MNUM-PROJEKT, zadanie 3.17

Marcin Dziedzic

7 maja 2018

Spis treści

1 Zadanie I

1.1 Tresc zadania

Prosze znalezc wszystkie zera funkcji

$$f(x) = 0.7xcos(x) - ln(x+1)$$

w przedziale $[2, 11]$, uzywajac dla kazdego zera programu z implementacja:
a) metody bisekcji
b) metody siecznych
c) metody Newtona

1.2 Metoda bisekcji

1.2.1 Opis teoretyczny

Teoretyczny zarys metody bisekcji mozemy przyblizyc ponizszym algorytmem:

1. Poczawszy od przedzialu startowego $[a, b] = [a_0, b_0]$ obliczamy srodek przedzialu c_n, $c_n = \frac{a_n + b_n}{2}$ i obliczamy wartosc $f(x)$ w tym punkcie.

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

Kroki te powtarzamy az do momentu uzyskania $f(c_n) < \delta$ gdzie δ to oczekiwana dokladnosc rozwiazania. W przypadku "plaskich"funkcji warto tez kontrolowac dlugosc rozpatrywanego przedzialu. Dokladnosc wyniku zalezy jedynie od ilosci iteracji dlatego metoda jest zbiezna liniowo z ilorazem zbieznosci 0.5, co czyni ja stosunkowo wolno zbiezna w przypadku wyboru szerokiego przedzialu poczatkowego.

1.2.2 Realizacja w programie Matlab

Listing 1: Implementacja metody bisekcji

```
% Funkcja wyznaczajaca punkty zerowe funkcji metoda bisekcji
%
% IN:
% a0, b0 - zakres
% fun - funkcja
% iter - maksymalna liczba iteracji
%
% OUT:
% solution - wyznaczone miejsce zerowe
```

```
function soluiton = md_bisection(fun,a0,b0,iter)

  a = a0;
  b = b0;
  % inicjalizacja wartosciami poczatkowymi
  fa =feval(fun,a);
  fb =feval(fun,b);
  for k=1:iter
    % obliczenie srodka odcinka
    xm = a + 0.5*(b-a);
    %  f(xm)
    fm = feval(fun,xm);
    fprintf('%3d    [%12.10f;%12.10f]   %12.16f     %12.3e\n',k,a,b,xm,fm);
    if(fm == 0)
        return
    end
    %  Zero lezy w przedziale [xm,b], zamiana a
    if sign(fm)==sign(fa)
        a = xm;
        fa = fm;
    %  Zero lezy w przedziale [a,xm], zamiana b
    else
        b = xm;
        fb = fm;
    end
    %dodatkowy warunek zakonczenia wykonywania
    if(fm == 0)
        return
    end
  end
  soluiton = xm;
  return
end
```

1.3 Metoda siecznych

Metoda siecznch jest bardzo podobna do metody bisekcji, rozni sie tym, iz przez krance aktualnego przedzialu prowadzimy prosta, ktora przecina os rzednych w punkcie x_0 . Nastepnie prowadzimy kolejna prosta przez punkt $f(x_n)$ i poprzedni punkt $f(x_{n-1})$. Warto zaznaczyc, ze nie dbamy tutaj o przedzial izolacji pierwiastka, ani nawet o znak iloczynu wartosci na krancach aktualnego przedzialu.

Wzor na n+1 punkt, przez ktory ma przejsc prosta: $x_{n+1} = \frac{x_{n-1}f(x_n)-x_n f(x_{n-1})}{f(x_n)-f(x_{n-1})}$

Rzad zbieznosci metody siecznych wynosi $\frac{1+\sqrt{5}}{2}$ zatem metoda ta jest szybsza od metody bisekcji, lecz w praktyce moze okazac sie niezbiezna kiedy przedzial izolacji nie jest dostatecznie maly, gdyz jest ona zbiezna tylko lokalnie, a nie globalnie jak poprzednia metoda.

