Sprawozdanie MNUM
Projekt nr.3
Zadanie 3.32

Kajetan Kaczmarek

4 maja 2018

1. Opis zastosowanych algorytmów :

   (a) W pierwszym zadaniu, tj. znalezienie zer dla funkcji

   $$f(x) = 0.5xcos(x) - ln(x)$$

   użyłem dwóch metod. Założeniami dla obywdu metod była a) ciągłość, co jest oczywiste dla ww. funkcji, oraz b) różne znaki na krańcach przedziału, do czego odnoszę się poniżej. Zastosowane metody :

   - Metoda bisekcji
     W metodzie bisekcji na początek liczony jest punkt wypadający pomiędzy podanymi wejściowymi punktami, tj. $x = \frac{a+b}{2}$ dla p. wejściowych a i b. Następnie sprawdzamy czy punkt ten jest naszym zerem z podaną dokładnością eps, czyli czy $|f(x)| < eps$. Jeśli tak jest kończymy wykonywanie algorytmu, jeśli nie to sprawdzamy warunek $f(a)f(b) < 0$ i w zależności od wyniku zastępujemy lewy lub prawy koniec przedziału w którym szukamy wyliczonym x, tak, aby krańce przedziału nadal miały przeciwne znaki. Alternatywnym warunkiem wyjściowym z pętli jest $|a - b| < eps$, czyli zbliżenie się do siebie punktów a i b tak, że dalsze obliczenia są niemożliwe.
   - Metoda Siecznych
     Metoda siecznych jest podobna do metody bisekcji - szukamy zer przez zawężanie zakresu poszukiwań, warunki końcowe są więc takie same.Różny jest jednak algorytm wyznaczania kolejnego punktu : tutaj kolejne punkty wyznaczamy ze wzoru

     $$x_i = x_{i-1} - f(x_{i-1})\frac{x_{i-1} - x_{i-2}}{f(x_{i-1}) - f(x_{i-2})}$$

Tak że łączenie kolejnych punktów daje nam sieczne naszej funkcji f(x) i przybliża jej zera.

- Uwaga techniczna - założeniami obydwu metod są różne znaki funkcji na krańcach przedziału. Jako że warunek ten nie jest spełniony dla zadanego przedziału w mojej funkcji, a do tego ww. metody znajdują tylko jedno zero, podzieliłem zadany przedział [2,11] na dwa mniejsze , tj. [2,7] i [7,11] tak aby w każdym znajdowało się jedno zero, i aby spełniały one założenia metod.

(b) W drugim zadaniu, tj. znalezienie zer wielomianu $f(x) = 2x^4 + 5x^3 - 2x^2 + 3x^3 + 7$ zastosowałem metody :

- Metoda Newtona Metoda Newtona, zwana inaczej metodą stycznych , opiera się na wykorzystaniu iteracjnego wzoru

$$x_i = x_{i-1} - \frac{f(x)}{f(x)'}$$

- Metoda Mullera MM2

2. Kod moich programów

- Funkcja main dla pierwszego zadania

```
imax = 50;

xb1 = bisect(11,7,10e-5,imax); %Pierwsze zero funkcji metoda bisekcji
xb2 = bisect(7,2,10e-5,imax); %Drugie zero funkcji metoda bisekcji

xs1 = secants(11,7,10e-5,imax); %Pierwsze zero funkcji metoda
    siecznych
xs2 = secants(7,2,10e-5,imax); %Drugie zero funkcji metoda siecznych

Ans = [xb1(imax - 1),xb2(imax - 1),xs1(imax - 1),xs2(imax - 1)]; %
    Wektor z rozwiazaniami dla ulatwienia rysowania

X = 2:0.1:11; % Nasz przedzial z probkowaniem co 0.1
Y = arrayfun(@(x) fzad(x),X); %Wyliczenie wartosci naszej funkcji do
    narysowania wykresu

figure; % Ponizszy kod rysuje wykres
plot(X,Y);% Rysowanie wykresu funkcji
hold on;
plot(Ans,0, '-o','MarkerEdgeColor','b','MarkerFaceColor'
    ,[0.5,0.5,0.5]); % Rysowanie miejsc zerowych
hold off;
```

- Pomocnicza funkcja licząca wartości naszej funkcji podanej dla zadania

```
function f = fzad(x) % Funkcja pomocnicza obliczajaca wartosci funkcji
    podanej w zadaniu
f = 0.5*x*cos(x) - log(x);
```

- Funkcja licząca zera funkcji metodą bisekcji

```
function [x]=bisect(a,b,eps, imax) % Funkcja realizujaca metode
    bisekcji

x = zeros(imax - 1 ,1);
```

```
i = 1;

while abs( a - b ) > eps && i < imax% Warunek koncowy - punkty pomiedzy ktorymi liczymy zblizyly sie na odleglosc abs
    x0 = ( a + b ) / 2 ; % Policzenie punktu pomiedzy p. wejsciowymi
    if( abs( fzad(x0) ) < eps) % Warunek koncowy - policzenie zera z dokladnoscia eps
        break;
    else
        if(fzad(x0) * fzad(a) <0) %W zaleznosci od tego z ktorej strony wypadl punkt x podmieniamy wartosc a lub b na x
            b = x0;
        else
            a = x0;
        end
    end
    x(i) = x0;
    i = i + 1;
end

x(imax -1) = x0;
```

