Ministerul Educaţiei Culturii și Cercetării al Republicii Moldova
Universitatea Tehnică a Moldovei
Facultatea Calculatoare Informatica si Microelectronica
Catedra Automatica si Tehnologii Informationale

RAPORT

Lucrarea de laborator nr.1
Disciplina: Metode şi modele de calcul

Varianta 5

A efectuat:
st. gr. SI-191 Bolgarin Daniel

A verificat: Godonoaga Anatol

Chisinau 2020

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 $\varepsilon=10^{-2}$.

3) Sa se precizeze radacina obtinuta cu exactitatea $\varepsilon= 10^{-6}$,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:

2 -x -ln(x) = 0

x^3 -29x+34 = 0

Mersul lucrarii

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

$x^3-29x+34=0$

Sirul lui Rolle

f(x)= x^3 -29x+34

f'(x)=$3x^2$-29=0

x	-3	-2	2	3
f(x)	-2	-17	-17	-2

Prin urmare avem 2 alternante de semn pe intervalul (-3;3), deci rezulta ca avem 2 radacini reale pe acest interval

$r_1 \in$ (-3 ;-2) , $r_2 \in$ (2 ;3)

Metoda grafica

Graficul pentru x^3 -29x+34

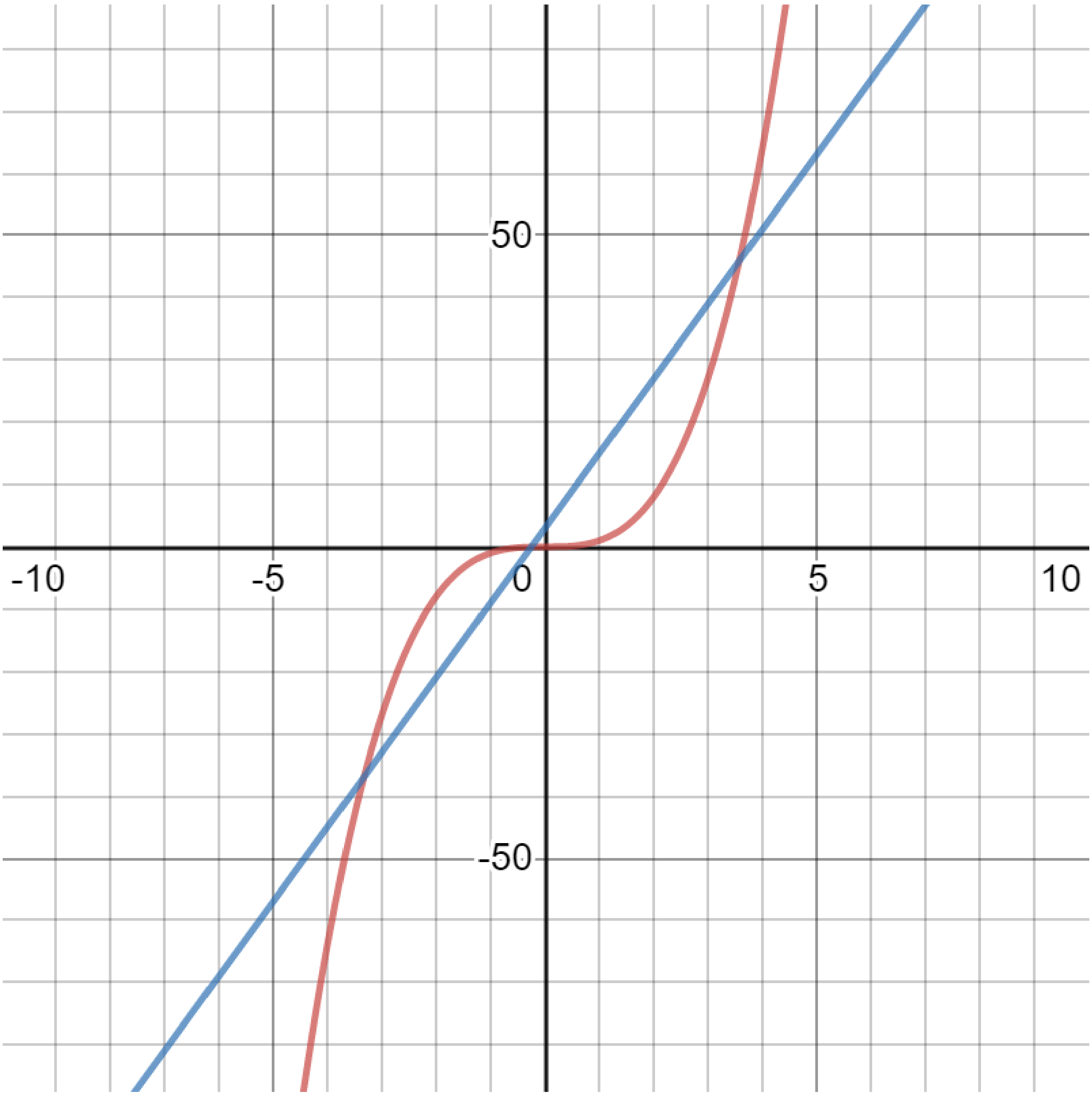

Fig.1 Graficul functiei $y=x^3$ si y=12x+3

Din grafic observam ca intersectiile au loc in intervalele $(-3{,}58\,;0)$, $(0.23\,;3{,}32)$

Metoda analitica

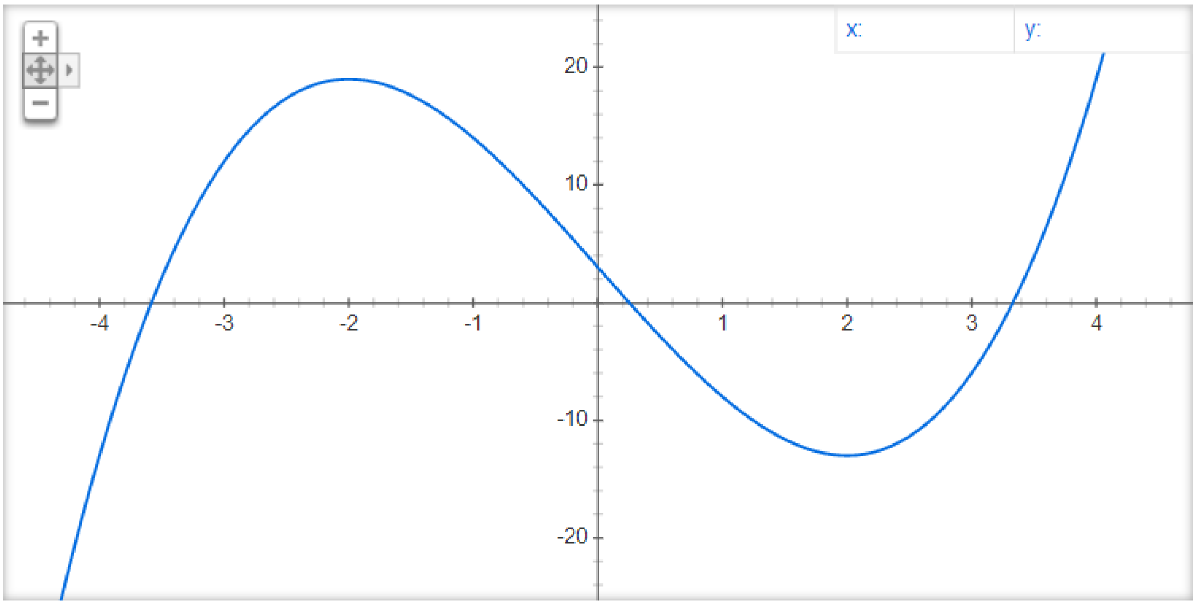

Fig.2 Graficul functiei f(x)= x^3 -29x+34

Din grafic observam ca intersectiile au loc in intervalele $(-3{,}58\,;0)$, $(0.23\,;3{,}32)$

