Universitatea Tehnică a Moldovei

Facultatea Calculatoare, Informatică ăi Microelectronică

Departamentul Informatică Sofware și Automate

RAPORT

despre lucrarea de laborator nr. 2

la disciplina Metode și modele de calcul

Tema: Rezolvarea ecuațiilor algebrice și transcendente

A efectuat: st. gr. TI-173 Heghea Nicolae

A verificat: conf. univ Tutunaru Eleonora

Cuprins

[1. Sarcina lucrării 3](#_Toc530467974)

[2. Metoda grafică 3](#_Toc530467975)

[3. Metoda ananitică 4](#_Toc530467976)

[4. Metodele iterative Coardelor, Secantelor 5](#_Toc530467977)

[4.1 Noțiuni generale 5](#_Toc530467978)

[4.2 Schema bloc 6](#_Toc530467979)

[4.3 Codul Sursă 8](#_Toc530467980)

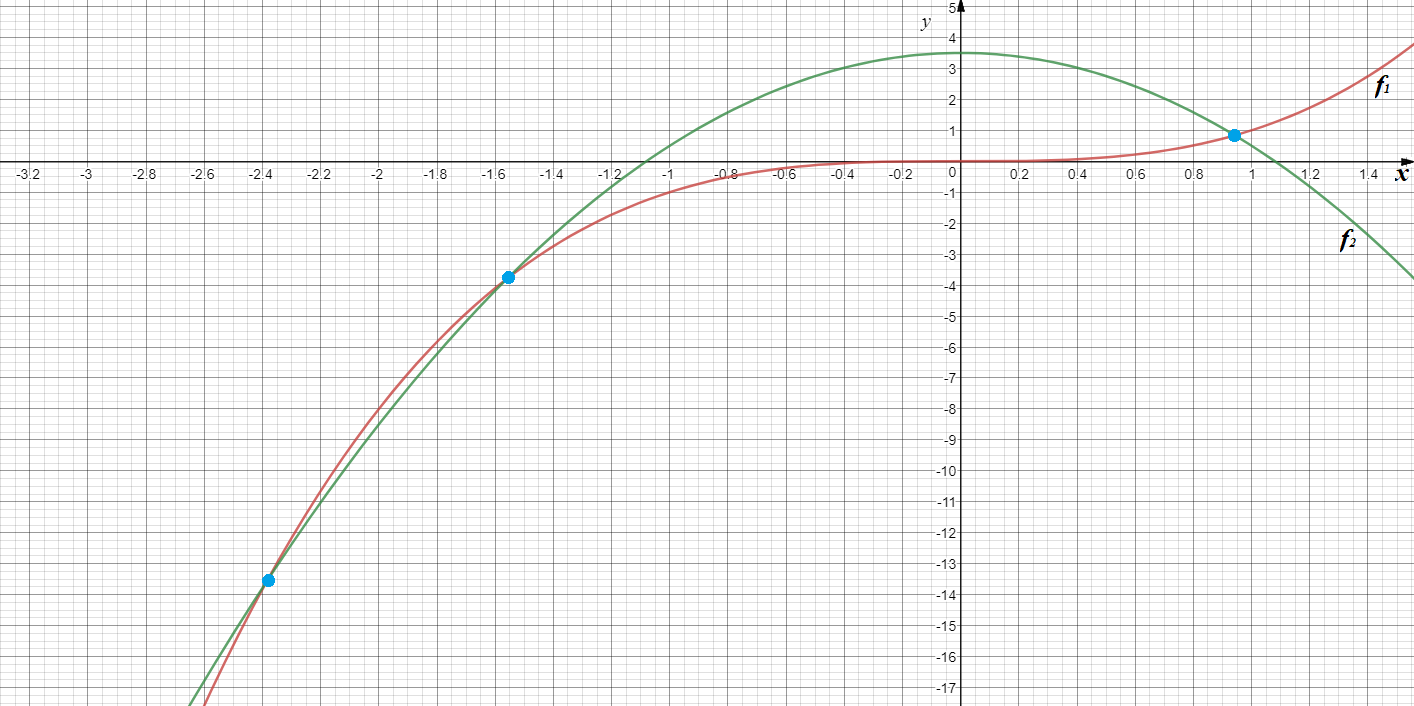
[4.4 Rezultate 10](#_Toc530467981)

[5. Concluzia 11](#_Toc530467982)

# Sarcina lucrării

1. Să se separe analitic și grafic toate rădăcinile reale ale ecuației.
2. Calculează toate rădăcinile reale ale unei ecuații, cu eroare , utilizând metodele Coardelor și Secandelor.

