Dokumentacja projektu laboratoryjnego numer 3 przedmiot MNUM

Kamil Foryszewski

16 maja 2016

Spis treści

1 Zadanie 1

1.1 Polecenie

Proszę znaleźć wszystkie zera funkcji

$$f(x) = 1.4 * sin(x) - e^x = 6 * x - 0.5$$

w przedziale $[-5, 5]$, używając dla każdego zera programu z implementacją
a) metody bisekcji
b) metody siecznych
c) metody Newtona

1.2 Metoda bisekcji

1.2.1 Opis teoretyczny

Teoretyczny zarys metody bisekcji możemy przybliżyć poniższym algorytmem:

1. Począwszy od przedziału startowego $[a, b] = [a_0, b_0]$ obliczamy środek przedziału c_n

$$c_n = \frac{a_n + b_n}{2}$$

i obliczamy wartość $f(x)$ w tym punkcie.

2. Obliczamy iloczyny $f(a_n) * f(c_n)$ oraz $f(b_n) * f(c_n)$ i jako nowy przedział $[a_{n=1}, b_{n+1}]$ wybieramy argumenty tego iloczynu którego wartość jest ujemna.

Kroki te powtarzamy aż do momentu uzyskania $f(c_n) < \delta$ gdzie δ to oczekiwana dokładność rozwiązania. W przypadku "płaskich"funkcji warto też kontrolować długość rozpatrywanego przedziału. Dokładność wyniku zależy jedynie od ilości iteracji dlatego metoda jest zbieżna liniowo z ilorazem zbieżności 0.5, co czyni ją stosunkowo wolno zbieżną w przypadku wyboru szerokiego przedziału początkowego.

1.2.2 Realizacja w programie Matlab

```
%funkcja wyznaczajaca zera funkcji metoda bisekcji
function bzeropoint = bisection(fun,l,r,iter)
%Dane wejsciowe:        l,r - lewa i prawa sterona przedzialu poszukiwan
%                       fun - funkcja
%                       iter - maksymalna liczba uteracji
%Dane wyjsciowe: zerospoint - wyznaczone miejsce zerowe
  a = l;
  b = r;
  fa =feval(fun,a);      %  Wartosci poczatkowe f(a) i f(b)
  fb =feval(fun,b);
  for k=1:iter
    xm = a + 0.5*(b-a);     %  Poprawne obliczenie srodka przedzialu
    fm = feval(fun,xm);       %  f(x) w srodku przedzialu
    fprintf('%3d %12.16f %12.16f %12.16f %12.3e\n',k,a,xm,b,fm);
    if(fm == 0)
        return
    end
    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
  bzeropoint = xm;
  return
end
```

1.3 Metoda siecznych

1.3.1 Opis teoretyczny

Teoretyczny zarys metody siecznych możemy przybliżyć poniższym algorytmem:

1. Począwszy od przedziału startowego $[a, b] = [a_0, b_0]$ obliczamy punkt δx_n jako miejsce przecięcia siecznej funkcji przechodzącej prze punkty $[a_n, b_n]$ gdzie $\delta x_n = \frac{f(x_n)(x_n - x_{n-1})}{f(x_n) - f(x_{n-1})}$

2. Następnie nowy przedział oznaczamy $x_{n+1} = x_n - \delta x_n$

Kroki te powtarzamy aż do momentu uzyskania $f(c_n) < \delta$ gdzie δ to oczekiwana dokładność rozwiązania. Rząd zbieżności metody siecznych wynosi $(1 + sqrt(5))/2$ co jest w przybliżeniu równe 1.618. Jest więc ona dużo szybsza od metody bisekcji, jednak jest zbieżna jedynie lokalnie. Jeżeli ine zadbamy o wybór odpowiedniego przedziału początkowego może okazać się w ogóle nie zbieżna.

1.3.2 Realizacja w programie Matlab

```
%funkcja obliczajaca zera funkcji metoda siecznych
function szeropoint = secant(fun,l,r,iter)
%Dane wejsciowe:       l.r - lewa i prawa sterona przedzialu poszukiwan
%                      fun - funkcja
%                      iter - maksymalna liczba uteracji
%Dane wyjsciowe: zerospoint - wwyznaczone miejsce zerowe
  a = l;
  b = r;
  fa = feval(fun,a); %wartosc funkcji w punkcie start.
  for k = 1:iter
    fb = feval(fun,b);
    dx = fb * (b-a) / (fb-fa); %wyznaczenie przeciecia sieczna
    xm = b-dx; %zawezenie przedzialu
    if(isnan(xm))
        return
    end
    a = b;
    b = xm;
    fa = fb;
    szeropoint = b;
    fprintf('%3d %12.16f %12.16f %12.16f %12.3e\n',k,a,xm,b,dx);
    if(fb == 0) %dodatkowy warunek zakonczenia wykonywania
        return
    end
  end
end
```

