Universitatea Tehnica a Moldovei
Facultatea Calculatoare, Informatica si Microelectronica
Departamentul Informatica Sofware si Automate

RAPORT

despre lucrarea de laborator nr. 1
la disciplina Metode si modele de calcul

Tema:

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

A verificat: Tutunaru

Chișinău 2018

Cuprins

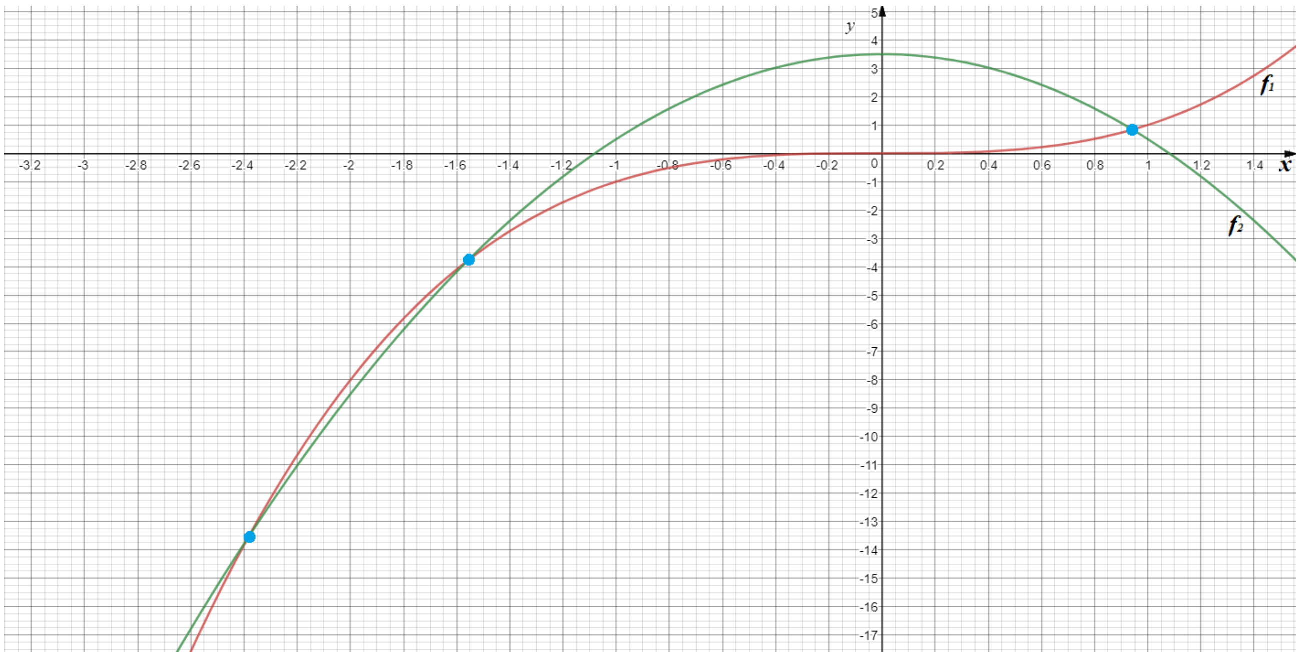
1. Metoda grafică

$$f(x) = x^3 + 3x^2 - 3.5 = 0$$
$$=> x^3 = -3x^2 + 3.5$$

sau

$f_1(x) = x^3$

$f_2(x) = -3x^2 + 3.5$



Din grafic observăm că avem 3 intersecții pe intervalele :

$[-3\,;\,-2]$, $[-2\,;\,-1]$, $[-1\,;\,1]$.

Verificare :

