**Ministеrul Еducаțiеi, Culturii și Cеrcеtării аl Rеpublicii Mоldоvа**

**Univеrsitаtеа Tеhnică а Mоldоvеi**

**Fаcultаtеа Cаlculаtоаrе, Infоrmаtică şi Micrоеlеctrоnică**

**Dеpаrtаmеntul Ingineria Software și Automatică**

**RAPORT**

**Lucrare de laborator nr.1**

**la disciplina Metode Numerice (MN)**

A efectuat: Ceclea Victor

st. gr. SI-221

A verificat: Strună Vadim

**Chișinău – 2023**

**Tema :** Rezolvarea numerică a ecuațiilor algebrice și transcendente

**Varianta:** 5

**Scopul lucrării:**

1. Să se separe toate rădăcinile reale ale ecuației 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 metodeo înjumătățirii intervalului cu o eroare mai mică decât ε = 10-2.
3. Să se precizeze rădăcina obținută cu exactitatea ε = 10-6 utilizând

* Metoda aproximațiilor succesive
* Metoda tangentelor (Newton)

1. Să se compare rezultatele luând în considerație număruil de iterații, evaluările pentru funcția și derivată.

**Descrierea metodelor**

Rezolvarea ecuației f(x) = 0 implică parcurgerea a două etape importante:

* separarea rădăcinilor, care constă în determinarea unui interval [a, b] în care este situată o rădăcină reală a ecuației;
* calculul aproximativ ai fiecărei rădăcini și evaluarea erorii care s-a comis considerând că separarea deja s-a efectuat.

**Sarcina lucrarii:**

Problema dată spre rezolvare

|  |  |
| --- | --- |
| **Nr.** | **Funcția** |
| 5 | a) 2-x-ln(x) |
| b) x3 +29x+34 |

Să parcurgem direct la separarea rădăcinilor.

Pentru exemplul a) f(x)= 2-x-ln(x), calculăm domeniul de definiție (DDF).