1.4 Metoda Newtona

1.4.1 Opis teoretyczny

Metoda Newtona polega na wyznaczeniu częściowego (uciętego) rozwinięcia w szereg Taylora danej funkcji, które możemy traktować jak liniowe przybliżenie funkcji według wzoru:

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

Następnie wyznaczamy kolejne punkty iteracji poprzez przyrównanie do zera otrzymanej aproksymacji:

$$f(x_n) + f'(x_n)(x_{n+1} - x_n) = 0$$

Prowadzi to do zależności iteracyjnej danej następującym wzorem:

$$x_{n+1} = x_n - \frac{f(x_n)}{f'(x_n)}$$

Metoda Newtona jest zbieżna jedynie lokalnie, ponieważ wyznaczając styczną do wykresu w danym punkcie możemy w przypadku ujemnego znaku pochodnej dojść do rozbieżności. Dla przypadków pochodnej większej od zera metoda jest zbieżna kwadratowo. Rząd zbieżności wynosi 2.

1.4.2 Realizacja w programie Matlab

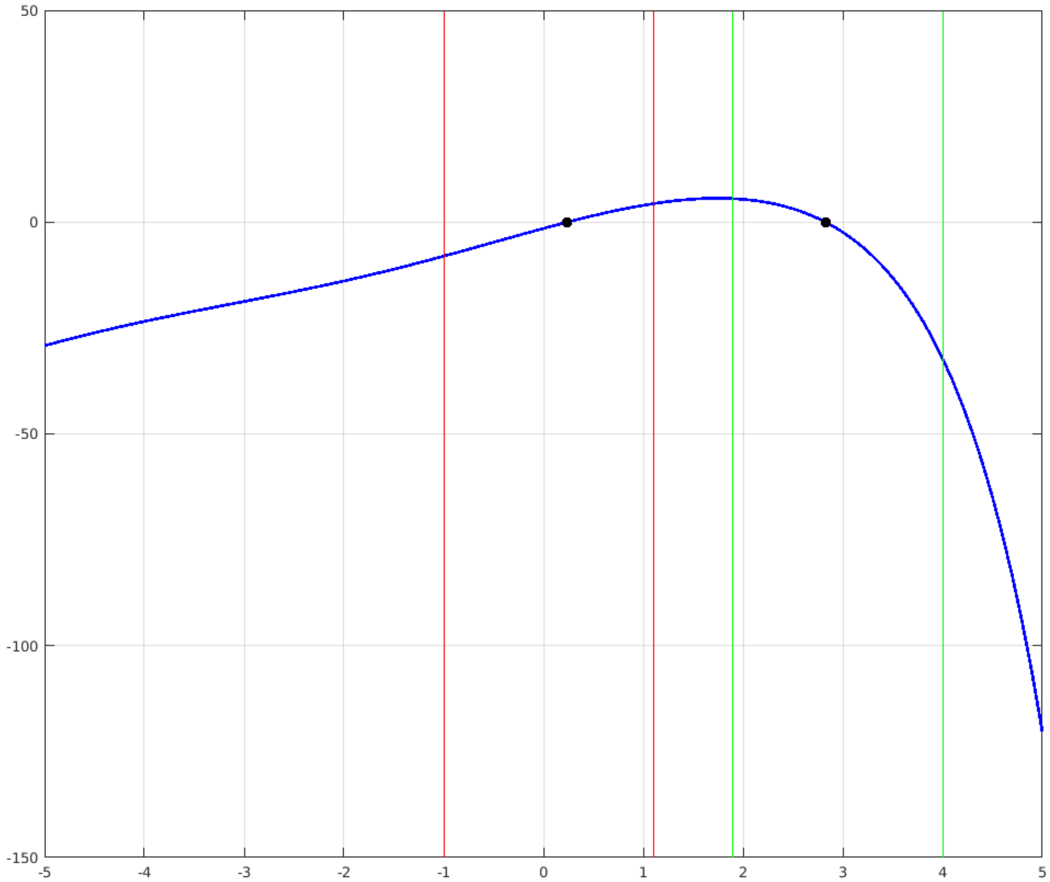
```
%funkcja obliczajaca zera funkcji metoda Newtona
function nzeropoint = newton(fun,l,iter)
%Dane wejsciowe:        l prawa sterona przedzialu poszukiwan
%                       fun - funkcja
%                       iter - maksymalna liczba uteracji
%Dane wyjsciowe: zerospoint - wyznaczone miejsce zerowe

  x0 = l;
  for k = 1:iter
    [fold, fpold] = feval(fun,x0);
    dx = fold / fpold; %wyznaczenie przyrostu funkcji
    x0 = x0 - dx;
    fprintf('%3d %12.16f %12.16f %12.3e\n',k,x0,dx,fold);
    if(fold == 0)
        return
    end
        if fold==0 %dodatkowy warunek zatrzumania
        nzeropoint = x0;
        break;
     end
  end
end
```