1.3.1 Realizacja w programie Matlab

Listing 2: Implementacja metody siecznych

```
% Funkcja wyznaczajaca punkty zerowe funkcji metoda siecznych
%
% IN:
% a0, b0 - zakres
% fun - funkcja
% iter - maksymalna liczba iteracji
%
% OUT:
% solution - wyznaczone miejsce zerowe
function solution = md_secans(fun, a0, b0, iter)
  a = a0;
  b = b0;
  % poczatkowa wartosc funkcji
  fa = feval(fun,a);
  for k = 1:iter
    fb = feval(fun,b);
    dx = fb * (b-a) / (fb-fa);
    xm = b-dx;
    if(isnan(xm))
        return
    end
    a = b;
    b = xm;
    fa = fb;
    solution = b;
    fprintf('%3d         [%12.10f;%12.10f]   %12.16f     %12.3e\n',k,a,b,xm,fb);
    %dodatkowy warunek zakonczenia wykonywania
    if(fb == 0)
        return
    end
```

```
  end
end
```

1.4 Metoda Newtona

1.4.1 Opis teoretyczny

Metoda Newtona polega na wyznaczeniu czesciowego (ucietego) rozwiniecia w szereg Taylora danej funkcji, ktore mozemy traktowac jak liniowe przyblizenie funkcji wedlug wzoru:

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

Nastepnie wyznaczamy kolejne punkty iteracji poprzez przyrownanie do zera otrzymanej aproksymacji:

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

Prowadzi to do zaleznosci iteracyjnej danej nastepujacym wzorem:

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

Metoda Newtona jest zbiezna jedynie lokalnie, poniewaz wyznaczajac styczna do wykresu w danym punkcie mozemy w przypadku ujemnego znaku pochodnej dojsc do rozbieznosci. Dla przypadkow pochodnej wiekszej od zera metoda jest zbiezna kwadratowo. Rzad zbieznosci wynosi 2.

1.4.2 Realizacja w programie Matlab

Listing 3: Implementacja metody Newtona

```
% Funkcja wyznaczajaca punkty zerowe funkcji metoda Newtona
%
% IN:
% a0 - lewa strona zakresu
% fun - funkcja
% iter - maksymalna liczba iteracji
%
% OUT:
% solution - wyznaczone miejsce zerowe
function solution = md_newton(fun, a0,iter)
  x0 = a0;
  for k = 1:iter
```

```
    [fold, fpold] = feval(fun,x0);
    dx = fold / fpold;
    x0 = x0 - dx;
    fprintf('%3d␣␣␣␣␣␣␣␣␣%12.10f␣␣␣␣␣%12.16f␣␣␣␣␣%12.3e␣\n',k,dx,x0,fold);
    if(fold == 0)
        return
    end
    %dodatkowy warunek zatrzumania
        if fold==0
        solution = x0;
        break;
    end
  end
end
```