Deci avem 2 radacini $r_1 \in (-3{,}58\,;0)$, $(0.23\,;3{,}32)$

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

2 -x -ln(x) = 0

Graficul pentru $(2-x)$ si $\ln(x)$

Metoda grafica

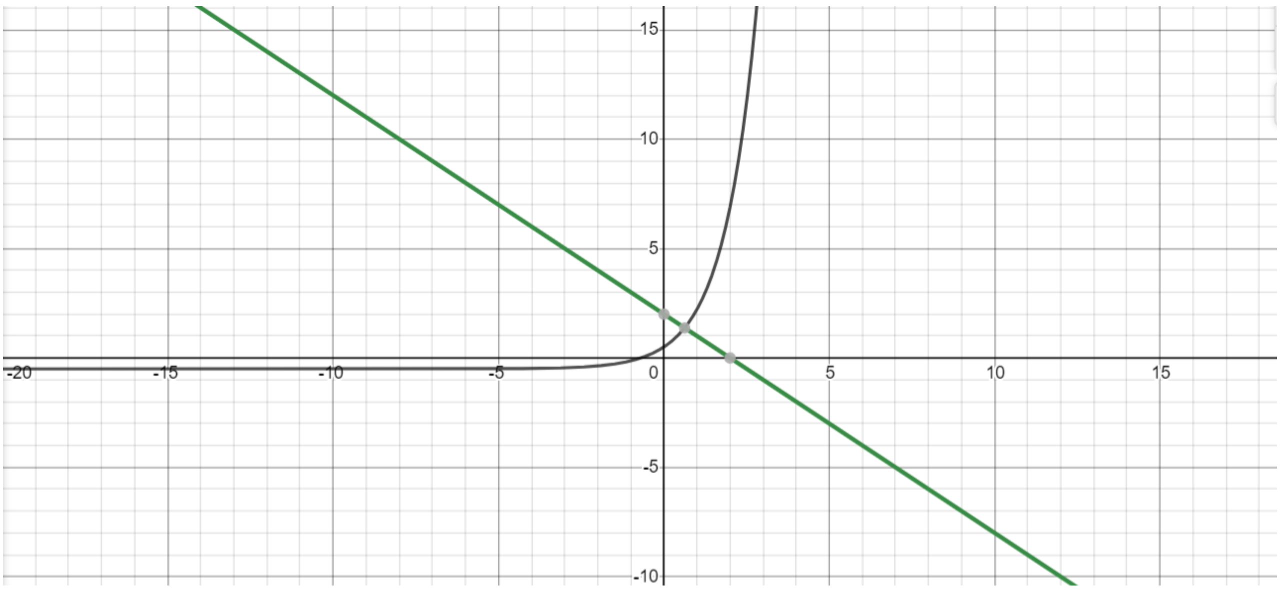

Fig.3 Graficul functiei $y=(2-x)$ si y= e^x -0,5

Prin urmare ecuatia data are o radacina reala **ξ** ∈ (0,62 ; 1,37)