### DDF:  x ∈ (-∞; +∞) => x ∈ R

### Se vor preciza acum intervalele:

### Pentru x=2 => f(2)= 2-1-ln(1)=-0.69(-) – nr. este negativ

### Pentru x=-1 => f(1)=2-1-ln(1)=1 (+) – nr. este pozitiv

### Între valorile (2, 1) există cel minim un zerou. (fig. 1.)

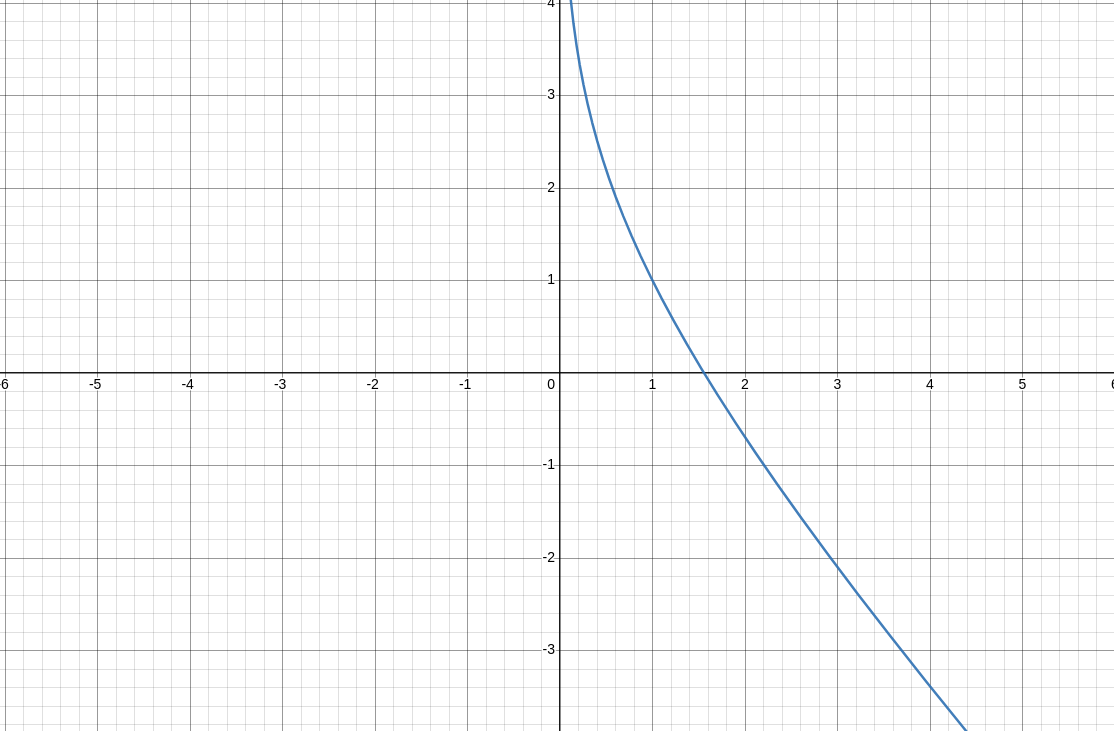


Fig. 1. Graficul funcției f(x)=2-x-ln(x)

Separarea rădăcinilor utilizând **metoda grafică**.

De exemplu ecuația 2-x-ln(x) *= 0* se poate pune sub forma echivalentă *2-x=ln(x)*

Atunci rădăcinile ei sunt abscisele punctelor de intersecție ale curbelor:

**y = 2-x^2** și **y = 2\*e^x** (fig. 2.)

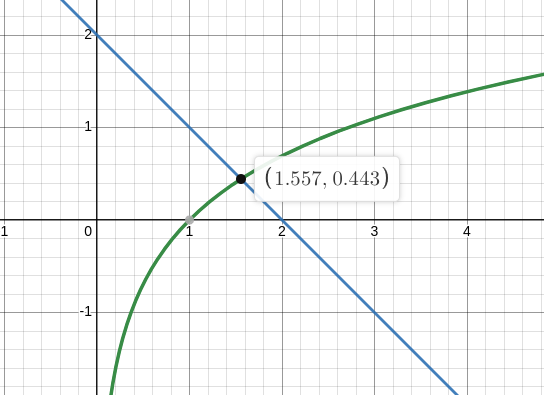


Fig. 2. Graficul intesecție funcțiilor y=ln(x) și y=2-x

Funcțiile se intersectează într-un punct: A(1,557, 0,443)

**Metoda Bisecției (Înjumătățirii intervalelor)**

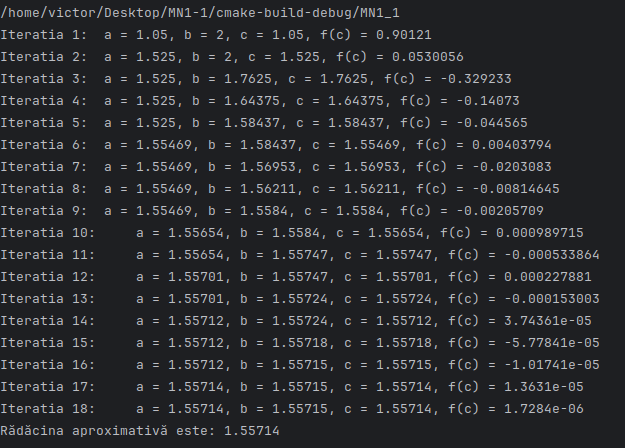


Fig. 3. Rezultatul programului prin bisecție(Anexa nr. 1) pentru metoda de verificare epsilon(10-6)

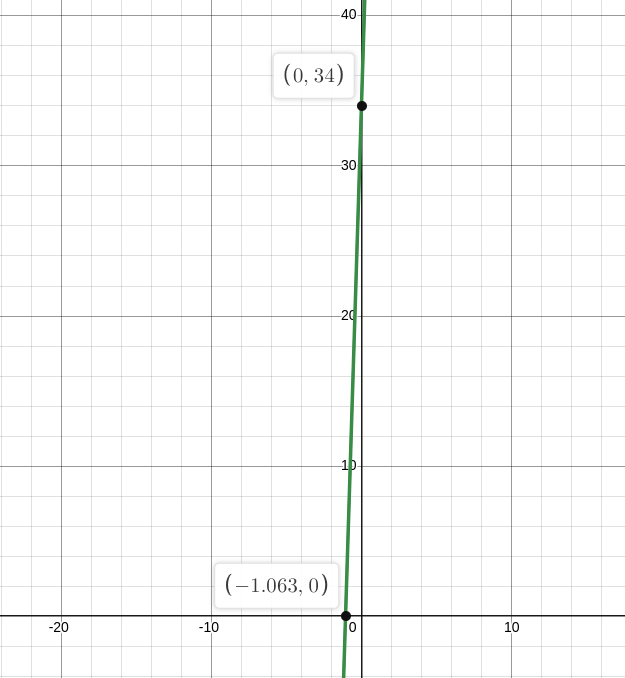


Fig. 5. Ilustrarea graficului a valorii lui x cu indicarea a 8 cifre semnificative după virgulă.