1.5 Analiza danych wejsciowych

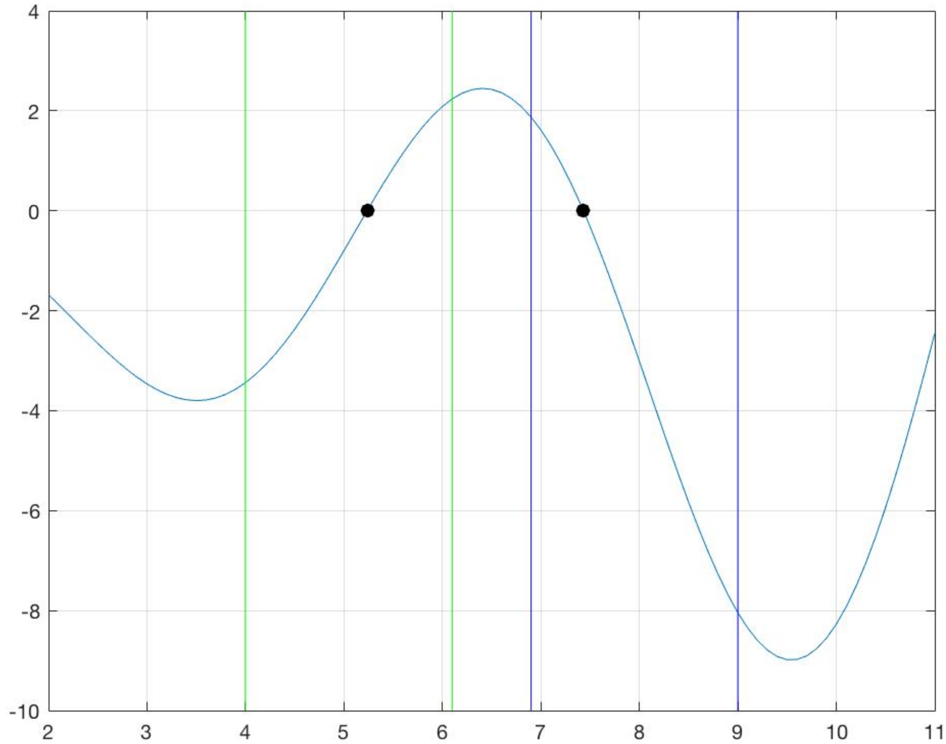
W celu wyznaczenia przedzialow izolacji miejsc zerowych zostal wykorzystany algorytm opisany w skrypcie prof. Tatjewskiego. Wstepna analiza danych rozpoczyna sie od wygenerowania wykresu funkcji w danym przedziale i na tej podstawie wyboru przedzialu startowego dla algorytmu. Nastepnie w podanym przedziale w petli badany jest znak iloczynu funkcji w punktach granicznych. Jezeli jest on ujemny, oznacza to wystepowanie miejsca zerowego w danym przedziale. Jezeli nie, to przedzial jest rozszerzany do momentu przekroczenia przedzialu danego w zadaniu. Ponizej wykres funkcji z zaznaczonymi przedzialami izolacji wyznaczonymi przez algorytm.

1.5.1 Realizacja w programie Matlab

Listing 4: Skrypt rozwiazujacy zadanie nr 1

```
% zadanie nr 1

clear;
x  = 2 : 0.1 : 11;
plot(x, md_fun_1(x))
grid on
axis([2 11 -10 4])
hold on
plot([4 4], [-10 4], 'g-');
plot([6.1 6.1], [-10 4], 'g-');
plot([6.9 6.9], [-10 4], 'b-');
```




```
plot([9 9], [-10 4], 'b-');
plot(5.2353163129141116, 0, '.','MarkerSize', 24, 'MarkerEdge', 'k');
plot(7.4317177653721664, 0,'.','MarkerSize',24,'MarkerEdge','k');

n=100;
x1 = 4;
x2 = 5;

% algorytmiczne wyznaczanie przedzialow izolacji
for k=1:2
    for j=1:n
        if md_fun_1(x1)*md_fun_1(x2)<0
            a = x1;
            b = x2;
            fprintf('Wyniki dla %d miejsca zerowego w przedziale [%d,%d]\n',k,a,
```

```
            bisection('md_fun_1',a,b,100);
            secant('md_fun_1',a,b,100);
            newton('md_fun_1',a,100);
            x1 = 8;
            x2 = 9;
            break;
        elseif abs(md_fun_1(x1))<abs(md_fun_1(x2))
            x1 = x1+1.1*(x1-x2);
        else
            x2 = x2+1.1*(x2-x1);
        end
        %wyjscie z petli po przekroczeniu przedzialu
        if(x1>11)&&(x2<(2))
            break;
        end
    end
end
```