1.5 Analiza danych wejściowych

W celu wyznaczenia przedziałów izolacji miejsc zerowych został wykorzystany algorytm opisany w skrypcie prof. Tatjewskiego. Wstępna analiza danych rozpoczyna się od wygenerowania wykresu funkcji w danym przedziale i na tej podstawie wyboru przedziału startowego dla algorytmu. Następnie w podanym przedziale w pętli badany jest znak iloczynu funkcji w punktach granicznych. Jeżeli jest on ujemny, oznacza to występowanie miejsca zerowego w danym przedziale. Jeżeli nie, to przedział jest rozszerzany do momentu przekroczenia przedziału danego w zadaniu. Poniżej wykres funkcji z zaznaczonymi przedziałami izolacji wyznaczonymi przez algorytm.

Rysunek 1: Wykres z zaznaczonymi przedziałami startowymi



1.6 Skrypt generujący rozwiązanie zadania w programie Matlab

```
%Realizacja zadania 1
clear;
%Generowanie wykresu funkcji aby sprawdzic poprwanosc otrzymanych rozwiazan
x = -5: .1 : 5;
plot(x, fun(x), 'b','LineWidth', 2)
grid on
axis([-5 5 -150 150])

n=100;
x1 = -1;
x2 = 0;

%wyznaczanie przedzialow izolacji na podstawie wkryptu MNUM
for k=1:2
    for j=1:n
        if fun(x1)*fun(x2)<0
            a = x1;
            b = x2;
```

```
            fprintf('Wyniki dla %d miejsca zerowego w przedziale [%d,%d]\n',k,a,b);
            bisection('fun',a,b,100);
            secant('fun',a,b,100);
            newton('fun',a,100);
            x1 = 3;
            x2 = 4;
            break;
        elseif abs(fun(x1))<abs(fun(x2))
            x1 = x1+1.1*(x1-x2);
        else
            x2 = x2+1.1*(x2-x1);
        end
        if(x1>5)&&(x2<(-5))
            break; %wyjscie z petli po przekroczeniu przedzialu
        end
    end
end
```

1.7 Wyniki

Metoda siecznych pierwsze miejsce zerowe

iteracja	przedział	wynik	wartość funkcji
1	[1.1000000000;0.3637775411]	0.3637775410734196	7.362e-01
2	[0.3637775411;0.2120880091]	0.2120880090516735	1.517e-01
3	[0.2120880091;0.2402303125]	0.2402303124756739	-2.814e-02
4	[0.2402303125;0.2397497010]	0.2397497010094280	4.806e-04
5	[0.2397497010;0.2397479765]	0.2397479764875909	1.725e-06
6	[0.2397479765;0.2397479766]	0.2397479765971351	-1.095e-10
7	[0.2397479766;0.2397479766]	0.2397479765971350	3.647e-17
8	[0.2397479766;0.2397479766]	0.2397479765971350	0.000e+00

Metoda siecznych drugie miejsce zerowe

iteracja	przedział	wynik	wartość funkcji
1	[4.0000000000;2.2085621561]	2.2085621561012472	1.791e+00
2	[2.2085621561;2.4401148217]	2.4401148217298374	-2.316e-01
3	[2.4401148217;3.1268105849]	3.1268105848860621	-6.867e-01
4	[3.1268105849;2.7431506522]	2.7431506521755535	3.837e-01
5	[2.7431506522;2.8107252731]	2.8107252730503189	-6.757e-02
6	[2.8107252731;2.8280523947]	2.8280523946681111	-1.733e-02
7	[2.8280523947;2.8270291416]	2.8270291416146467	1.023e-03
8	[2.8270291416;2.8270409015]	2.8270409014519227	-1.176e-05
9	[2.8270409015;2.8270409099]	2.8270409098837272	-8.432e-09
10	[2.8270409099;2.8270409099]	2.8270409098836571	7.003e-14
11	[2.8270409099;2.8270409099]	2.8270409098836571	-0.000e+00

Metoda Newtona pierwsze miejsce zerowe			
iteracja	**przedział**	**wynik**	**wartość funkcji**
1	-1.2594323665	0.2594323664525493	-8.046e+00
2	0.0197367824	0.2396955840431176	1.195e-01
3	-0.0000523922	0.2397479762357552	-3.190e-04
4	-0.0000000004	0.2397479765971350	-2.200e-09
5	0.0000000000	0.2397479765971350	0.000e+00