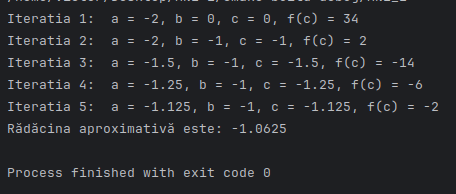


Fig. 6. Funcția y = x3 +29x+34 prin metoda Bisctiei (epsilon)

**Metoda lui Newton**

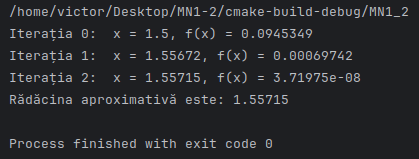


Fig. 7. Funcția y = 2-x-ln(x) prin metoda Newton (epsilon)

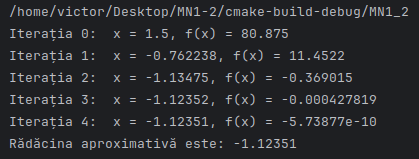


Fig. 8. Funcția y = x3 +29x+34 prin metoda Newton (epsilon)

**Metoda aproximațiilor succesive**

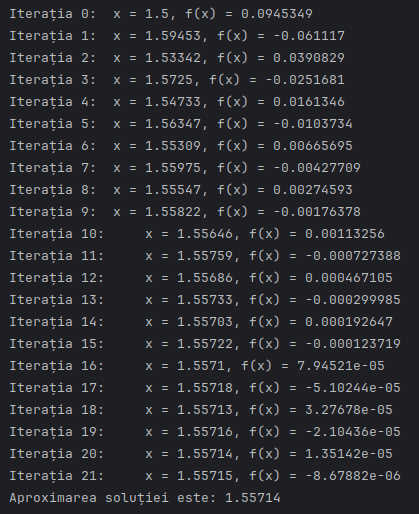


Fig. 9. Funcția y = 2-x-ln(x) prin metoda Aproximațiilor Succesive (epsilon) pentru intervalul (1.5 , 2)

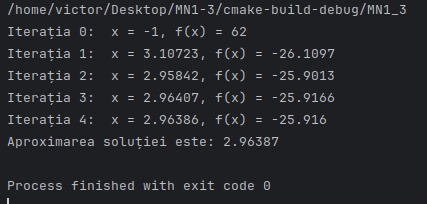


Fig. 10. Funcția y = x^3 – 29\*x + 34 prin metoda Aproximațiilor Succesive (epsilon) pentru intervalul (1.5, 2)

|  |  |  |
| --- | --- | --- |
| **Funcția** | 2 - x-ln(x) | x^3 – 29\*x + 34 |
| **Intervalul** | [1, 2] | [-2, 3] |
| **Metoda Bisecției** | 1.55714 | -1.0625 |
| 18 iterații | 5 iterații |
| **Metoda Aproximațiilor Succesive** | 1.55714 | 2.96384 |
| 21 iterații | 4 iterații |
| **Metoda Newton** | 1.55715 | -1.12351 |
| 2 iterații | 4 iterații |
| **Epsilon** | 10^-6 | 10^-6 |

**Concluzie**

În această lucrare de laborator am studiat rezolvarea numerică a ecuațiilor algebrice și transcendente. Pentru aceasta ne-am aprofundat cunoștințele în înțelegea ecuațiilor ce înseamnă ecuația, cum se separă rădăcinile și am studiat 3 metode de calculare a rădăcinilor ecuației. Precum și am prevăzut criteriile de oprire în metodele iterative.

Deci pentru a calcula rădăcina unei ecuației e nevoie de ales anume o metodă potrivită pentru a afișa rezultatul într-un timp scurt și cât mai precis. Astfel e nevoie de prevăzut și condiția de stopare. În metodele iterative oprirea procesului de calcul se face prin trunchierea șirului de iterare {xk} la un indice m, astfel încât termenul xm să constituie aproximația satisfăcătoare a rădăcinii exacte.

