(x\_h, f1);

x\_h[0] -= 2 \* h;

computeFunctions(x\_h, f2);

J[0][0] = (f1[0] - f2[0]) / (2 \* h);

J[1][0] = (f1[1] - f2[1]) / (2 \* h);

x\_h[0] = x[0];

x\_h[1] += h;

computeFunctions(x\_h, f1);

x\_h[1] -= 2 \* h;

computeFunctions(x\_h, f2);

J[0][1] = (f1[0] - f2[0]) / (2 \* h);

J[1][1] = (f1[1] - f2[1]) / (2 \* h);

}

void gaussElimination(double J[n][n], double f[], double delta[]) {

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

int max\_row = i;

for (int k = i + 1; k < n; ++k) {

if (fabs(J[k][i]) > fabs(J[max\_row][i]))

max\_row = k;

}

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

swap(J[max\_row][k], J[i][k]);

swap(f[max\_row], f[i]);

for (int k = i + 1; k < n; ++k) {

double factor = J[k][i] / J[i][i];

for (int j = i; j < n; ++j) {

J[k][j] -= factor \* J[i][j];

}

f[k] -= factor \* f[i];

}

}

for (int i = n - 1; i >= 0; --i) {

delta[i] = f[i];

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

delta[i] -= J[i][j] \* delta[j];

delta[i] /= J[i][i];

}

}

int main() {

double x[n] = {0.0, 0.0};

double f[n];

double J[n][n];

double delta[n];

int iter = 0;

while (true) {

computeFunctions(x, f);

computeJacobian(x, J);

gaussElimination(J, f, delta);

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

x[i] -= delta[i];

}

double norm = 0.0;

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

norm += delta[i] \* delta[i];

}

norm = sqrt(norm);

if (norm < e) break;

iter++;

if (iter > 1000) {

cout << "Not found" << endl;

return 1;

}

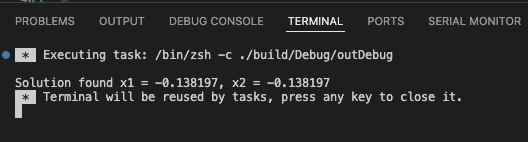
}

cout << "Solution found x1 = " << x[0] << ", x2 = " << x[1] << endl;

return 0;

}

**Результат:**

****

**Висновок:**

Під час виконання даної лабораторної . Було імплементовано метод для локалізації коренів системи нелінійних рівнянь мовою програмування С++.