Metoda Newtona drugie miejsce zerowe			
iteracja	**przedział**	**wynik**	**wartość funkcji**
1	-4.8651088967	6.7651088967106006	5.539e+00
2	0.9610395411	5.8040693556058418	-8.263e+02
3	0.9185069972	4.8855623584001924	-2.980e+02
4	0.8319658399	4.0535965184699823	-1.049e+02
5	0.6650562218	3.3885402966240363	-3.489e+01
6	0.4056676310	2.9828726656584705	-1.013e+01
7	0.1405409130	2.8423317526851513	-2.126e+00
8	0.0151272015	2.8272045512095620	-1.890e-01
9	0.0001636224	2.8270409288572882	-2.001e-03
10	0.0000000190	2.8270409098836575	-2.320e-07
11	0.0000000000	2.8270409098836571	-3.553e-15
12	-0.0000000000	2.8270409098836571	0.000e+00

Metoda bisekcji pierwsze miejsce zerowe			
iteracja	**przedział**	**wynik**	**wartość funkcji**
1	[-1.0000000000;1.1000000000]	0.0500000000000000	-1.181e+00
2	[0.0500000000;1.1000000000]	0.5750000000000001	1.934e+00
3	[0.0500000000;0.5750000000]	0.3125000000000001	4.386e-01
4	[0.0500000000;0.3125000000]	0.1812500000000000	-3.589e-01
5	[0.1812500000;0.3125000000]	0.2468750000000001	4.336e-02
6	[0.1812500000;0.2468750000]	0.2140625000000000	-1.569e-01
7	[0.2140625000;0.2468750000]	0.2304687500000001	-5.657e-02
8	[0.2304687500;0.2468750000]	0.2386718750000001	-6.553e-03
9	[0.2386718750;0.2468750000]	0.2427734375000001	1.841e-02
10	[0.2386718750;0.2427734375]	0.2407226562500001	5.934e-03
11	[0.2386718750;0.2407226563]	0.2396972656250000	-3.088e-04
12	[0.2396972656;0.2407226563]	0.2402099609375000	2.813e-03
13	[0.2396972656;0.2402099609]	0.2399536132812500	1.252e-03
14	[0.2396972656;0.2399536133]	0.2398254394531250	4.717e-04
15	[0.2396972656;0.2398254395]	0.2397613525390626	8.145e-05
16	[0.2396972656;0.2397613525]	0.2397293090820313	-1.137e-04
17	[0.2397293091;0.2397613525]	0.2397453308105469	-1.611e-05
18	[0.2397453308;0.2397613525]	0.2397533416748047	3.267e-05
19	[0.2397453308;0.2397533417]	0.2397493362426759	8.279e-06
20	[0.2397453308;0.2397493362]	0.2397473335266114	-3.916e-06
21	[0.2397473335;0.2397493362]	0.2397483348846436	2.182e-06
22	[0.2397473335;0.2397483349]	0.2397478342056275	-8.670e-07
23	[0.2397478342;0.2397483349]	0.2397480845451356	6.573e-07
24	[0.2397478342;0.2397480845]	0.2397479593753815	-1.049e-07
25	[0.2397479594;0.2397480845]	0.2397480219602586	2.762e-07
26	[0.2397479594;0.2397480220]	0.2397479906678200	8.568e-08
27	[0.2397479594;0.2397479907]	0.2397479750216008	-9.593e-09
28	[0.2397479750;0.2397479907]	0.2397479828447104	3.804e-08
29	[0.2397479750;0.2397479828]	0.2397479789331556	1.422e-08
30	[0.2397479750;0.2397479789]	0.2397479769773782	2.315e-09
31	[0.2397479750;0.2397479770]	0.2397479759994895	-3.639e-09
32	[0.2397479760;0.2397479770]	0.2397479764884338	-6.619e-10
33	[0.2397479765;0.2397479770]	0.2397479767329060	8.267e-10
34	[0.2397479765;0.2397479767]	0.2397479766106699	8.241e-11
35	[0.2397479765;0.2397479766]	0.2397479765495519	-2.897e-10
36	[0.2397479765;0.2397479766]	0.2397479765801109	-1.037e-10
37	[0.2397479766;0.2397479766]	0.2397479765953904	-1.062e-11
38	[0.2397479766;0.2397479766]	0.2397479766030302	3.590e-11
39	[0.2397479766;0.2397479766]	0.2397479765992103	1.264e-11
40	[0.2397479766;0.2397479766]	0.2397479765973003	1.007e-12
41	[0.2397479766;0.2397479766]	0.2397479765963453	-4.808e-12
42	[0.2397479766;0.2397479766]	0.2397479765968228	-1.901e-12
43	[0.2397479766;0.2397479766]	0.2397479765970616	-4.472e-13
44	[0.2397479766;0.2397479766]	0.2397479765971809	2.796e-13
45	[0.2397479766;0.2397479766]	0.2397479765971213	-8.371e-14
46	[0.2397479766;0.2397479766]	0.2397479765971511	9.814e-14
47	[0.2397479766;0.2397479766]	0.2397479765971362	7.105e-15
48	[0.2397479766;0.2397479766]	0.2397479765971287	-3.830e-14
49	[0.2397479766;0.2397479766]	0.2397479765971325	-1.521e-14
50	[0.2397479766;0.2397479766]	0.2397479765971343	-3.997e-15
51	[0.2397479766;0.2397479766]	0.2397479765971353	1.443e-15
52	[0.2397479766;0.2397479766]	0.2397479765971348	-1.332e-15
53	[0.2397479766;0.2397479766]	0.2397479765971350	0.000e+00

