Sprawozdanie MNUM
Projekt nr.3
Zadanie 3.32

Kajetan Kaczmarek

7 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 $f(a)f(b) <$ 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. W ten sposób kolejne punkty są łączone prostą a jej przecięcie z osią x wyznacza kolejny punkt

- 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

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

  , który wynika z rozwinięcia funkcji w szereg Taylora, czyli

$$f(x) = \ f(x_n) + f'(x_n)(x - x_n)$$

  . Metoda Newtona jest lokalnie zbieżna, dla punktów zbytnio oddalonych od p. zerowego może nie dawać rezultatów.

- Metoda Mullera MM2
  Ogólną ideą metod Mullera jest przybliżanie lokalnie funkcji w okolicy zera przez funkcję kwadratową. Jest pewnym uogólnieniem metody siecznych, z dodaną liniową interpolacją pomiędzy kolejnymi punktami. Werjsa MM2 algorytmu, użyta w moim rozwiązaniu, zakłada użycie wartości wielomianu i jego pierwszej oraz drugiej pochodnej w danym punkcie.Kolejne punkty wyliczamy z wzoru

$$x_{k+1} = x_k + z_{min},$$

  gdzie z_{min} jest jedną z wartości

$$z_{+/-} = \frac{-2f(x_k)}{f'(x_k) + / - \sqrt{(f'(x_k))^2 - 2f(x_k)f''(x_k)}}$$

  w zależności od modułu mianownika - wybieramy tę z większym modułem

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
hax = axes;
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 on;
line([11 11],get(hax,'YLim'),'Color',[1 0 0])
line([7 7],get(hax,'YLim'),'Color',[1 0 0])
line([7 7],get(hax,'YLim'),'Color',[0 1 0])
line([2 2],get(hax,'YLim'),'Color',[0 1 0])

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,Y0] = newton(X,Y);

figure
plot(X,Y0);
```

- Funkcja licząca zera funkcji metodą Newtona

```
function [X ,y]= newton(eps,x0, imax)

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

i = 1;

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

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

- Funkcja licząca zera funkcji metodą Mullera

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

% Dane wej ciowe
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);
```

```
y = 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;
    y(i)= polyval(A,x0);
    i= i +1;
end
```

- Funkcja licząca zera funkcji metodą Newtona z uwzględnieniem deflacji

```
function [x,y] = newtonDef(eps,x0, imax)% Modyfikacja metody Newtona z uwzgl. deflacji liniowej

i = 1;
% Dane wej ciowe
A = [2 5 -2 3 7]; % Macierz wspolczynnikow wielominau
B = polyder(A); % Macierz wspolczynnikow pochodnej wielomianu A

r = size(A) - 1;
r = r(2);
xm = x0;

x = zeros(r , 1);
y = zeros(r - 1,1);

for j = 1:r
    while abs( polyval(A,x0)) > eps && i< imax
        x0 = x0 - (polyval(A,x0 )/polyval(B,x0 ));
        i = i+1;
    end
    A = deflation(A,x0);
    B = polyder(A);
    i = 1;
    x(j) = x0;
    y(j)= polyval(A,x0);
    x0 = xm;
end
```

- Funkcja licząca zera funkcji metodą Mullera z uwzględnieniem deflacji

```
function [x,y] = mullerDef(eps, x0, imax) % modyfikacja alg. Mullera z uwzglednieniem deflacji liniowej

% Dane wej ciowe
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

r = size(A) - 1;
r = r(2);
xm = x0;

x = zeros(r - 1,1);
y = zeros(r - 1,1);
i = 1;
for j = 1:r
    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 ))
```

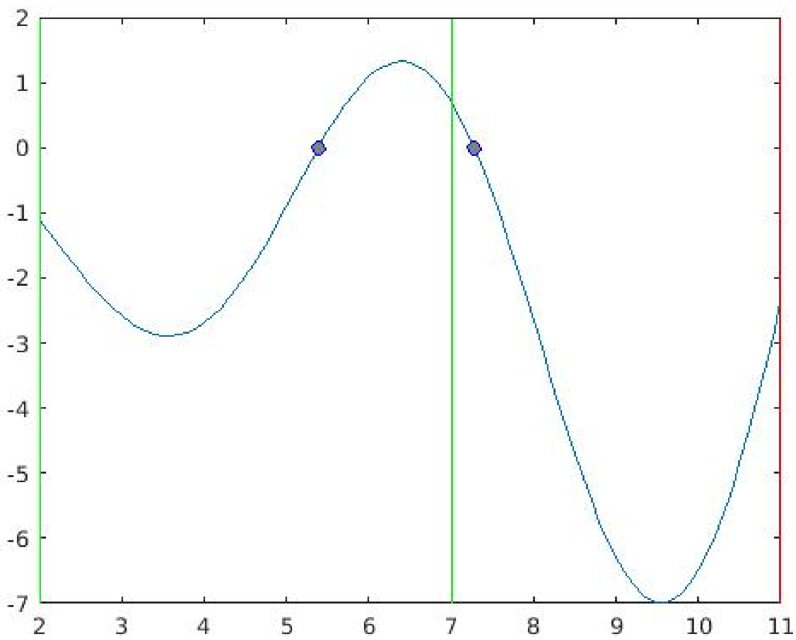
```
            x0 = x0 + zmin;
        else
            x0 = x0 + zpl;
        end