- Funkcja licząca zera funkcji metodą siecznych

```
function x = secants(x1,x2,eps,imax) % Funkcja realizujaca metode siecznych

if(abs(fzad(x2)-fzad(x1))<eps) % Sprawdzenie czy punkty wejsciowe nie leza zbyt blisko
    error("Zle punkty wyjsciowe");
    return;
end

i = 1;
x = zeros(imax - 1 ,1);
f1 = fzad(x1); % Policzenie wartosci funkcji w p. wejsciowych
f2 = fzad(x2);

if(f1<f2) % Zamienienie miejscami punktow jesli wartosc w x1 jest mniejsza niz w x2
    a = x1;
    x1= x2;
    x2 = a;
    f1 = fzad(x1);
    f2 = fzad(x2);
end

while abs(x2-x1)>eps && i<imax% Warunek koncowy - jesli punkty pomiedzy ktorymi liczymy zbliza sie do odleglosci eps
    x0 = x1 - (f1*(x1-x2))/(f1-f2); % Obliczenie nastepnego punkty
    f = fzad(x0);

    if(abs(f) < eps) % Warunek koncowy -  znalezienie zera z dokladnoscia do eps
        break;
    end
    x(i) = x0;
    i = i + 1;
    x2 = x1; % Zmiana na punkty dla nastepnej iteracji
    f2 = f1;
    x1 = x0;
    f1 = f;
end
x(imax -1) = x0;
```

- Funkcja main dla drugiego zadania

```
% Dane wej ciowe
A = [2  -2 3 7]; % Macierz wspolczynnikow wielominau
X = [-4 : 0.1 :0 ]; % Przedzia  X na kt rym nasz wielomian ma zera rzeczywiste
Y = polyval(A,X) ; % Warto ci naszego wielomianu w przedziale X

X0 = newton(X,Y);
Y0 = polyval(X0, X);

figure
plot(X,Y0);
```

- Funkcja licząca zera funkcji metodą Newtona

```
function X = newton(eps,x0, imax)

X = zeros(imax -1 , 1);
i = 1;

% Dane wejciowe
A = [2 5 -2 3 7]; % Macierz wspolczynnikow wielominau
B = [8 15 -4 3]; % Macierz wspolczynnikow pochodnej wielomianu A

for i = 1: imax - 1%while abs( polyval(A,x0)) > eps && i< imax
    %polyval(A,x0 )
    %disp(x0 )
    x0 = x0 - (polyval(A,x0 )/polyval(B,x0 ));
    X(i) = x0;
    i = i+1;
end
X(imax-1) = x0;
```

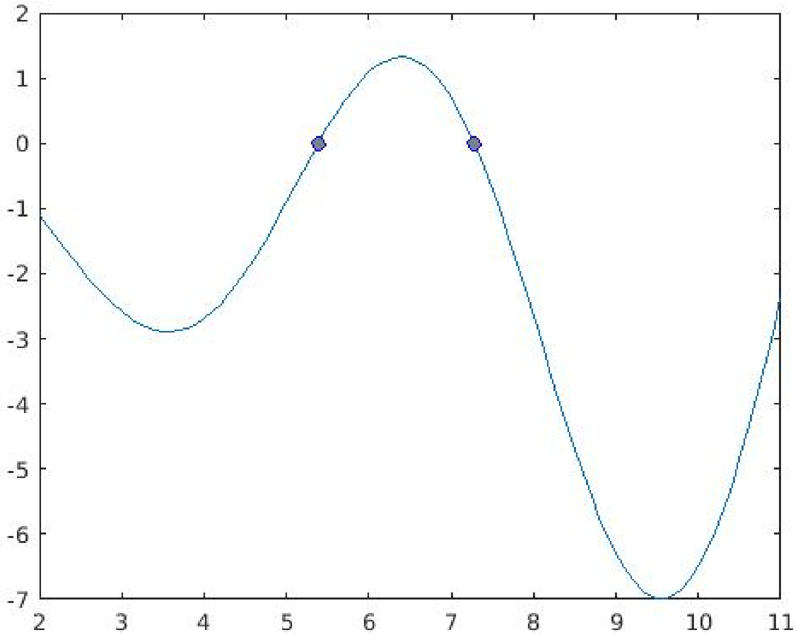
- Funkcja licząca zera funkcji metodą Mullera

```
function [x] = muller(eps, x0, imax)

% Dane wejciowe
A = [2 5 -2 3 7]; % Macierz wspolczynnikow wielominau
B = [8 15 -4 3]; % Macierz wspolczynnikow pochodnej wielomianu A
C = [24 30 -4] ; % Macierz wspolczynnikow pochodnej drugiego stopnia wielomianu A

x = zeros(imax - 1,1);
i = 1;

while abs(polyval(A,x0) ) > eps && i < imax

    sqr = sqrt(polyval(B,x0)^2 - 2*polyval(A,x0)*polyval(C,x0));

    zpl = -2*polyval(A,x0)/(polyval(B,x0) + sqr );
    zmin = -2*polyval(A,x0)/(polyval(B,x0) - sqr );

    if abs((polyval(B,x0) - sqr ))>abs((polyval(B,x0) + sqr ))
        x0 = x0 + zmin;
    else
        x0 = x0 + zpl;
    end

    x(i)=x0;

    i= i +1;
end
x(imax - 1) = x0;
```

3. Wyniki :

- Dla zadania pierwszego obydwie metody zwróciły zbliżone wyniki, tj.

Metoda	Zero nr.1	Zero nr. 2
Metoda Bisekcji	7.27703857421875	5.38775634765625
Metoda Siecznych	7.27702154631274	5.38773923503257



4. Wnioski :