Ministerul Educației, Culturii și Cercetării al Republicii Moldova

Universitatea Tehnică a Moldovei

Facultatea Calculatoare, Informatică şi Microelectronică

Departamentul Ingineria Software și Automatică

RAPORT

Lucrarea de laborator Nr.1

la disciplina **Metode și modele de calcul 1**

VARIANTA 13

A efectuat: st.gr.TI-192

Mereuță Ana

A verificat: Lector Universitar

Godonoaga Anatol

CHIŞINĂU – 2020

Tema: **Rezolvarea numerică a ecuațiilor algebrice și transcendenta. Separarea rădăcinilor**

Scopul lucrării:

1. Să se separe toate rădăcinile reale f(x)= 0, unde y=f(x) este o funcție reală de variabilă reală.
2. Să se determine o rădăcină reală a ecuației date cu ajutorul înjumătățirii intervalului cu o eroare mai mică decât e=0,001;
3. Să se precizeze rădăcina obținută cu exacticitaea de e=0,001, utilizând:
   -metoda aproximărilor succesive;
   -Metoda tangentelor(Newton);
   -Metoda secantelor;
4. Să se compare rezultatele obținute luând în considerație numărul de iterații, evaluările pentru funcții și derivată.

PARTEA PRACTICĂ

Ecuațiile propuse:

1. $(x+1)^3+\ln(x)$
2. $x^3+14x-6$

Mersul lucrării:

Pentru a separa rădăcinile primei ecuații, am folosit metoda grafică, metoda analitică și șirul lui Rolle de separare a soluțiilor.

Șirul lui Rolle

$f(x)= (x+1)^3+\ln(x)$

$f'(x)=3x^2+6x+3+1/x$

x	0,01	0,02	0,1	1
f(x)	-0. 277379	0. 24462	0.74	8.69

Prin urmare pe intervalul (0.01; 1) avem o alternanțe de semn, deci rezultă că avem cel puțin o rădăcină r€(0.01; 1).

Metoda grafică:

$y=(x+1)^3$

$y=-\ln(x)$

$(x+1)^3 =-\ln(x)$

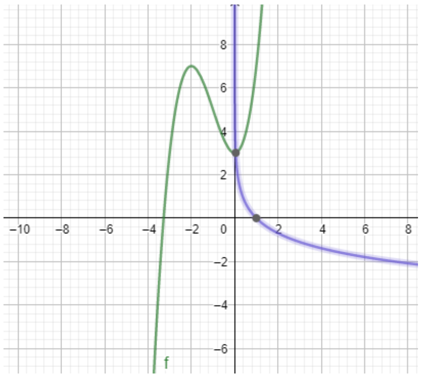


fig 1 Graficele funcțiilor $y=x^3$ și $y=26x-43$

Prin metoda grafică observăm că funcțiile se intersectează în preajma intervalului (0;1)

Metoda analitică

$(x+1)^3+\ln x=0$

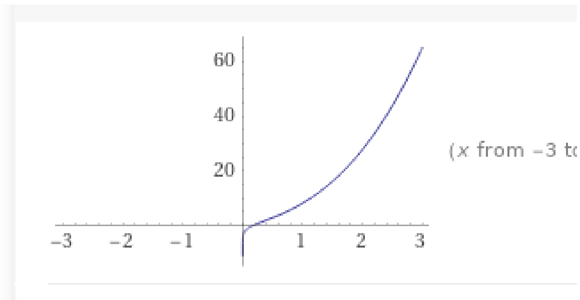


fig 2 Graficul funcției $f(x)= x^3-26x+43=0$

Prin această metodă observăm că funcția are zerou pe intervalul (0;1)

Șirul lui Rolle

$f(x)= x^3+14x-6$

$f'(x)=3x^2+14$

x	0	0,25	0.5	1
f(x)	-6	-2,48	1.125	9

Prin urmare pe intervalul (0,01; 0,5) avem alternanță de semn, deci pe acest interval este minim o soluție.

Metoda grafică

$y=x^3$

$y=-14x+6$

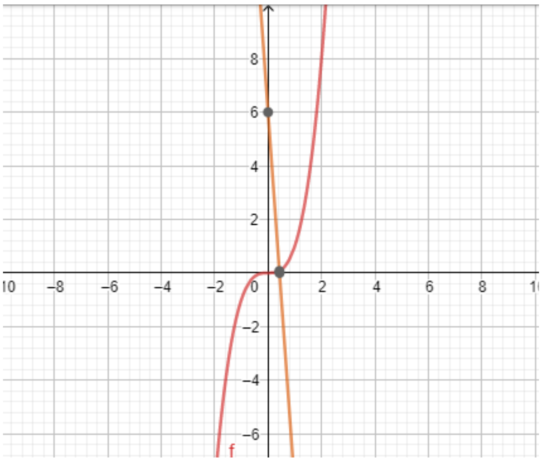


fig 3

Din acest grafic se observă că pe intervalul (0; 1) funcțiile se intersectează, respectiv, pe acest interval este o soluție.

