# A elaborat: Sandiuc Vitalie, IS21Z

**A verificat: Țîcău Vitalie**

# PROIECT DE PROGRAM Nr. 1. REZOLVAREA ECUAŢIILOR NELINIARE

# Tema: Rezolvarea ecuatiilor neliniare

## I. Formularea problemei

De determinat toate soluţiile ecuaţiei x3+4x2-37x-148= 0.cu exactitatea dată *10-9>0*, folosind în acest scop 7 metode:

* metoda bisecţiei;
* metoda iteraţiei;
* metoda lui Tangentelor;
* metoda modificată a tangentelor;
* metoda secantelor;
* metoda coardelor;
* metoda mixtă a tangentelor şi coardelor.

Funcţia *y = f(x)* este o funcţie reală de variabilă reală şi continuă pe domeniul ei de definiţie.

## II. Datele iniţiale

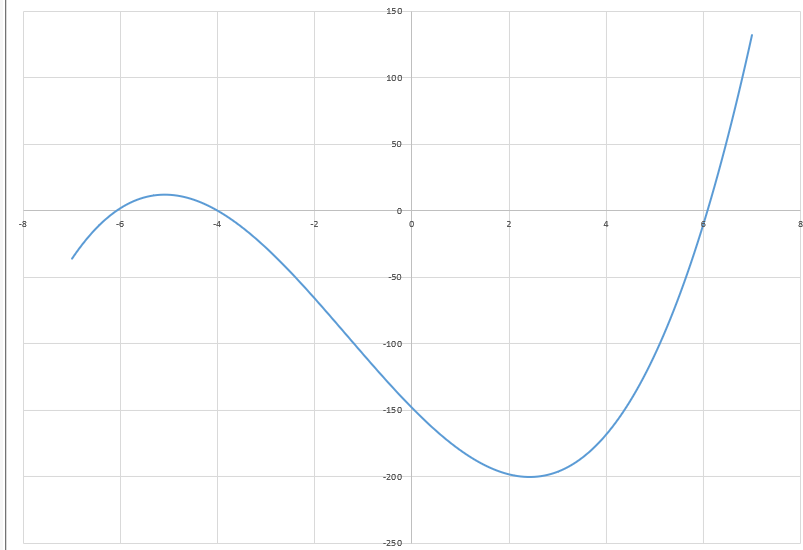
Expresia funcției f(x) de forma unei ecuații de gradul III: x3+4x2-37x-148= 0.

h = 0.7; ε = 10–9.

## III. Studiu preliminar

Problema formulată se rezolvă în 2 etape:

1. Se separă soluţiile ecuaţiei date în formă grafică din Microsoft Excel:



A = -7; B = 7; h = 0.07. Solutiile separate se afla pe segmentele: **[-6.09; -6.02]**; **[ -4.06; -3.99]**; Segmentul **[ 6.02; 6.09]**.

1. La metoda iteratiei simple am utilizat urmatorii :

; ; -> rezultatul e aproape de 0, respsectiv algoritmul va merge mai rapid pentru rădăcinile [-6.09; -6.02] și [6.02; 6.09]

; ; pentru rădăcina [3.99; 4.06]

