

Laboratorium

Ćwiczenie1:

Na podstawie kodów Matlaba utwórz trzy skrypty funkcyjne w SCILABIE wykorzystujące:

- 1) metodę bisekcji,
- 2) metodę siecznych,
- 3) metodę stycznych (Newton-Raphson)

Ćwiczenie 2:

Używając skryptów funkcyjnych utworzonych w punkcie 2 znajdź wszystkie miejsca zerowe funkcji:

$$f(x) = x^{1.5} \sin^3(x) - 5(x - 50)$$

w przedziale $[40,60]$. Przyjmij dokładność szukanego rozwiązania na poziomie 10^{-8} .

Ćwiczenie 3:

Trajektorię lotu piłki wyrzuconej przez zawodnika można zamodelować parabolą opisaną równaniem:

$$y = x \tan \theta - \frac{g}{2v_0^2 \cos^2 \theta} x^2 + y_0$$

Znajdź odpowiedni kąt rzutu θ wiedząc, że prędkość początkowa $v_0 = 20 \text{ m/s}$, a dystans do drugiego zawodnika wynosi 40 m (y_0 jest współrzędną początkowej wysokości piłki).

Przyspieszenie ziemskie przyjmij na poziomie $g = 9,81 \text{ m/s}^2$. Piłka opuszcza rękę pierwszego gracza na wysokości 1,8 m, a drugi gracz łapie ją na wysokości 1 m.

Znajdź maksymalną wysokość lotu piłki.

Sporządź wykres prezentujący trajektorię lotu piłki.

Kody Matlaba

BISEKCJA

```
function xm = demoBisect(fun,xl,xp,n)

% demoBisect  Znajduje zero funkcji u>ywaj;c metody bisekcji
%
% Dane wejsciowe:  xl,xp - lewa i prawa granica obszaru w ktorym
%                  poszukiwany jest pierwiastek,
```

```

%                               n - (opcjonalnie) liczba iteracji;
%                               domyslnie:  n = 15.
%
% Dane wyjsciowe:      x - przyblizone polozenie pierwiastka.
if nargin<4, n=15; end    % Ustawienie domyslnej liczby iteracji
a = x1;    b = xp;
fa =feval(fun,a)% a - a^(1/3) - 2;      % Wartosci poczatkowe f(a) i f(b)
fb =feval(fun,b)% b - b^(1/3) - 2;
fprintf('  k          a          xmid          b          f(xmid)\n');
for k=1:n
xm = a + 0.5*(b-a);      % Poprawne obliczenie srodka przedzialu
fm = feval(fun,xm);      % f(x) w srodku przedzialu
fprintf('%3d  %12.8f  %12.8f  %12.8f  %12.3e\n',k,a,xm,b,fm);
if sign(fm)==sign(fa)     % Zero lezy w przedziale [xm,b], zamiana a
a = xm;
fa = fm;
else                      % Zero lezy w przedziale [a,xm], zamiana b
b = xm;
fb = fm;
end
end
end

```

wywołanie funkcji

```

clc
funkcja = @fun
demoBisect(funkcja,1.5,2.0)

```

Siecznych

```

function y = fun(x)
y = .25 * x^2 - sin(x);

```

Skrypt 2:

```

function y = secant ( f, x0, x1, TOL, Nmax )
%SECANT      uzywa metody siecznych do znalezienia zera rownania nieliniowego
%
%      wywolanie funkcji:

```

```

%          y = secant ( 'f', x0, x1, TOL, Nmax )
%          secant ( 'f', x0, x1, TOL, Nmax )
%
%   dane wejsciowe:
%           f          nazwa pliku m-file definiujacego funkcje ktorej
%                       zero jest poszukiwane
%           x0,x1      poczatkowe przyblizenia polozenia zera funkcji
%           TOL        dokladnosc z jaka ma byc wyznaczone zero
%           NMax       maksymalna liczba iteracji
%
%   dane wyjsciowe:
%           y          przyblizone polozenie zera funkcji
%
%   UWAGA:
%           jezeli SECANT jest wywolane bez argumentu wyjsciowego, to
%           wyswietlany bedzie numer iteracji, aktualny przedzial poszukiwan
%           oraz przyblizona wartosc zera funkcji
%
older = x0;    old = x1;
folder = feval(f,older);
for i = 2 : Nmax
    fold = feval(f,old);
    dx = fold * ( old - older ) / ( fold - folder );
    new = old - dx;
    if ( nargin == 0 )
        disp ( sprintf ( '\t\t %3d \t %.15f \n', i, new ) )
    end
    if ( abs(dx) < TOL )
        if ( nargin == 1 )
            y = new;
        end
        return
    else
        older = old;
        old = new;
        folder = fold;
    end
end

```

```

end

disp('Osiagnieta maksymalna liczba iteracji')

if ( nargout == 1 ) y = new; end

```

wywołanie

```

clc

secant('fun',1.5,2.0,1e-14,50)

y = secant('fun',1.5,2.0,1e-14,50)

```

NEWTON

```

function y = newton ( f, x0, TOL, Nmax )

%NEWTON      uzywa metody Newtona do znalezienia zera rownania nieliniowego
%
%      wywołanie funkcji:
%
%          y = newton ( 'f', x0, TOL, Nmax )
%
%          newton ( 'f', x0, TOL, Nmax )
%
%
%      dane wejsciowe:
%
%          f      nazwa pliku m-file definiujacego funkcje ktorej zero jest
%
%                  poszukiwane oraz jej pierwsza pochodna
%
%          x0      poczatkowe przyblizenie polozenia zera funkcji
%
%          TOL      dokladnosc z jaka ma byc wyznaczone zero
%
%          NMax     maksymalna liczba iteracji
%
%
%      dane wyjsciowe:
%
%          y      przyblizone polozenie zera funkcji
%
%
%      UWAGA:
%
%          jezeli NEWTON jest wywolane bez argumentu wyjsciowego, to
%
%          wyswietlany bedzie numer iteracji, aktualny przedzial poszukiwan
%
%          oraz przyblizona wartosc zera funkcji
%
%
for i = 1 : Nmax
[fold, fpold] = feval(f,x0);

dx = fold / fpold;

x0 = x0 - dx;

if ( nargout == 0 )

```

```
disp ( sprintf ( ' (p. Newtona)  \t\t %3d \t %.10f ', i, x0 ) )
end
if ( abs(dx) < TOL )
if ( nargout == 1 )
y = x0;
end
return
end
end
disp('Osiagnieta maksymalna liczba iteracji')
if ( nargout == 1 ) y = x0; end
```