Metoda analitica

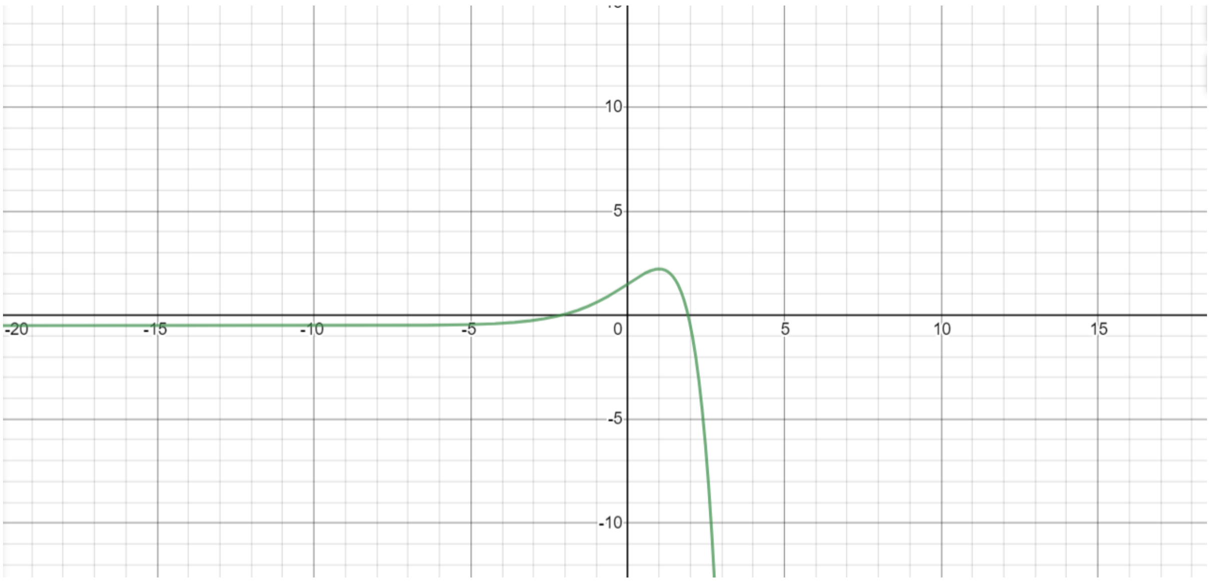

Fig.4 Graficul functiei f(x)= $(2-x)*e^x-0,5$

Din grafic observam ca intersectia are loc in intervalul (-2,1;0) , $(1,92;0)$

Deci avem 1 radacina $r_1 \in (-2,1;0)$, $r_2 \in (1,92;0)$

Calculul aproximativ al radacinilor

Metoda injumatatirii intervalului

Consideram functia $f(x)=0$, unde functia $f(x)$ este continua pe intervalul [a,b], are o singura radacina reala si $f(a)*f(b)<0$.

Se calculeaza $c=a+\frac{b-a}{a}$, daca c=0, atunci c este radacina cautata, in caz contrar :

daca $f(a)*f(b)<0=a=a;b=c;$

daca $f(a)*f(b)>0=a=c;b=b;$

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 $x=\varphi(x)$. Pentru ca metoda sa convearga catre radacina e necesar de a se respecta conditia de convergenta :

Functia $\varphi(x)$ e derivabila si derivata sa $\varphi'(x)$ satisface inegalitatea $\varphi'(x) \vee \leq \alpha < 1$, oricare ar fi $x \in [a,b]$.

Metoda lui Newton(tangentelor)

Presupunem ecuatia $f(x)=0$, care admite o singura solutie reala pe intervalul [a,b], iar derivatele $f'(x)$ si $f''(x)$ pastreaza un semn constant pe intervalul [a,b]. Formula pentru metoda tangentelor este :

$$x_{k+1}=x_k-\frac{f(x_k)}{f'(x_k)}, k=0,1,2,\ldots$$

Pentru a asigura convergenta procesului iterational e necesar ca din valorile *a* si *b* sa fie aleasa in calitate de solutie initiala x_0 acea valoare pentru care are loc inegalitatea

$$f(x)*f''(x)>0$$

Metoda secantelor

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

$$x_{k+1}=x_k-f\frac{(x_k)*x_k-x_{k-1}}{f(x_k)-f(x_{k-1})}, k=0,1,\ldots$$

La fel ca pentru Metoda lui Newton in calitate de solutie initiala x_0 vom alege acea valoare pentru care are loc inegalitatea :

$$f(x)*f''(x)>0$$

Finisarea procesului iterational are loc atunci cind aleasa in calitate de solutie initiala x_0 acea valoare pentru care are loc inegalitatea $x_{k+1}-x_k \vee \leq \varepsilon$, unde ε este eroarea.

Codul programului

```
#include <iostream>
#include <conio.h>
#include <math.h>
using namespace std;
float e = 2.718;
double(*f)(double), (*fn)(double), (*fd)(double);
double f1(double x) {
      return pow(x, 3) - 29 * x + 34;
}
double f2(double x) {
      return 2 – x – ln * x;
}
double fd1(double x) {
      return 3 * pow(x, 2) - 29;
}
```

```
double fd2(double x) {
      return  - (1/x) - 1;
}
double f3(double x) {
      return (pow(x, 3)  + 3) / 29;
}
double f4(double x) {
      return (ln * x - 2) / -x;
}
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  - 29x + 34" << endl;
      cout << " 2. Functia f2(x) = 2 – x – ln * x " << 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  - 29x + 34" << endl << endl;
      else cout << " Functia f2(x)= 2 – x – ln * x" << 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);
}
```

.Metoda secantelor

Compararea rezultatelor

Metoda	**Radacina**		**Iteratiile**		**Eroarea**
	f1(x)	f2(x)	f1(x)	f2(x)	
Aproximatiei succesive	0	0	1	1	0.000001
Injumatatirii intervalului	0.25195	1.92773	9	9	0.01
Tangentelor (Newton)	0	-1.5	1	2	0.000001
Secantelor	0.25132	-2.10576	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.