Ministerul Educaţiei al Republicii Moldova

Universitatea Tehnică a Moldovei

**RAPORT**

Lucrarea de laborator nr.1

Disciplina: *Metode numerice*

Varianta 19

A efectuat:

st. gr. TI-15X Vlașițchi Ștefan

A verificat:

lect., sup. Daniela Istrati

**Tema:** Rezolvarea numerica a ecuatiilor algebrice si transcendenta.Separarea radacinilor.

**Scopul lucrarii** :

1) Sa se separe toate radacinile reale ale ecuatiei f(x)=0 unde y=f(x) este o functie reala de variabila reala.

2) Să se determine o radacina reala a ecuatiei date cu ajutorul metodei injumatatirii intervalului cu o eroare mai mica decit ε=10-2 .

3) Sa se precizeze radacina obtinuta cu exactitatea ε= 10-6 ,utilizind:

* metoda aproximatiilor succesive ;
* metoda tangentelor (Newton);
* metoda secantelor .

4) Sa se compare rezultatele luind in consideratie numarul de iteratii , evaluarile pentru functii si derivata.

**Ecuatiile propuse spre rezolvare:**

**Mersul lucrarii**

Pentru separarea radacinilor pentru prima ecuatie am folosit metoda grafica, metoda analitica si sirul lui Rolle de separare a radacinilor:

**Sirul lui Rolle**

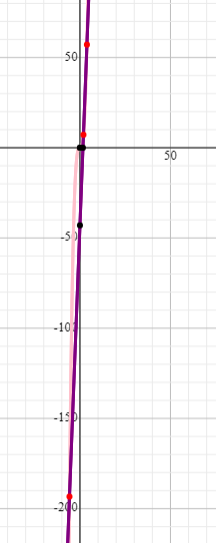
f(x)= x3-26x+43

f’(x)=3x2-26

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| x | -2 | -1 | 0 | 1 | 2 |
| f(x) | -14 | -23 | -26 | -23 | -14 |

**Metoda grafica**

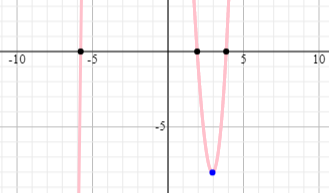
Graficul pentru si



**Fig.1** Graficul functiei y=x3 si y=x2+15x-19

Din grafic observam ca intersectiile au loc in intervalele (1.5;2),(3.5;4),(-6;-5.5)

**Metoda analitica**



**Fig.2** Graficul functiei f(x)=x3-26x+43

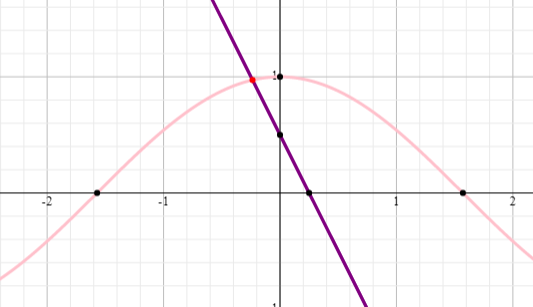
Din grafic observam ca intersectiile au loc in intervalele (-6;-5),(1;2),(3,4)

Deci avem 3 radacini , ,

Pentru cea de-a 2-a ecuatie am folosit la fel metoda grafica si metoda analitica de separare a radacinilor:

Graficul pentru si

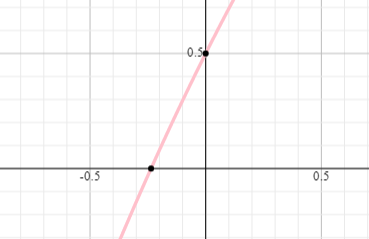
**Metoda grafica**



**Fig.3** Graficul functiei y=cos(x) si y=0.5-2x

Prin urmare ecuatia data are o radacina reala **ξ** ∈ (-0.5,0)

**Metoda analitica**



**Fig.4** Graficul functiei f(x)=cos(x)+2x-0.5

Din grafic observam ca intersectia are loc in intervalul (0;1)

Deci avem 1 radacina

**Calculul aproximativ al radacinilor**

**Metoda injumatatirii intervalului**

Consideram functia , unde functia este continua pe intervalul [a,b], are o singura radacina reala si .