Metoda analitică

$f(x)= x^3+14x-6$

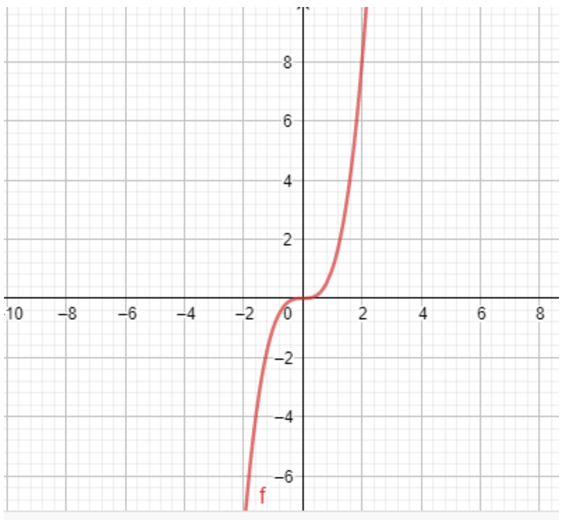


fig 4

Din acest grafic din nou obținem că pe intervalul (0;1) avem o rădăcină reală.

Listing-ul programelor:

```
#include <iostream>
#include <math.h>

using namespace std;
```

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

double f1(double x) {
    return  x*x*x+1+log(x);
}
double f2(double x) {
    return x*x*x+14*x-6;
}

double fd1(double x)
{
    return 3*x*x+6*x+3+1/x;
}
double fd2(double x)
{
    return 3*x*x+14;
}
double f3(double x) {
    return x*x*x+1+log(x);
}
double f4(double x) {
    return x*x*x+1/23;
}

void Met_aproximatiei()
{
    int k = 0;
    double x0, x1, eps = 0.000001;
    cout << "Introduceti valoarea initiala x0" << endl;
    cout << "x0= ";
    cin >> x0;
    while (1)
    {
```

```
        x1 = fn(x0);
        k++;
        if (abs(x1 - x0) < eps) {
            cout << "Radacina este: " << x0 << " Numarul de iteratii este " <<
k << endl;
            break;
        }
        x0 = x1;
    }
}

void Met_injumatatirii() {
    int k = 0;
    double a, b, c = 0;
    double eps = 0.01;
    cout << "Introduceti intervalul a si b :  " << 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 este : " << k;
```

```
}
void Met_Tangentelor() {
    int k = 0;
    double x0, x1, eps = 0.000001;
    cout << "Introduceti valoarea 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;
    }
}
void Met_Secantelor() {
    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;
```

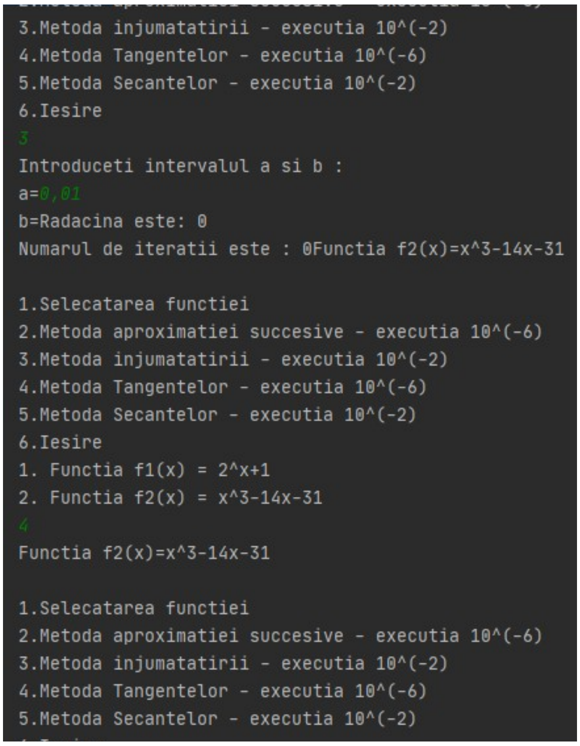
```
    } while (fabs(y - x3) >= eps);
    cout << "Radacina este: " << x3 << endl;
    cout << "Numarul de iteratii: " << n << endl;
}

void selectFunction() {
    cout << "1. Functia f1(x) = 2^x+1" << endl;
    cout << "2. Functia f2(x) = x^3-14x-31" << endl;
    int opt;
    do {
        opt = getchar();