# Metoda grafică



Din grafic observăm că avem 3 intersecții pe intervalele :

Verificare :

# Metoda ananitică

|  |  |
| --- | --- |
|  | nu are soluții |
|  | o singura soluție |

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
|  |  |  | 0.1 | 1 | 2 |
|  |  |  |  |  |  |

|  |
| --- |
|  |
|  |
|  |
|  |
|  |

Rezultă că avem o singură rădăcină pe segmentul

# Metodele iterative Coardelor, Secantelor

## Noțiuni generale

Metoda Coardelor

Formula generală :

Condiția de alegere a punctului de pornire :

|  |  |
| --- | --- |
|  |  |

Condiția de stop :

de obicei

Metoda Secantelor

Formula generală :

Condiția de alegere a punctului de pornire :

|  |  |
| --- | --- |
|  |  |

Condiția de stop :

de obicei

## Schema bloc

Metoda Coardelor

Nu

Da

Da

Nu

Nu

Da

Metoda Secantelor

Nu

Da

Da

Nu

Nu

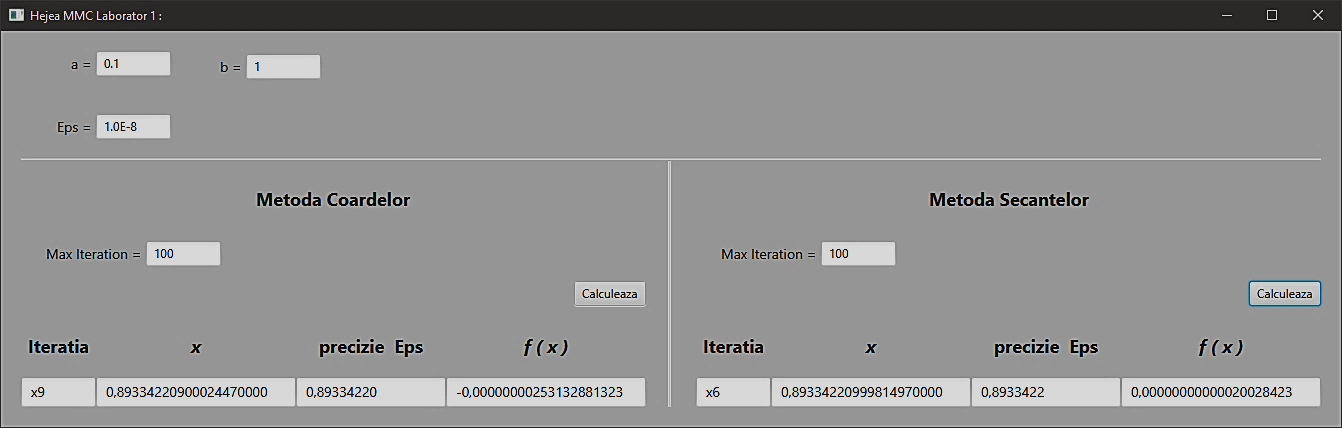
Da

## Codul Sursă

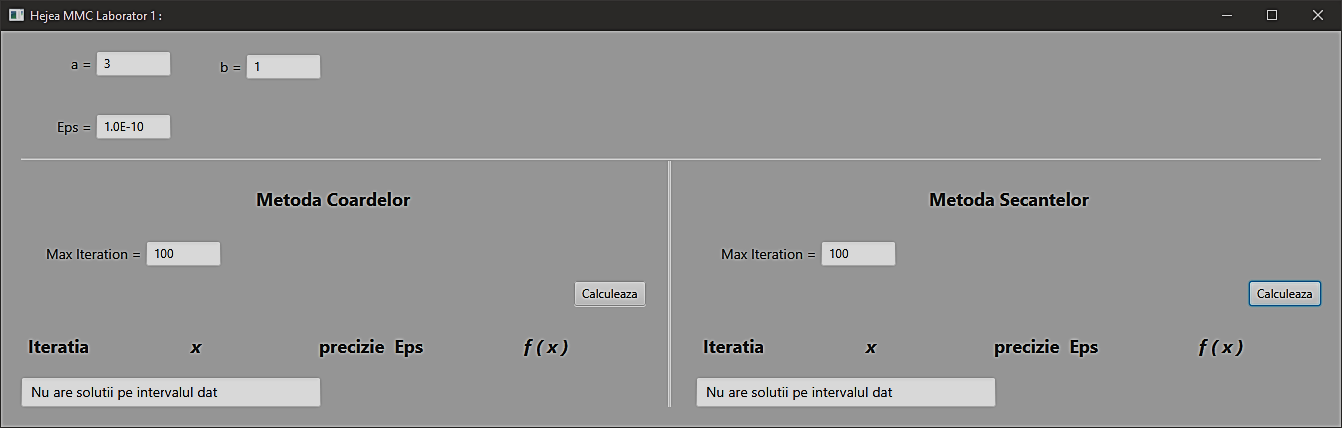
|  |
| --- |
| package app.components;  import static app.components.Function2Data.*F*; import static app.components.Function2Data.*F2*; import static java.lang.Math.*abs*;  public class Methods {  public double eps;  public double a;  public double b;  public double[] xC;  public double[] fxC;  public double[] xS;  public double[] fxS;  public int nrItrC;  public int nrItrS;  public int nrItrMaxChord;  public int nrItrMaxSecant;  public String isSolC;  public String isSolS;    public void runChord() {  double xf, fa, fb, fxf, q, w;    xC = new double[nrItrMaxChord];  fxC = new double[nrItrMaxChord];  nrItrC = 0;  isSolC = "";    validData();  fa = *F*(a);  fb = *F*(b);    if (fa \* fb < 0) {    if (fa \* *F2*(a) < 0) {  xC[0] = a;  xf = b;  } else {  xC[0] = b;  xf = a;  }    fxC[0] = *F*(xC[0]);  fxf = *F*(xf);    do {  nrItrC++;  q = fxC[nrItrC - 1] \* (xf - xC[nrItrC - 1]);  w = fxf - fxC[nrItrC - 1];    xC[nrItrC] = xC[nrItrC - 1] - q / w;  fxC[nrItrC] = *F*(xC[nrItrC]);  } while (*abs*(xC[nrItrC] - xC[nrItrC - 1]) > eps);  } else {  isSolC = "Nu are solutii pe intervalul dat";  }  }    public void runSecant() {  double fa, fb, q, w;    xS = new double[nrItrMaxChord];  fxS = new double[nrItrMaxChord];  nrItrS = 0;  isSolS = "";    validData();  fa = *F*(a);  fb = *F*(b);  if (fa \* fb < 0) {  if (fa \* *F2*(a) > 0) {  xS[0] = a;  xS[1] = a + eps;  } else {  xS[0] = b;  xS[1] = b - eps;  }    fxS[0] = *F*(xS[0]);  fxS[1] = *F*(xS[1]);  nrItrS = 1;  do {  nrItrS++;    q = fxS[nrItrS - 1] \* (xS[nrItrS - 1] - xS[nrItrS - 2]);  w = fxS[nrItrS - 1] - fxS[nrItrS - 2];    xS[nrItrS] = xS[nrItrS - 1] - q / w;  fxS[nrItrS] = *F*(xS[nrItrS]);    } while (*abs*(xS[nrItrS] - xS[nrItrS - 1]) >= eps);  } else {  isSolS = "Nu are solutii pe intervalul dat";  }  }  private void validData() {  if (a > b) {  double temp = a;  a = b;  b = temp;  }  } } |
| package app.components;  public class Function2Data {  // F(x).  public static double F(double x) {  return x \* x \* x - 0.2 \* x \* x + 0.5 \* x - 1.0;  }  // derivata 1 F'(x).  public static double F1(double x) {  return 3 \* x \* x - 0.4 \* x + 0.5;  }  // derivata 2 F"(x).  public static double F2(double x) {  return (6 \* x - 0.4);  }  } |

## Rezultate

Când are soluții :



Când nu are soluții :



# Concluzia

Separarea rădăcinilor prin metoda grafică este simplu și ușor, dar sunt funcții care nu pot fi reprezentate ușor pe grafic. Pentru asemenea funcții se aplică metoda analitică de separare a rădăcinilor. Care ne dă același rezultat.

Cele două metode iterative converg destul de repede. Metodele au atât avantaje cât și dezavantaje. Și aceste 2 metode sunt similare.

Avantaje :

1. nu au nevoie de derivate în iterațiilor
2. necesită evaluare doar la o singură funcție

Dezavantaje :

1. poate să nu conveargă
2. are probleme atunci când Aceasta inseamnă ca graful funcției este tangent la axa OX.