$f(-3) = (-3)^3 + 3(3)^2 - 3.5 = -27 + 27 - 3.5 = -3.5 = -$

$f(-2) = (-2)^3 + 3(2)^2 - 3.5 = \quad -8 + 12 - 3.5 = +0.5 = +$

$f(-1) = (-1)^3 + 3(1)^2 - 3.5 = \quad -1 + \ 3 - 3.5 = -0.5 = -$

$f(1) \ = \ (1)^3 + 3(1)^2 - 3.5 = \quad 1 + \ 3 - 3.5 = +0.5 = +$

2. Metoda ananitică

$$f(x) = x^3 - 0.2x^2 + 0.5x - 1$$

$$k = 1 + \frac{a_{max}}{|a_0|} => k = 1 + \frac{1}{1} = 2 => k = [-2\ ,2\ ]$$

$f'(x) = 3x^2 - 0.4x + 0.5$

$\Delta = 0.4^2 - 4 * 3 * 0.5 = -5.84 < 0$

nu are soluții

$f''(x) = 6x - 0.4$

$x = \frac{0.4}{6}$

o singura soluție

x	-2	-1	0.1	1	2
$semn\ f(x)$	$-$	$-$	$-$	$+$	$+$

$f(-2) = (-2)^3 - 0.2(-2)^2 + 0.5(-2) - 1 = -10.8$

$f(-1) = (-1)^3 - 0.2(-1)^2 + 0.5(-1) - 1 = -2.7$

$f(0.1) = (0.1)^3 - 0.2(0.1)^2 + 0.5(0.1) - 1 = -0.951$

$f(1) = (1)^3 - 0.2(1)^2 + 0.5(1) - 1 = +0.3$

$f(2) = (2)^3 - 0.2(2)^2 + 0.5(2) - 1 = +7.2$

$f'(0.1) = 3(0.1)^2 - 0.4(0.1) + 0.5 = +0.49 => +$

$f'(1) = 3(1)^2 - 0.4(1) + 0.5 = +3.1 => +$