Se calculeaza , daca c=0, atunci c este radacina cautata, in caz contrar :

daca

daca

Iteratiile se repeta pina cind este respectata conditia (b-a)<ε unde ε este eroarea

**Metoda aproximatiilor succesive**

Pentru a folosi Metoda aproximatiilor succesive, e necesar de pus ecuatia sub forma . Pentru ca metoda sa convearga catre radacina e necesar de a se respecta conditia de convergenta :

Functia e derivabila si derivata sa satisface inegalitatea , oricare ar fi .

**Metoda lui Newton(tangentelor)**

Presupunem ecuatia , care admite o singura solutie reala pe intervalul [a,b], iar derivatele si pastreaza un semn constant pe intervalul [a,b].

Formula pentru metoda tangentelor este :

Pentru a asigura convergenta procesului iterational e necesar ca din valorile *a* si *b* sa fie aleasa in calitate de solutie initiala acea valoare pentru care are loc inegalitatea

**Metoda secantelor**

Metoda secantelor poate fi dedusa din metoda lui Newton. Formula iteativa pentru calcularea aproximativa a solutiilor prin metoda secantelor este :

La fel ca pentru Metoda lui Newton in calitate de solutie initiala vom alege acea valoare pentru care are loc inegalitatea :

Finisarea procesului iterational are loc atunci cind aleasa in calitate de solutie initiala acea valoare pentru care are loc inegalitatea , unde este eroarea.

**Codul programului**

#include <iostream>

#include <conio.h>

#include <math.h>

using namespace std;

double(\*f)(double), (\*fn)(double), (\*fd)(double);

double f1(double x) {

    return pow(x, 3) - 26 \* x + 43;

}

double f2(double x) {

    return cos(x) + 2 \* x - 0.5;

}

double fd1(double x) {

    return 3 \* pow(x, 2)  - 26;

}

double fd2(double x) {

    return  2 - sin(x);

}

double f3(double x) {

    return (pow(x, 3) + 43) / 26;

}

double f4(double x) {

    return (cos(x) - 0.5) / (-2);

}

void aproximatie() {

    int k = 0;

    double x0, x1, eps = 0.000001;

    cout << " Introduceti valoare initiala x0" << endl;

    cout << " x0 = ";

    cin >> x0;

    while (1) {

        x1 = fn(x0);

        k++;

        if (abs(x1 - x0)<eps) {

            cout << " Radacina este: " << x0 << endl << " Numarul de iteratii " << k << endl;

            break;

        }

        x0 = x1;

    }

    \_getch();

}

void injumatatire() {

    int k = 0;

    double a, b, c = 0, eps=0.01;

    cout << " Introduceti intervalul " << endl;

    cout << " a = ";

    cin >> a;

    cout << " b = ";

    cin >> b;

    while ((b - a) > eps) {

        k++;

        c = (a + b) / 2;

        if (f(c) == 0)

            break;

        if (f(a)\*f(c) < 0)

            b = c;

        else

            a = c;

    }

    cout << " Radacina este: " << c << endl;

    cout << " Numarul de iteratii: " << k;

    \_getch();

}

void newton() {

    int k = 0;

    double x0, x1, eps = 0.000001;

    cout << " Introduceti valoare initiala x0" << endl;

    cout << " x0 = ";

    cin >> x0;

    while (1) {

        x1 = x0 - f(x0) / fd(x0); k++;

        if (abs(x1 - x0) < eps) {

            cout << " Radacina este: " << x0 << endl << " Numarul de iteratii " << k << endl; break;

        }

        x0 = x1;

    } \_getch();

}

void secante() {

    double x2, x1, x3 = 0, y, eps = 0.000001;

    int n = 0;

    cout << " Introduceti intervalul " << endl;

    cout << " a = ";

    cin >> x1;

    cout << " b = ";

    cin >> x2;

    do {

        n++;

        y = x3;

        x3 = x2 - (f(x2)\*(x2 - x1) / (f(x2) - f(x1)));

        x1 = x2;

        x2 = x3;