Metoda bisekcji drugie miejsce zerowe			
iteracja	**przedział**	**wynik**	**wartość funkcji**
1	[1.9000000000;4.0000000000]	2.9500000000000002	-1.639e+00
2	[1.9000000000;2.9500000000]	2.4249999999999998	3.667e+00
3	[2.4250000000;2.9500000000]	2.6875000000000000	1.544e+00
4	[2.6875000000;2.9500000000]	2.8187500000000001	1.008e-01
5	[2.8187500000;2.9500000000]	2.8843750000000004	-7.300e-01
6	[2.8187500000;2.8843750000]	2.8515625000000000	-3.051e-01
7	[2.8187500000;2.8515625000]	2.8351562499999998	-9.980e-02
8	[2.8187500000;2.8351562500]	2.8269531250000002	1.073e-03
9	[2.8269531250;2.8351562500]	2.8310546875000000	-4.922e-02
10	[2.8269531250;2.8310546875]	2.8290039062500001	-2.403e-02
11	[2.8269531250;2.8290039063]	2.8279785156250004	-1.147e-02
12	[2.8269531250;2.8279785156]	2.8274658203125003	-5.197e-03
13	[2.8269531250;2.8274658203]	2.8272094726562500	-2.061e-03
14	[2.8269531250;2.8272094727]	2.8270812988281251	-4.938e-04
15	[2.8269531250;2.8270812988]	2.8270172119140629	2.897e-04
16	[2.8270172119;2.8270812988]	2.8270492553710938	-1.020e-04
17	[2.8270172119;2.8270492554]	2.8270332336425783	9.385e-05
18	[2.8270332336;2.8270492554]	2.8270412445068360	-4.091e-06
19	[2.8270332336;2.8270412445]	2.8270372390747074	4.488e-05
20	[2.8270372391;2.8270412445]	2.8270392417907715	2.040e-05
21	[2.8270392418;2.8270412445]	2.8270402431488035	8.152e-06
22	[2.8270402431;2.8270412445]	2.8270407438278200	2.030e-06
23	[2.8270407438;2.8270412445]	2.8270409941673282	-1.031e-06
24	[2.8270407438;2.8270409942]	2.8270408689975741	4.999e-07
25	[2.8270408690;2.8270409942]	2.8270409315824514	-2.653e-07
26	[2.8270408690;2.8270409316]	2.8270409002900125	1.173e-07
27	[2.8270409003;2.8270409316]	2.8270409159362320	-7.400e-08
28	[2.8270409003;2.8270409159]	2.8270409081131223	2.165e-08
29	[2.8270409081;2.8270409159]	2.8270409120246773	-2.618e-08
30	[2.8270409081;2.8270409120]	2.8270409100688996	-2.265e-09
31	[2.8270409081;2.8270409101]	2.8270409090910107	9.691e-09
32	[2.8270409091;2.8270409101]	2.8270409095799551	3.713e-09
33	[2.8270409096;2.8270409101]	2.8270409098244276	7.242e-10
34	[2.8270409098;2.8270409101]	2.8270409099466636	-7.704e-10
35	[2.8270409098;2.8270409099]	2.8270409098855458	-2.310e-11
36	[2.8270409098;2.8270409099]	2.8270409098549867	3.505e-10
37	[2.8270409099;2.8270409099]	2.8270409098702665	1.637e-10
38	[2.8270409099;2.8270409099]	2.8270409098779061	7.032e-11
39	[2.8270409099;2.8270409099]	2.8270409098817257	2.361e-11
40	[2.8270409099;2.8270409099]	2.8270409098836358	2.629e-13
41	[2.8270409099;2.8270409099]	2.8270409098845910	-1.142e-11
42	[2.8270409099;2.8270409099]	2.8270409098841132	-5.578e-12
43	[2.8270409099;2.8270409099]	2.8270409098838742	-2.650e-12
44	[2.8270409099;2.8270409099]	2.8270409098837552	-1.201e-12
45	[2.8270409099;2.8270409099]	2.8270409098836957	-4.725e-13
46	[2.8270409099;2.8270409099]	2.8270409098836655	-1.030e-13
47	[2.8270409099;2.8270409099]	2.8270409098836504	8.171e-14
48	[2.8270409099;2.8270409099]	2.8270409098836580	-1.066e-14
49	[2.8270409099;2.8270409099]	2.8270409098836540	3.908e-14
50	[2.8270409099;2.8270409099]	2.8270409098836558	1.421e-14
51	[2.8270409099;2.8270409099]	2.8270409098836566	3.553e-15
52	[2.8270409099;2.8270409099]	2.8270409098836575	-3.553e-15
53	[2.8270409099;2.8270409099]	2.8270409098836571	0.000e+00