    } while (opt < '1' || opt < '2');
    switch (opt) {
        case'1': {
            f = f1;
            fn = f3;
            fd = fd1;
            break;
        }
        case'2': {
            f = f2;
            fn = f4;
            fd = fd2;
            break;
        }
    }
}
int meniu()
{
    if (f == f1)
        cout << "Functia f1(x)=2^x+1" << endl << endl;
    else cout << "Functia f2(x)=x^3-14x-31" << endl << endl;
```

```
    cout << "1.Selecatarea functiei " << endl;
    cout << "2.Metoda aproximatiei succesive - executia 10^(-6)" << endl;
    cout << "3.Metoda injumatatirii - executia 10^(-2)" << endl;
    cout << "4.Metoda Tangentelor - executia 10^(-6)" << endl;
    cout << "5.Metoda Secantelor - executia 10^(-2)" << endl;
    cout << "6.Iesire" << endl;

    int opt;
    do {
        opt = getchar();
    } while (opt < '1' || opt>'6');
    return opt - '0';
}

int main()
{
    int opt;
    f = f1;
    fn = f3;
    fd = fd1;
    do {
        switch (opt = meniu()) {
            case 1: {
                selectFunction();
                break;
            }
            case 2: {
                Met_aproximatiei();
                break;
            }
            case 3: {
                Met_injumatatirii();
                break;
```

```
            }
            case 4: {
                Met_Tangentelor();
                break;
            }
            case 5: {
                Met_Secantelor();
                break;
            }

        }
    } while (opt != 6);
}
```

Rezultatul :

```
1.Selecatarea functiei
2.Metoda aproximatiei succesive - executia 10^(-6)
3.Metoda injumatatirii - executia 10^(-2)
4.Metoda Tangentelor - executia 10^(-6)
5.Metoda Secantelor - executia 10^(-2)
6.Iesire
1
1. Functia f1(x) = 2^x+1
2. Functia f2(x) = x^3-14x-31
2
Functia f2(x)=x^3-14x-31

1.Selecatarea functiei
2.Metoda aproximatiei succesive - executia 10^(-6)
3.Metoda injumatatirii - executia 10^(-2)
4.Metoda Tangentelor - executia 10^(-6)
5.Metoda Secantelor - executia 10^(-2)
6.Iesire
2
Introduceti valoarea initiala x0
x0=1
 Radacina este: 1 Numarul de iteratii este 1
Functia f2(x)=x^3-14x-31

1.Selecatarea functiei
2.Metoda aproximatiei succesive - executia 10^(-6)
3.Metoda injumatatirii - executia 10^(-2)
```



```
3.Metoda injumatatirii - executia 10^(-2)
4.Metoda Tangentelor - executia 10^(-6)
5.Metoda Secantelor - executia 10^(-2)
6.Iesire
3
Introduceti intervalul a si b :
a=0,01
b=Radacina este: 0
Numarul de iteratii este : 0Functia f2(x)=x^3-14x-31

1.Selecatarea functiei
2.Metoda aproximatiei succesive - executia 10^(-6)
3.Metoda injumatatirii - executia 10^(-2)
4.Metoda Tangentelor - executia 10^(-6)
5.Metoda Secantelor - executia 10^(-2)
6.Iesire
1. Functia f1(x) = 2^x+1
2. Functia f2(x) = x^3-14x-31
4
Functia f2(x)=x^3-14x-31

1.Selecatarea functiei
2.Metoda aproximatiei succesive - executia 10^(-6)
3.Metoda injumatatirii - executia 10^(-2)
4.Metoda Tangentelor - executia 10^(-6)
5.Metoda Secantelor - executia 10^(-2)
```

Concluzie:

În urma efectuării acestei lucrări de laborator, am însușit cîteva din metodele de rezolvare a ecuațiilor algebrice și transcendente. Am aplicat aceste metode în limbaj de programare C++. Mi-am format deprinderi de realizare a functiilor ce implementează algoritmele sus-menționate. Astfel am observat că cea mai efectivă metodă este cea a lui Newton pentru că are cea mai mare precizie și returnează un număr mic de iterații .

BIBLIOGRAFIE

1. https://www.wolframalpha.com/
2. geogebra.com