    } while (fabs(y - x3) >= eps);

    cout << " Radacina este: " << x3 << endl;

    cout << " Numarul de iteratii : " << n << endl;

    \_getch();

}

void selectFunction() {

    system("cls");

    cout << " 1. Functia f1(x) = x^3 - 26x + 43" << endl;

    cout << " 2. Functia f2(x) = cos(x) + 2x - 0.5" << endl;

    int opt;

    do {

        opt = \_getch();

    } while (opt<'1' || opt>'2');

    system("cls");

    switch (opt) {

    case '1': {

        f = f1;

        fn = f3;

        fd = fd1;

        break;

    }

    case '2': {

        f = f2;

        fn = f4;

        fd = fd2;

        break;

    }

    }

}

int meniu() {

    system("cls");

    if (f == f1)

        cout << " Functia f1(x)=x^3 - 26x + 43" << endl << endl;

    else cout << " Functia f2(x)=cos(x) + 2x - 0.5" << endl << endl;

    cout << " 1. Selectarea functiei" << endl;

    cout << " 2. Metoda aproximatiei succesive - exactitatea 10^(-6)" << endl;

    cout << " 3. Metoda injumatatirii - exactitatea 10^(-2)" << endl;

    cout << " 4. Metoda Newton - exactitatea 10^(-6)" << endl;

    cout << " 5. Metoda secantelor - exactitatea 10^(-6)" << endl;

    cout << " 6. Iesire" << endl;

    int opt;

    do {

        opt = \_getch();

    } while (opt<'1' || opt>'6');

    system("cls");

    return opt - '0';

}

int main() {

    int opt;

    f = f1;

    fn = f3;

    fd = fd1;

    do {

        switch (opt = meniu()) {

        case 1: {selectFunction();

            break;

        }

        case 2: {aproximatie();

            break;

        }

        case 3: {injumatatire();

            break;

        }

        case 4: {newton();

            break;

        }

        case 5: {secante();

            break;

        }

       }

}

while (opt != 6);

}

**Rezultatele obtinute**

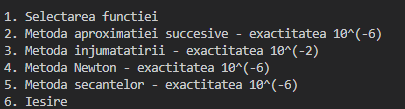


Fig.1 Meniul programului

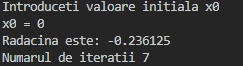


Fig.2 Metoda aproximatiei succesive

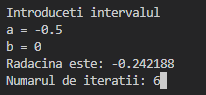
****

Fig.3 Metoda injumatatirii intervalului

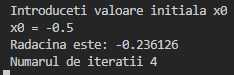


Fig.4 Metoda lui Newton

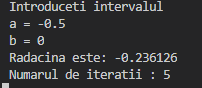


Fig.5.Metoda secantelor

**Compararea rezultatelor**

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **Metoda** | **Radacina** | | **Iteratiile** | | **Eroarea** |
| f1(x) | f2(x) | f1(x) | f2(x) |
| Aproximatiei succesive | 1.93061 | -0.236125 | 16 | 7 | 0.000001 |
| Injumatatirii intervalului | 1.92969 | -0.242188 | 6 | 6 | 0.01 |
| Tangentelor  (Newton) | 1.93061 | -0.236126 | 4 | 4 | 0.000001 |
| Secantelor | 1.93061 | -0.236126 | 5 | 5 | 0.000001 |

**Concluzii**

Efectuind aceasta lucrare de laborator am insusit metodele de rezolvare a ecuatiilor algebrice si transcendente,obtinind aproximativ aceeasi radacina. Totusi cea mai simpla mi s-a parut metoda grafica deoarece nu necesita calcule si usor se poate determina intervalul in care se gaseste solutia,dar cea mai eficienta pare a fi metoda Newton,luind in consideratie ca a facut cele mai putine iteratii pina a fost gasita solutia.