1.8 Wnioski

Analizując wykres funkcji danej w zadaniu możemy wstępnie określić miejsca zerowe i zachowanie funkcji w ich otoczeniu. Rozpatrywana funkcja posiada dwa rodzaje miejsc zerowych które mają wpływ na szybkość ich wyznaczania, ponieważ w otoczeniu jednego z nich funkcja jest nachylona pod małym kątem do osi X natomiast w przypadku drugiego miejsca obserwujemy duży lokalny przyrost, więc wykres funkcji jest mniej odchylony od osi pionowej. Jak wynika z otrzymanych rezultatów metoda bisekcji w obu przypadkach potrzebowała stosunkowo dużej ilości iteracji aby osiągać zadaną dokładność. Jej zaletami jest niewrażliwość na szczególne zachowania funkcji więc mimo słabej zbieżności nadaje się do wyznaczania miejsc zerowych funkcji których przebiegu nie znamy w celu zabezpieczenia przez niezbieżnością. Metoda siecznych potrzebowała około 10 iteracji aby dojść do wyniku. W przypadku wyznaczania miejsca zerowego w otoczeniu którego mamy do czynienia z płaską funkcją, jak miało to miejsce w drugim miejscu zerowym, metoda okazuje się najlepsza spośród wszystkich zastosowanych. Jej wadą jest niezbieżność w przypadku gdy sieczna przetnie wykres funkcji w większej ilości miejsc niż 2. Dlatego najeży wybierać wąskie przedziały startowe co zostało uczynione w rozwiązaniu. Ostatnią wypróbowaną metodą jest metoda Newtona (stycznych), która w przypadku dobrego uwarunkowania (wąski przedział, brak odcinków o pochodnej mniejszej niż zero w przedziale startowym) okazuje się być najszybsza (w zadaniu ok 3-4 iteracje). Wynika to z najwyższego współczynnika zbieżności. W przypadku wyznaczania drugiego miejsca zerowego metodą Newtona rozpatrywany przedział zawierał fragment funkcji z ujemną pochodną. Spowodowało to zwiększenie liczby iteracji dla metody Newtona co można uznać za przypadek kiedy funkcja zawiodła.

2 Zadanie 2

2.1 Polecenie

Używając metody Mullera MM2, proszę znaleźć wszystkie pierwiastki rzeczywiste i zespolone wielomianu

$$f(x) = a_4x^4 + a_3x^3 + a_2x^2 + a_1x + a_0 \begin{bmatrix} a_4 & a_3 & a_2 & a_1 & a_0 \end{bmatrix} = \begin{bmatrix} 1 & 2 & 4 & -1 & 8 \end{bmatrix}$$

2.2 Metoda Mullera MM2

Metoda Mullera polega na przybliżeniu rozpatrywanej funkcji trójmianem kwadratowym w otoczeniu zera i na tej podstawie wyznaczenia miejsc zerowych rzeczywistych bądź zespolonych. Wyróżniamy dwa rodzaje metody Mullera. Pierwsza (MM1) polega na aproksymacji wielomianu na podstawie jego wartości w 3 różnych punktach, natomiast druga (MM2) wykorzystuje wartość funkcji, pierwszej oraz drugiej pochodnej w danym punkcie i na tej podstawie zostają wyznaczone współczynniki a, b ,c trójmianu kwadratowego. Służą do tego następujące zależności:

$$y(0) = c = f(x_k),\ y'(0) = b = f'(x_k),\ y''(0) = 2a = f''(x_k)$$

Mając wyznaczone współczynniki trójmianu kwadratowego możemy bezpośrednio przejść do wzorów na jego pierwiastki:

$$z_1 = \frac{-2f(x_k)}{f'(x_k)+\sqrt{(f'(x_k))^2-2f(x_k)f''(x_k)}}$$

$$z_2 = \frac{-2f(x_k)}{f'(x_k)-\sqrt{(f'(x_k))^2-2f(x_k)f''(x_k)}}$$

Przeprowadzając iteracyjne przybliżanie zera wybieramy pierwiastek o mniejszym module:

$$x_{k+1} = x_k + min(|z_1|, |z_2|)$$

Metoda Mullera podobnie jak Newtona jest zbieżna jedynie lokalnie z rzędem zbieżności 1.84. Jest szybsza od metody siecznych i niewiele wolniejsza od metody Newtona. Znaczącą różnicą jest implementacja pozwalająca na wyznaczanie zespolonych zer funkcji.

2.3 Realizacja w programie Matlab

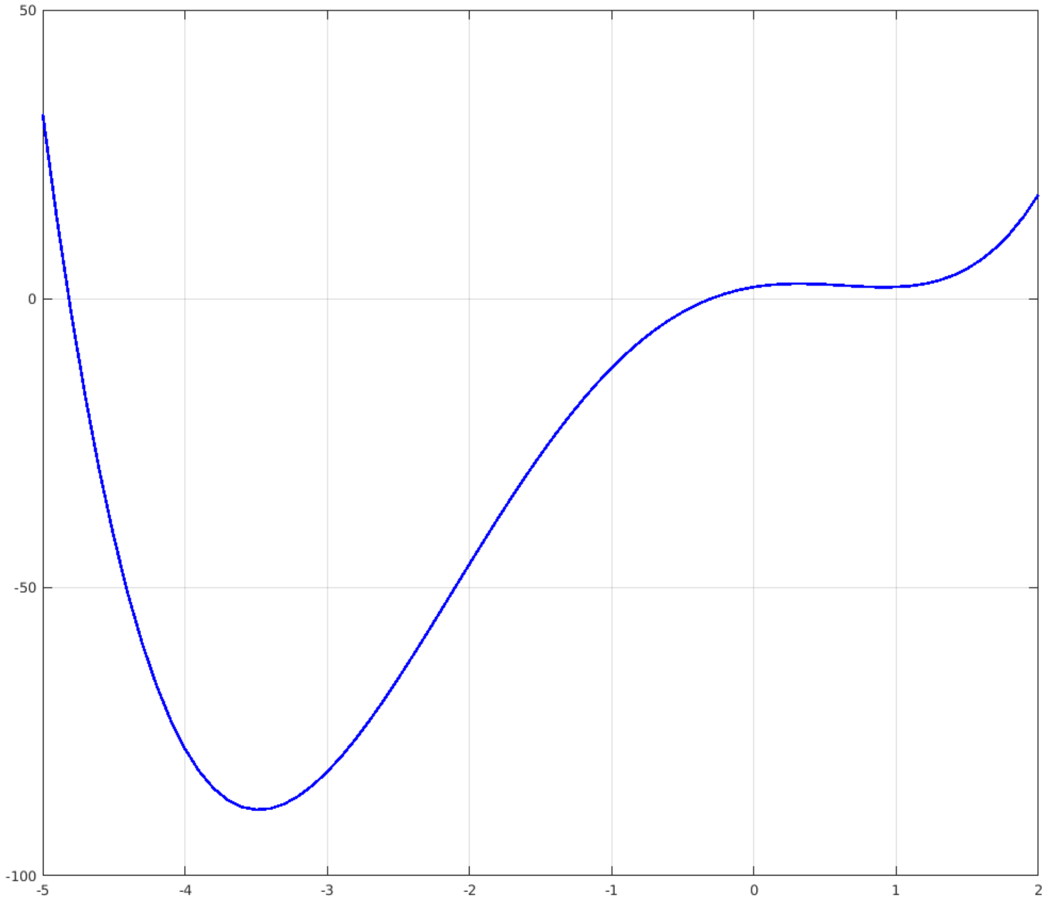
```
function [value] = df(x,n)
%DF funkcja zwracajaca wyanaczona alanlitycznie pochodna wielomianu
%danego w zadaniu.
%x - argument
%n  - 0-wartosac funkcji,1-pierwsza pochodna,2-druga pochodna
% f(x) = x^4+3x^3-8x^2+4x+2
% f'(x) = 4x^3+9x^2-16x+4
% f''(x) = 12x^2+18x-16
    if n == 0
        value = x.^4+3*x.^3-8*x.^2+4*x+2;
    elseif n == 1
        value = 4*x.^3+9*x.^2-16*x+4;
    elseif n == 2
        value = 12*x.^2+18*x-16;
    else
        value = 0;
    end
end

function [z] = muller(x,n)
%Funkcja zwracajaca wektor pierwiastkow rzeczywistych wielomianu
%danego w zadaniu
%A - wektor wspolczynnikow wielomianu
%x - punkt startowy
%n - ilosc iteracji
    %z = (1:n);
    for i = 1:n
        %obliczamy mianowniki punktow w celu ich porownania
        z1 = df(x,1)+sqrt(df(x,1)^2-2*df(x,0)*df(x,2));
        z2 = df(x,1)-sqrt(df(x,1)^2-2*df(x,0)*df(x,2));

        if abs(z1) > abs(z2)
            zmin = -2*df(x,0)/z1;
        else
            zmin = -2*df(x,0)/z2;
        end
        x = x+zmin;
    end
    z = x;
end
```