$f''(0.1) = 6(0.1) - 0.4 = +0.2 => +$

$f''(1) = 6(1) - 0.4 = +5.6 => +$

Rezultă că avem o singură rădăcină pe segmentul [0.1 ; 1] .

3. Metodele iterative Coardelor, Secantelor

3.1 Noțiuni generale

Metoda Coardelor

Formula generală : $x_{n+1} = x_n \cdot \frac{f(x_n) \cdot (xf - x_n)}{f(xf) - f(x_n)}$

Condiția de alegere a punctului de pornire :

$$f(a) \cdot f''(a) < 0 \; => \begin{cases} x_0 = a \\ x_f = b \end{cases} \qquad f(a) \cdot f''(a) \geq 0 \; => \begin{cases} x_0 = b \\ x_f = a \end{cases}$$

Condiția de stop : $|x_{n+1} - x_n| \leq \xi$

de obicei $\xi = 10^{-4}$

Metoda Secantelor

Formula generală : $x_{n+1} = x_n \cdot \frac{f(x_n) \cdot (x_n - x_{n-1})}{f(x_n) - f(x_{n-1})}$

Condiția de alegere a punctului de pornire :

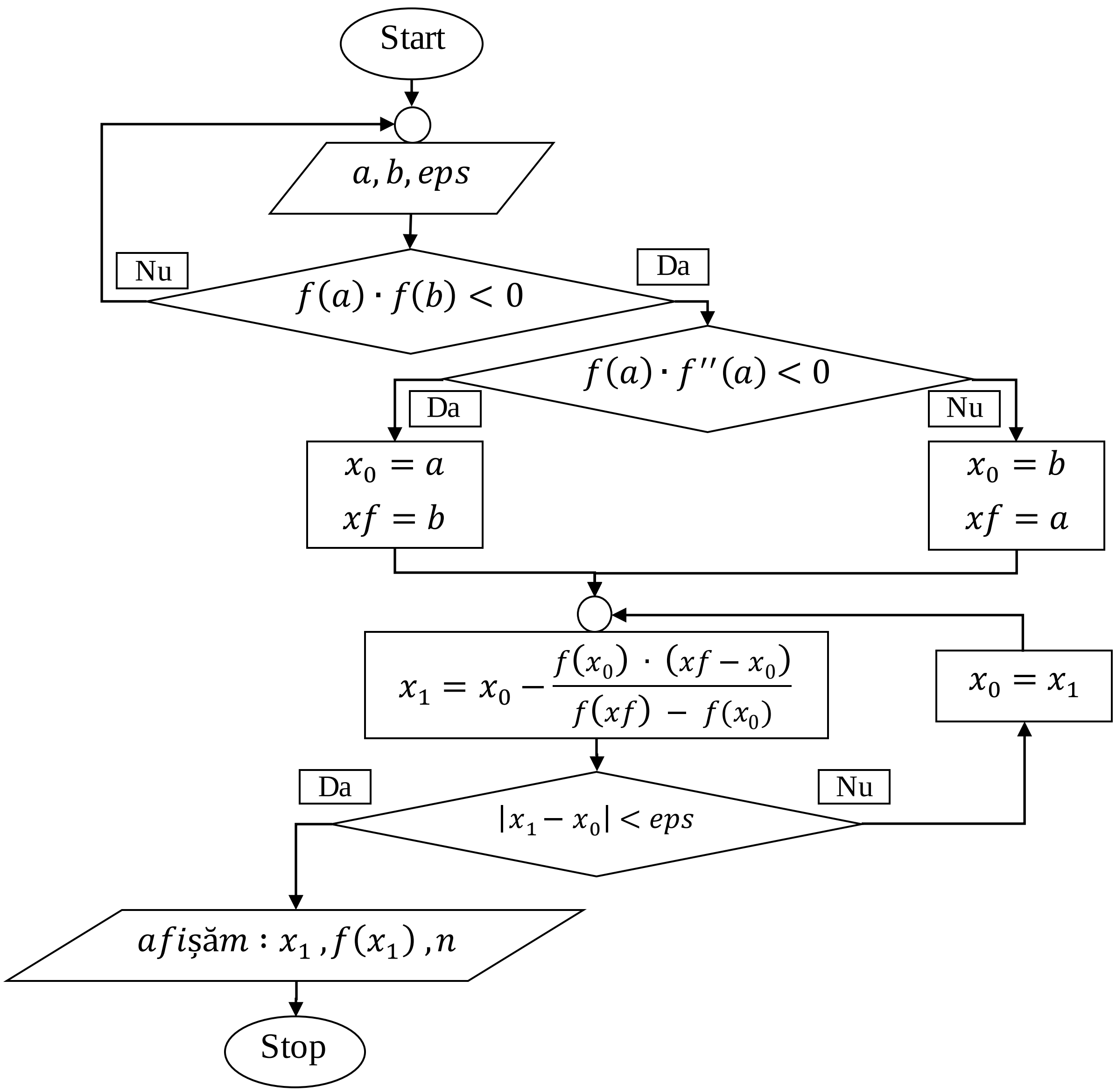
$$f(a) \cdot f''(a) < 0 \; => \begin{cases} x_0 = b \\ x_1 = b - \xi \end{cases} \qquad f(a) \cdot f''(a) \geq 0 \; => \begin{cases} x_0 = a \\ x_1 = a + \xi \end{cases}$$