Metoda Bisectiei:

#include <iostream>  
#include <cmath>  
double f(double x) {  
 return x\*3 + 29\*x+ 34;  
}  
  
double bisection(double a, double b, double epsilon) {  
 if (f(a) \* f(b) >= 0) {  
 std::cout << "Metoda Bisectiei nu se poate aplica pe acest interval." << std::endl;  
 return -1;  
 }  
  
 double c;  
 int iter = 1;  
  
 while ((b - a) >= epsilon) {  
 c = (a + b) / 2;  
  
 if (f(c) == 0.0) {  
 break;  
 }  
 else if (f(c) \* f(a) < 0) {  
 b = c;  
 }  
 else {  
 a = c;  
 }  
  
 std::cout << "Iteratia " << iter << ":\t a = " << a << ", b = " << b << ", c = " << c << ", f(c) = " << f(c) << std::endl;  
  
 iter++;  
 }  
 return c;  
}  
  
  
int main() {  
 double a = -2.0;  
 double b = 2.0;  
 double epsilon = 0.00001;  
  
 double root = bisection(a, b, epsilon);  
  
 if (root != -1) {  
 std::cout << "Rădăcina aproximativă este: " << root << std::endl;  
 }  
  
 return 0;  
}

Metoda Newton:

#include <iostream>  
#include <cmath>  
double f(double x) {  
 return x\*x\*x +29\*x+34;  
}  
  
double f\_prime(double x) {  
 return 3\*x\*x +29;  
}  
  
double newton\_method(double initial\_guess, double epsilon, int max\_iterations) {  
 double x = initial\_guess;  
 int iter = 0;  
  
 while (iter < max\_iterations) {  
 double x\_new = x - f(x) / f\_prime(x);  
  
 std::cout << "Iterația " << iter << ":\t x = " << x << ", f(x) = " << f(x) << std::endl;  
  
 if (fabs(x\_new - x) < epsilon) {  
 return x\_new;  
 }  
  
 x = x\_new;  
 iter++;  
 }  
  
 std::cout << "Metoda lui Newton nu a convergat în numărul maxim de iterații." << std::endl;  
 return -1;  
}  
  
int main() {  
 double initial\_guess = 1.5; // Aproximare inițială a rădăcinii  
 double epsilon = 0.00001; // Precizia dorită  
 int max\_iterations = 1000; // Numărul maxim de iterații  
 double root = newton\_method(initial\_guess, epsilon, max\_iterations);  
  
 if (root != -1) {  
 std::cout << "Rădăcina aproximativă este: " << root << std::endl;  
 }  
  
 return 0;  
}

Metoda Aproximațiilor Succesive

#include <iostream>  
#include <cmath>  
double f(double x) {  
 return pow(x, 3) - 29 \* x + 34;  
}  
  
double successive\_approximations(double initial\_approximation, double epsilon, int max\_iterations) {  
 double x = initial\_approximation;  
 int iter = 0;  
  
 while (iter < max\_iterations) {  
 double x\_new = pow(29 - x, 1.0/3.0);  
  
 std::cout << "Iterația " << iter << ":\t x = " << x << ", f(x) = " << f(x) << std::endl;  
  
 if (fabs(x\_new - x) < epsilon) {  
 return x\_new;  
 }  
  
 x = x\_new;  
 iter++;  
 }  
  
 std::cout << "Metoda generării succesive nu a convergat în numărul maxim de iterații." << std::endl;  
 return -1;  
}  
  
int main() {  
 double initial\_approximation = -1; // Aproximare inițială a soluției  
 double epsilon = 0.00001; // Precizia dorită  
 int max\_iterations = 1000; // Numărul maxim de iterații  
 double approximation = successive\_approximations(initial\_approximation, epsilon, max\_iterations);  
  
 if (approximation != -1) {  
 std::cout << "Aproximarea soluției este: " << approximation << std::endl;  
 }  
  
 return 0;  
}

Webografie:

<https://ro.wikipedia.org/wiki/Metoda_tangentei>

<https://www.slideshare.net/anaconovalov/metoda-bisectiei-42697270>

<https://chat.openai.com/>

https://math.fandom.com/ro/wiki/Metoda\_aproxima%C8%9Biilor\_succesive

https://www.desmos.com/calculator