        i= i +1;
    end
    A = deflation(A,x0);
    B = polyder(A);
    C = polyder(B);
    i = 1;
    x(j) = x0;
    y(j)= polyval(A,x0);
    x0 = xm;
end
```

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 :

   - Dla metod z pierwszego punktu :
     Metoda bisekcji zbiega liniowo, a jej błąd jest związany jedynie z ilością iteracji.Z kolei metoda siecznych zbiega szybciej, jednak dla zbyt dużego przedziału [a,b] może nie osiągnąć wyniku,

szczególnie jeżeli występuje punkt gdzie $f'(x) = 0$.Dla podanaje funkcji obydwie metody dały satysfakcjonujące rezultaty

- Dla metod z drugiego punktu : Obydwie metody okazały się dość skuteczne i dla podanych wartości startowych skutecznie znajdowały wartości miejsc zerowych. Dla wybranego przeze mnie przedziału Metoda MM2 była skuteczniejsza i znajdowała pierwiastki w mniejszej liczbie iteracji. Pozwoliła ona również znaleźć wartości pierwiastków urojonych. Jako że były one swoimi sprzężeniami, metoda znalazła tylko jeden z nich.

Iteracja	Metoda Bisekcji 1 przedział		Metoda Bisekcji 2 przedział		Metoda Siecznych 1 przedział		Metoda Siecznych 2 przedział	
Dane	X	Y	X	Y	X	Y	X	Y
1	9	-6.2973	4.5000	-1.9784	7.9037	-2.2637	5.0779	-0.7175
2	8	-2.6614	5.7500	0.7267	7.2118	0.1841	6.0558	1.1489
3	7.5000	-0.7150	5.1250	-0.6066	7.2638	0.0383	5.4538	0.1452
4	7.2500	0.0777	5.4375	0.1098	7.2775	-0.0013	5.3667	-0.0471
5	7.3750	-0.2986	5.2812	-0.2417			5.3880	0.0006
6	7.3125	-0.1051	5.3594	-0.0636				
7	7.2812	-0.0123	5.3984	0.0238				
8	7.2656	0.0330	5.3789	-0.0197				
9	7.2734	0.0104	5.3887	0.0021				
10	7.2773	-0.0009	5.3838	-0.0088				

Tablica 1: Wyniki metody Bisekcji i Siecznych

X_0	-6		-5.500		-5	
Dane	X	Y	X	Y	X	Y
1	-4.769	439.495	-4.411	282.809	-4.058	170.154
2	-3.898	130.539	-3.655	82.062	-3.424	47.484
3	-3.324	35.443	-3.179	20.895	-3.055	10.815
4	-3.007	7.449	-2.946	3.630	-2.907	1.351
5	-2.895	0.737	-2.885	0.211	-2.882	0.033
6	-2.882	0.010	-2.881	0.001	-2.881	0
7	-2.881	0	-2.881	0	-2.881	0
8						
9						

Tablica 2: Wyniki metody Newtona cz. 1

X_0	-4.500		-4		-3.500	
Dane	X	Y	X	Y	X	Y
1	-3.715	92.767	-3.387	42.867	-3.094	13.761
2	-3.214	24.080	-3.037	9.510	-2.918	1.963
3	-2.960	4.430	-2.902	1.100	-2.883	0.068
4	-2.887	0.301	-2.882	0.022	-2.881	0
5	-2.881	0.002	-2.881	0	-2.881	0
6	-2.881	0				
7						
8						
9						

Tablica 3: Wyniki metody Newtona cz. 2

X_0	-3		-2.500		-2	
Dane	X	Y	X	Y	X	Y
1	-2.894	0.664	-3.212	23.950	0.143	7.403
2	-2.882	0.008	-2.959	4.397	-2.541	-12.183
3	-2.881	0	-2.887	0.297	-3.114	15.368
4			-2.881	0.002	-2.924	2.319
5			-2.881	0	-2.883	0.093
6					-2.881	0
7					-2.881	0
8						
9						

Tablica 4: Wyniki metody Newtona cz. 3

X_0	-1.500		-1		-0.500	
Dane	X	Y	X	Y	X	Y
1	-0.944	-0.238	-0.929	-0.027	-1.081	-2.160
2	-0.927	-0.002	-0.927	0	-0.934	-0.101
3	-0.927	0	-0.927	0	-0.927	0
4					-0.927	0
5						
6						