## IV. Programul

## #include <iostream>

## #include <iomanip>

## #include <stdio.h>

## #include <math.h>

## #include <conio.h>

## using namespace std;

## double eps; // = 1e-12;

## double Bisectie(double a, double b, int& k);

## double Tangenta(double a, double b, int& k);

## double Modificata\_Tangentei(double a, double b, int& k);

## double Secante(double a, double b, int& k);

## double Coarde(double a, double b, int& k);

## double Mixta(double a, double b, int& k);

## double Iteratie(double a, double b, int& k);

## double f(double x)

## {

## return x \* x \* x + 4 \* x \* x - 37 \* x - 148;

## }

## double fpr(double x)

## {

## return 3 \* x \* x + 8 \* x - 37;

## }

## double fi13(double x) {

## double a, b;

## a = 4 \* x \* x - 37 \* x - 148;

## b = pow(a \* a, 1.0 / 6);

## return (x > 0) ? +b : -b;

## }

## double fi13p(double x) {

## double a, b;

## a = 4 \* x \* x - 37 \* x - 148;

## b = pow(a \* a, 1.0 / 3);

## return (8 \* x - 37) / (3 \* b);

## }

## double fi2(double x) {

## return (x \* x \* x + 4 \* x \* x - 148) / 37;

## }

## double fi2p(double x) {

## return (3 \* x \* x + 8 \* x) / 37;

## }

## double fi(double x) {

## if (fabs(fi13p(x)) < 1) {

## return fi13(x);

## }

## else if (fabs(fi2p(x)) < 1) {

## return fi2(x);

## }

## return 1e+38;

## }

## struct interval {

## double left;

## double right;

## } r[10];

## int Separare(double A, double B, double h)

## {

## int n = 0; //numarul de solutii

## double x0, x;

## x0 = A;

## do {

## x = x0 + h;

## if (f(x) == 0) {

## printf("\n %5.2lf este solutie exacta ", x);

## x += eps;

## }

## else if (f(x) \* f(x0) < 0) {

## r[n].left = x0;

## r[n].right = x;

## n++;

## }

## x0 = x;

## } while (x < B);

## return n;

## }

## int main()

## {

## int i, n, k;

## double A = -7, B = 7, h; // = 0.07;

## double x;

## cout << "Dati pasul h= \n";

## cin >> h;

## cout << "Dati precizia de calcul Epsilon: \n";

## cin >> eps;

## cout << endl;

## cout << "Proiect Nr:1 realizat de Sandiuc Vitalie";

## n = Separare(A, B, h);

## int j = 0;

## for (i = 0; i < n; i++) {

## printf("\n\n Pe [%5.2lf; %5.2lf] este solutie separata \n", r[i].left, r[i].right);

## for (j = 1; j <= 63; j++) {

## printf("=");

## }

## printf("\n| Metoda ");

## printf("| Nr. aprox ");

## printf("| Aproximatie ");

## printf("| Verificare |\n");

## for (j = 1; j <= 63; j++) {

## printf("=");

## }

## x = Bisectie(r[i].left, r[i].right, k);

## printf("\n| Bisectie | %d | %15.12lf | %11.3le |", k, x, f(x));

## x = Tangenta(r[i].left, r[i].right, k);

## printf("\n| Tangenta | %d | %15.12lf | %11.3le |", k, x, f(x));

## x = Modificata\_Tangentei(r[i].left, r[i].right, k);

## printf("\n| Mod. Tan. | %d | %15.12lf | %11.3le |", k, x, f(x));

## x = Secante(r[i].left, r[i].right, k);

## printf("\n| Secante | %d | %15.12lf | %11.3le |", k, x, f(x));

## x = Coarde(r[i].left, r[i].right, k);

## printf("\n| Coarde | %d | %15.12lf | %11.3le |", k, x, f(x));

## x = Mixta(r[i].left, r[i].right, k);

## printf("\n| Mixta | %d | %15.12lf | %11.3le |", k, x, f(x));

## x = Iteratie(r[i].left, r[i].right, k);

## printf("\n| Iteratie | %d | %15.12lf | %9.3le |", k, x, f(x));

## cout << endl;

## for (j = 1; j <= 63; j++) {

## printf("=");

## }

## }

## return 0;

## }

## double Bisectie(double a, double b, int& k)

## {

## double x;

## k = 0;

## do {

## k++;

## x = (a + b) / 2;

## if (f(x) == 0)

## return x;

## if (f(a) \* f(x) < 0)

## b = x;

## else

## a = x;

## } while (b - a > eps);

## return (a + b) / 2;

## }

## double Tangenta(double a, double b, int& k)

## {

## double x, x0, delta;

## k = 0;

## x0 = (a + b) / 2;

## do {

## k++;

## x = x0 - f(x0) / fpr(x0);

## if (f(x) == 0)

## return x;

## delta = fabs(x - x0);

## x0 = x;

## } while (delta > eps);

## return x;

## }

## double Modificata\_Tangentei(double a, double b, int& k)

## {

## double x, x0, delta, derivata;

## k = 0;

## x0 = (a + b) / 2;

## derivata = fpr(x0);

## do {

## k++;

## x = x0 - f(x0) / derivata;

## if (f(x) == 0)

## return x;

## delta = fabs(x - x0);

## x0 = x;

## } while (delta > eps);

## return x;

## }

## double Secante(double a, double b, int& k)

## {

## double x, x0, x1, delta;

## k = 1;

## x0 = a;

## x1 = b;

## do {

## k++;

## x = x1 - (x1 - x0) \* f(x1) / (f(x1) - f(x0));

## if (f(x) == 0)

## return x;

## delta = fabs(x - x1);

## x0 = x1;

## x1 = x;

## } while (delta > eps);

## return x;

## }

## double Coarde(double a, double b, int& k)

## {

## double x, x0, x1, delta;

## k = 1;

## x0 = b;

## x1 = a;

## do {

## k++;

## x = x1 - (x1 - x0) \* f(x1) / (f(x1) - f(x0));

## if (f(x) == 0)

## return x;

## delta = fabs(x - x1);

## x1 = x;

## } while (delta > eps);

## return x;

## }

## double Mixta(double a, double b, int& k)

## {

## double x, x0, x1, delta;

## k = 1;

## x0 = (a + b) / 2;

## x1 = x0 - f(x0) / fpr(x0);

## if (f(x1) == 0)

## return x1;

## do {

## k++;

## x = x1 - (x1 - x0) \* f(x1) / (f(x1) - f(x0));

## if (f(x) == 0)

## return x;

## delta = fabs(x - x1);

## x0 = x1;

## x1 = x;

## } while (delta > eps);

## return x;

## }

## double Iteratie(double a, double b, int& k)

## {

## k = 0;

## double x0 = (a + b) / 2;

## double x;

## double delta;

## do {

## k++;

## x = fi(x0);

## if (f(x) == 0)

## return x;

## delta = fabs(x - x0);

## x0 = x;

## } while (delta > eps);

## return x;

## }

## V. Rezultatele sugestive

Dati pasul h=

0.07

Dati precizia de calcul Epsilon:

1e-9

Proiect Nr:1 realizat de Sandiuc Vitalie

Pe [-6.09; -6.02] este solutie separata

===============================================================

| Metoda | Nr. aprox | Aproximatie | Verificare |

===============================================================

| Bisectie | 27 | -6.082762530185 | 2.862e-009 |

| Tangenta | 4 | -6.082762530298 | 8.868e-015 |

| Mod. Tan. | 6 | -6.082762530283 | 3.898e-010 |

| Secante | 6 | -6.082762530298 | 8.868e-015 |

| Coarde | 7 | -6.082762530315 | -4.263e-010 |

| Mixta | 5 | -6.082762530298 | 8.868e-015 |

| Iteratie | 62 | -6.082762527343 | 7.488e-008 |

===============================================================

Pe [-4.06; -3.99] este solutie separata

===============================================================

| Metoda | Nr. aprox | Aproximatie | Verificare |

===============================================================

| Bisectie | 27 | -4.000000000186 | 3.911e-009 |

| Tangenta | 3 | -4.000000000000 | 0.000e+000 |

| Mod. Tan. | 6 | -4.000000000001 | 1.457e-011 |

| Secante | 5 | -4.000000000000 | 0.000e+000 |

| Coarde | 6 | -4.000000000000 | 1.026e-012 |

| Mixta | 4 | -4.000000000000 | 0.000e+000 |

| Iteratie | 21 | -4.000000000578 | 1.214e-008 |

===============================================================

Pe [ 6.02; 6.09] este solutie separata

===============================================================

| Metoda | Nr. aprox | Aproximatie | Verificare |

===============================================================

| Bisectie | 27 | 6.082762530185 | -1.385e-008 |

| Tangenta | 4 | 6.082762530298 | -4.297e-014 |

| Mod. Tan. | 5 | 6.082762530300 | 1.842e-010 |

| Secante | 5 | 6.082762530298 | -4.297e-014 |

| Coarde | 5 | 6.082762530298 | -2.292e-011 |

| Mixta | 4 | 6.082762530298 | -4.297e-014 |

| Iteratie | 9 | 6.082762530341 | 5.264e-009 |

**VI. Concluzii**

* Pe segmentul [-6.09; -6.02], [ 6.02; 6.09] cea mai rapida metoda este Tangentelor. Pe segmentul

[ -4.06; -3.99] cele mai rapide metode sunt metoda Tangentelor si Mixta.

* Metodele Tangentelor, Newton Modificat, Secantelor, Coardelor, Mixta converg rapid, deoarece numărul de aproximații nu depășește 10.
* Metoda iterației lucrează cel mai rapid pe segmentul [6.02; 6.09] pentru ca este cel mai aproape de 0.
* Eroarea obţinuta nu e mai mare decât eroarea admisibilă, dar are loc depășirea la metoda iterației.