2.4 Wyniki działania programu

Rysunek 2: Wykres wielomianu z zadania 2



Miejsca zerowe wyznaczone alalitycznie			
-4.8158	-0.30075	1.05826 - 0.51087i	1.05826 + 0.51087i
Miejsca zerowe wyznaczone metodą Mullera			
-4.8158	-0.3007	1.0583 - 0.5109i	1.0583 + 0.5109i

Wyniki zadanie 2 cz. 1			
przedział startowy	iteracja	wynik	przyrost
-5	1	-4.815098	0.184902
-5	2	-4.815775	-0.000677
-5	3	-4.815775	0.000000
-5	4	-4.815775	0.000000
-5	5	-4.815775	-0.000000
-5	6	-4.815775	0.000000
-4	1	-4.872683	-0.872683
-4	2	-4.815756	0.056927
-4	3	-4.815775	-0.000019
-4	4	-4.815775	0.000000
-4	5	-4.815775	0.000000
-4	6	-4.815775	-0.000000
-3	1	-1.478763	1.521237
-3	2	-0.472623	1.006140
-3	3	-0.300054	0.172569
-3	4	-0.300747	-0.000693
-3	5	-0.300747	0.000000
-3	6	-0.300747	-0.000000
-2	1	-0.774964	1.225036
-2	2	-0.296363	0.478601
-2	3	-0.300747	-0.004383
-2	4	-0.300747	0.000000
-2	5	-0.300747	-0.000000
-2	6	-0.300747	-0.000000
-1	1	-0.311312	0.688688
-1	2	-0.300746	0.010565
-1	3	-0.300747	-0.000000
-1	4	-0.300747	-0.000000
-1	5	-0.300747	-0.000000
-1	6	-0.300747	-0.000000

Wyniki zadanie 2 cz. 2			
przedział startowy	iteracja	wynik	przyrost
0	1	-0.309017	-0.309017
0	2	-0.300747	0.008270
0	3	-0.300747	-0.000000
0	4	-0.300747	-0.000000
0	5	-0.300747	-0.000000
0	6	-0.300747	-0.000000
1	1	0.928571	-0.071429
1	2	1.057209	0.128638
1	3	1.058261	0.001052
1	4	1.058261	-0.000000
1	5	1.058261	-0.000000
1	6	1.058261	0.000000
2	1	1.411765	-0.588235
2	2	1.089902	-0.321863
2	3	1.058224	-0.031678
2	4	1.058261	0.000037
2	5	1.058261	0.000000
2	6	1.058261	-0.000000
3	1	2.006849	-0.993151
3	2	1.257315	-0.749534
3	3	1.044287	-0.213028
3	4	1.058128	0.013841
3	5	1.058261	0.000133
3	6	1.058261	0.000000
4	1	2.629032	-1.370968
4	2	2.004778	-0.624254
4	3	1.414140	-0.590638
4	4	1.090117	-0.324023
4	5	1.058222	-0.031896
4	6	1.058261	0.000039

2.5 Wnioski

Metoda Mullera dla wielomianu danego w zadaniu okazała się bardzo szybka, potrzebnych było jedynie kilka iteracji aby dojść do wyników zbliżonych do rozwiązania analitycznego. Komentarza wymaga realizacja zadania, ponieważ metoda została zastosowana kolejno do przedziałów o długości 1 rozpoczynając od pierwszego zawierającego zero funkcji (co zostało ustalone na podstawie obliczeń analitycznych). Dzięki aproksymacji kwadratowej metoda zawsze zbiegała do bliższego od wierzchołka paraboli zera, co uniemożliwiło pominięcie rozwiązania podczas badania kolejnych przedziałów. Jeżeli chodzi o pierwiastki zespolone spełniają one warunek wzajemnego sprzężenia. Wszystkich rozwiązań wyszło tyle ile wynosi stopień wielomianu co również jest zgodne z teorią.