7						
8						
9						

Tablica 5: Wyniki metody Newtona cz. 4

X_0	0		0.500		1	
Dane	X	Y	X	Y	X	Y
1	-2.333	-15.123	-1.022	-1.307	0.318	7.934
2	-4.316	248.641	-0.930	-0.043	-1.946	-14.579
3	-3.591	71.539	-0.927	0	-0.256	6.026
4	-3.144	17.787	-0.927	0	-1.493	-8.632
5	-2.934	2.878			-0.946	-0.255
6	-2.884	0.138			-0.927	-0.002
7	-2.881	0			-0.927	0
8	-2.881	0				
9						

Tablica 6: Wyniki metody Newtona cz. 5

X_0	1.500		2		2.500	
Dane	X	Y	X	Y	X	Y
1	0.911	13.236	1.353	26.481	1.753	51.905
2	0.170	7.479	0.763	11.018	1.143	18.696
3	-2.507	-12.868	-0.139	6.532	0.519	8.863
4	-3.193	22.161	-1.847	-13.590	-0.937	-0.141
5	-2.952	3.945	-0.629	3.393	-0.927	-0.001
6	-2.886	0.245	-0.988	-0.827	-0.927	0
7	-2.881	0.001	-0.928	-0.019		
8	-2.881	0	-0.927	0		
9			-0.927	0		

Tablica 7: Wyniki metody Newtona cz. 6

X_0	-6		-5.500		-5	
Dane	X	Y	X	Y	X	Y
1	-4.293+1.135i	44.904-338.682i	-3.970+0.996i	22.933-217.825i	-3.650+0.849i	7.994-130.893i
2	-3.562+0.027i	66.899-4.224i	-3.355+0.033i	38.931-3.854i	-3.167+0.039i	19.663-3.460i
3	-2.790-0.228i	-6.902+9.575i	-2.819-0.022i	-3.100+1.006i	-2.872-0.004i	-0.466+0.228i
4	-2.877-0.002i	-0.231+0.096i	-2.881+	0.003-0.004i	-2.881+	0
5	-2.881+	0	-2.881-	0	-2.881+	0
6						
7						
8						
9						

Tablica 8: Wyniki metody Mullera cz. 1

X_0	-4.500		-4		-3.500	
Dane	X	Y	X	Y	X	Y
1	-3.335+0.688i	-1.361-71.042i	-3.027+0.495i	-6.408-32.008i	-2.731+0.182i	-8.234-6.658i
2	-3.010+0.043i	7.535-2.891i	-2.905+0.032i	1.217-1.735i	-2.878-0.002i	-0.195+0.109i
3	-2.881-0.001i	-0.026+0.040i	-2.881	0.001	-2.881	0
4	-2.881	-	-2.881	0		
5						
6						
7						
8						
9						

Tablica 9: Wyniki metody Mullera cz. 2

X_0	-3		-2.500		-2	
Dane	X	Y	X	Y	X	Y
1	-2.881	-0.032	-2.900	1.015	-1.226	-4.381
2	-2.881	0	-2.881	0	-0.934	-0.105
3			-2.881	0	-0.927	0
4						
5						
6						
7						
8						
9						

Tablica 10: Wyniki metody Mullera cz. 3

X_0	-1.500		-1		-0.500	
Dane	X	Y	X	Y	X	Y
1	-0.986	-0.810	-0.927	-0.001	-0.927	-0.011
2	-0.927	-0.001	-0.927	0	-0.927	0
3	-0.927	0				
4						
5						
6						
7						
8						
9						

Tablica 11: Wyniki metody Mullera cz. 4

X_0	0		0.500		1	
Dane	X	Y	X	Y	X	Y
1	-1.266	-5.004	0.162+0.957i	8.452-2.850i	0.560+0.637i	5.328+1.923i
2	-0.938	-0.157	0.637+0.895i	1.052+0.467i	0.647+0.953i	-0.104-0.381i
3	-0.927	0	0.654+0.940i	-0.001-0.001i	0.654+0.940i	0
4			0.654+0.940i	0	0.654+0.940i	0
5						
6						
7						
8						
9						

Tablica 12: Wyniki metody Mullera cz. 5

X_0	1.500		2		2.500	
Dane	X	Y	X	Y	X	Y
1	0.892+0.588i	5.893+7.559i	1.217+0.633i	7.791+18.269i	1.542+0.717i	11.503+37.932i
2	0.654+0.907i	0.572+0.568i	0.743+0.910i	-0.895+2.300i	0.865+0.980i	-4.859+4.415i
3	0.654+0.940i		0.654+0.940i	-0.010+0.004i	0.652+0.945i	-0.066-0.122i
4	0.654+0.940i		0.654+0.940i		0.654+0.940i	
5						
6						
7						
8						
9						

Tablica 13: Wyniki metody Mullera cz. 6