Condiția de stop : $|x_{n+1} - x_n| \leq \xi$

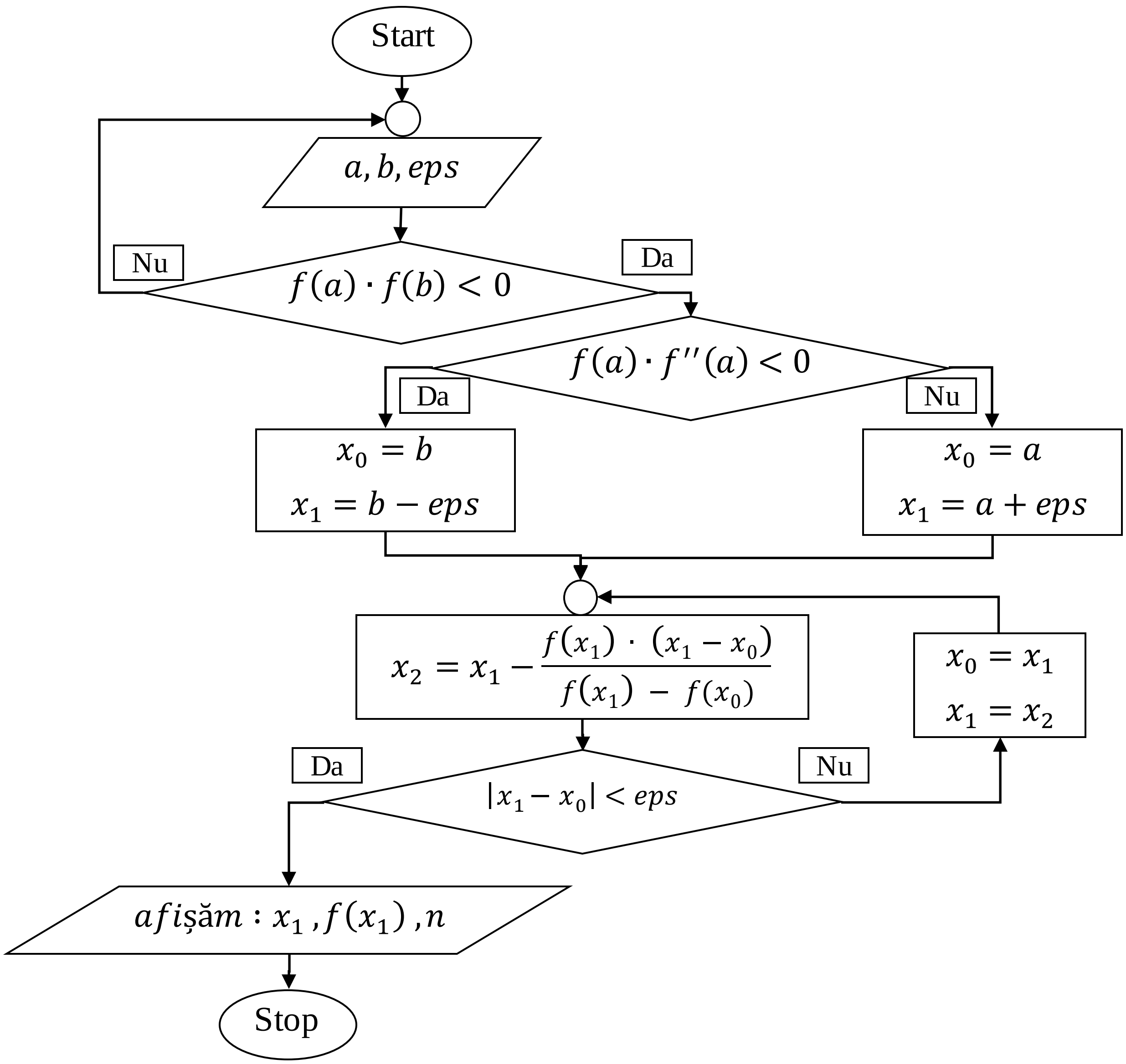
de obicei $\xi = 10^{-4}$

3.2 *Schema bloc*

Metoda Coardelor



Metoda Secantelor



3.3

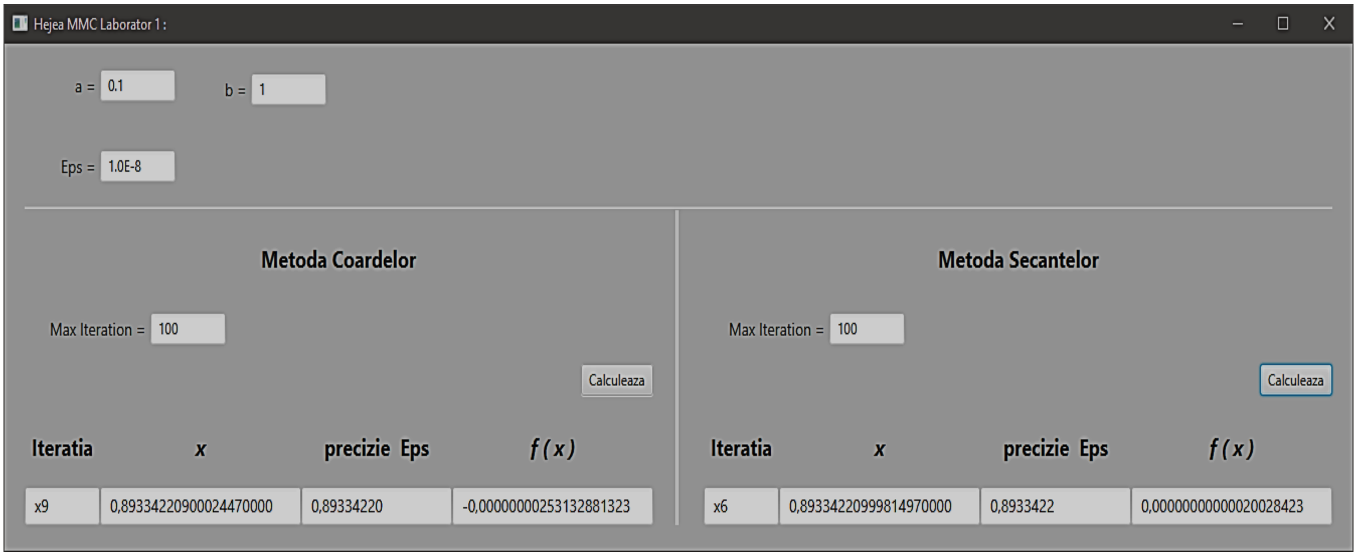
```
package app.components;

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 void runChord() {
        double xf, fa, fxf, q, w;

        xC = new double[nrItrMaxChord];
        fxC = new double[nrItrMaxChord];

        validData();
        fa = F(a);

        if (fa * F(b) < 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);
        nrItrC = 0;

        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);
    }
}

public void runSecant() {
    double fa, q, w;

    xS = new double[nrItrMaxChord];
    fxS = new double[nrItrMaxChord];
    validData();
    fa = F(a);

    if (fa * F(b) < 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);
        }
    }

    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);
    }
}
```

3.4 *Rezultate*



4. 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 $f'(\beta) = 0$. Aceasta inseamnă ca graful funcției este tangent la axa OX.