function [miejsceZerowe, bledyRozwiazan] = **NewtonRaphson1**(f, df, x0, maksIteracji, eps)

%NEWTONRAPSOD1 Znajdowanie miejsca zerowego metoda Newtona-Raphsona I rzedu

miejsceZerowe = x0;

%Bledy roziazan beda wartosciamy bezwzglednymy f(xi)

bledy = zeros(1, maksIteracji+1);

bledy(1) = abs(f(miejsceZerowe));

i = 1;

while (abs(f(miejsceZerowe)) > eps) && (i <= maksIteracji)

miejsceZerowe = miejsceZerowe - f(miejsceZerowe)/df(miejsceZerowe);

i = i+1;

bledy(i) = (abs(f(miejsceZerowe)));

end

bledyRozwiazan = bledy(1, 1:i);

end

function [miejsceZerowe, bledyRozwiazan] = **bisekcja**(f, a, b, maksIteracji, eps)

%BISEKCJA Znajdowanie miejsca zerowego metoda bisekcji

s = (a + b) / 2;

bledy = zeros(1, maksIteracji);

bledy(1) = abs(f(s)); %Poniewaz juz na poczatku moze byc ponizej eps

i = 1;

while (abs(f(s)) > eps) && (i <= maksIteracji)

s = (a + b) / 2;

if f(a)\*f(s) < 0

b = s;

else

a = s;

end

bledy(i) = abs(f(s));

i = i+1;

end

miejsceZerowe = s;

bledyRozwiazan = bledy(1, 1:i-1);

end

function [miejsceZerowe , bledyRozwiazan] = **polaczoneMetody**(f, df, a, b, maksIteracji, eps, alfa)

%POLACZONEMETODY Znajdowanie miejsca zerowego z wykozystaniem polaczenia

%metod bisekcji i Newtona Raphsona

s = (a + b) / 2;

bledy = zeros(1, maksIteracji);

bledy(1) = abs(f(s)); %Poniewaz juz na poczatku moze byc ponizej eps

i = 1;

%Metoda bisekcji

while (abs(f(s)) > eps) && (i <= maksIteracji) && (abs(f(s)) > alfa)

s = (a + b) / 2;

if f(a)\*f(s) < 0

b = s;

else

a = s;

end

bledy(i) = abs(f(s));

i = i+1;

end

miejsceZerowe = s;

%Metoda NR

while (abs(f(s)) > eps) && (i <= maksIteracji)

miejsceZerowe = miejsceZerowe - f(miejsceZerowe)/df(miejsceZerowe);

bledy(i) = (abs(f(miejsceZerowe)));

i = i+1;

end

bledyRozwiazan = bledy(1, 1:i-1);

end

**%% Zadanie przykladowe A**

clf;

clc

% Szukana funkcja: 1,1x^2+0,765x = 3,55x^3+0.74 x nalezy do <-1, 1>

% Przeksztalcenie: 3,55x^3 - 1,1x^2 - 0,765x + 0,74 = 0

% f'(x): 10.65 x^2 - 2.2 x - 0.765

% f''(x): 21.3 x - 2.2

f = @(x) (3.55\*x^3 - 1.1\*x^2 - 0.765\*x + 0.74);

df = @(x) (10.65\*x^2 - 2.2\*x - 0.765);

ddf = @(x) (21.3\*x - 2.2);

%Przebieg funkcji w podanej dziedzinie

x = linspace(-1, 1, 2000);

y = arrayfun(f, x);

subplot(2, 1, 1);

plot(x, y, "-", "DisplayName", "f(x) = 3.55\*x^3 - 1.1\*x^2 - 0.765\*x + 0.74");

grid on;

hold on;

xlabel("x");

ylabel("f(x)");

legend("Location", "best");

[x0NR, bledyNR] = NewtonRaphson1(f, df, 0, 100, 10^(-8)); % Wynik jest zly bo x0 jest po zlejs stronie zbocza

x0NR;

[x0NR, bledyNR] = NewtonRaphson1(f, df, -1, 100, 10^(-8)); %Teraz jest dobrze

x0NR

[x0Bi, blledyBi] = bisekcja(f, -1, 1, 10000, 10^(-8));

plot(x0NR, f(x0NR), "o", "DisplayName", "Wynik metoda Newtona Raphsona");

plot(x0Bi, f(x0Bi), "o", "DisplayName", "Wynik metoda bisekcji");

%Wykres bledow

subplot(2, 1, 2);

hold on;

grid on;

semilogy(bledyNR, "DisplayName", "Błędy metody Newtona Raphsona");

semilogy(blledyBi, "DisplayName", "Błędy metody bisekcji");

legend("Location", "best");

**%% Testy NewtonRaphson1**

clc;

f = @(x) (4\*x^3 + 1\*x^2 + 2\*x + 8);

df = @(x) (12\*x^2 + 2\*x + 2);

[x0, b] = NewtonRaphson1(f, df, 1, 100, 10^(-8))

**%% Testy bisekcji**

clc;

f = @(x) (4\*x^3 + 1\*x^2 + 2\*x + 8);

[x0, b] = bisekcja(f, -2, 2, 5, 10^(-8))

**%% Test połączenia**

clc;

f = @(x) (4\*x^3 + 1\*x^2 + 2\*x + 8);

df = @(x) (12\*x^2 + 2\*x + 2);

[x0, b] = polaczoneMetody(f, df, -2, 2, 10000, 10^(-8), 0.001)

x0