1.6 Rozwiazanie

1.6.1 Metoda bisekcji

Pierwsze miejsce zerowe			
nr	przedzial	rozwiazanie	wartosc
1	[4.0000000000;6.1000000000]	5.0499999999999998	-6.291e-01
2	[5.0500000000;6.1000000000]	5.5749999999999993	1.081e+00
3	[5.0500000000;5.5750000000]	5.3125000000000000	2.576e-01
4	[5.0500000000;5.3125000000]	5.1812500000000004	-1.826e-01
5	[5.1812500000;5.3125000000]	5.2468750000000002	3.884e-02
6	[5.1812500000;5.2468750000]	5.2140625000000007	-7.163e-02
7	[5.2140625000;5.2468750000]	5.2304687500000000	-1.631e-02
8	[5.2304687500;5.2468750000]	5.2386718749999996	1.129e-02
9	[5.2304687500;5.2386718750]	5.2345703124999998	-2.510e-03
10	[5.2345703125;5.2386718750]	5.2366210937499993	4.389e-03
11	[5.2345703125;5.2366210937]	5.2355957031250000	9.399e-04
12	[5.2345703125;5.2355957031]	5.2350830078125004	-7.849e-04
13	[5.2350830078;5.2355957031]	5.2353393554687502	7.752e-05
14	[5.2350830078;5.2353393555]	5.2352111816406257	-3.537e-04
15	[5.2352111816;5.2353393555]	5.2352752685546875	-1.381e-04
16	[5.2352752686;5.2353393555]	5.2353073120117184	-3.028e-05
17	[5.2353073120;5.2353393555]	5.2353233337402347	2.362e-05
18	[5.2353073120;5.2353233337]	5.2353153228759766	-3.331e-06
19	[5.2353153229;5.2353233337]	5.2353193283081056	1.014e-05
20	[5.2353153229;5.2353193283]	5.2353173255920407	3.407e-06
21	[5.2353153229;5.2353173256]	5.2353163242340086	3.808e-08
22	[5.2353153229;5.2353163242]	5.2353158235549930	-1.646e-06
23	[5.2353158236;5.2353163242]	5.2353160738945004	-8.041e-07
24	[5.2353160739;5.2353163242]	5.2353161990642541	-3.830e-07
25	[5.2353161991;5.2353163242]	5.2353162616491318	-1.725e-07
26	[5.2353162616;5.2353163242]	5.2353162929415706	-6.719e-08
27	[5.2353162929;5.2353163242]	5.2353163085877892	-1.455e-08
28	[5.2353163086;5.2353163242]	5.2353163164108985	1.176e-08
29	[5.2353163086;5.2353163164]	5.2353163124993438	-1.395e-09
30	[5.2353163125;5.2353163164]	5.2353163144551207	5.184e-09
31	[5.2353163125;5.2353163145]	5.2353163134772327	1.894e-09
32	[5.2353163125;5.2353163135]	5.2353163129882887	2.495e-10
33	[5.2353163125;5.2353163130]	5.2353163127438158	-5.729e-10
34	[5.2353163127;5.2353163130]	5.2353163128660523	-1.617e-10
35	[5.2353163129;5.2353163130]	5.2353163129271705	4.393e-11
36	[5.2353163129;5.2353163129]	5.2353163128966109	-5.887e-11
37	[5.2353163129;5.2353163129]	5.2353163129118911	-7.469e-12
38	[5.2353163129;5.2353163129]	5.2353163129195313	1.823e-11
39	[5.2353163129;5.2353163129]	5.2353163129157112	5.382e-12
40	[5.2353163129;5.2353163129]	5.2353163129138007	-1.045e-12

Drugie miejsce zerowe			
nr	przedzial	rozwiazanie	wartosc
1	[6.9000000000;9.0000000000]	7.9500000000000002	-2.725e+00
2	[6.9000000000;7.9500000000]	7.4250000000000007	3.067e-02
3	[7.4250000000;7.9500000000]	7.6875000000000000	-1.270e+00
4	[7.4250000000;7.6875000000]	7.5562500000000004	-5.950e-01
5	[7.4250000000;7.5562500000]	7.4906250000000005	-2.754e-01
6	[7.4250000000;7.4906250000]	7.4578125000000011	-1.206e-01
7	[7.4250000000;7.4578125000]	7.4414062500000009	-4.450e-02
8	[7.4250000000;7.4414062500]	7.4332031250000004	-6.802e-03
9	[7.4250000000;7.4332031250]	7.4291015625000005	1.196e-02
10	[7.4291015625;7.4332031250]	7.4311523437500000	2.587e-03
11	[7.4311523438;7.4332031250]	7.4321777343750002	-2.106e-03
12	[7.4311523438;7.4321777344]	7.4316650390624996	2.413e-04
13	[7.4316650391;7.4321777344]	7.4319213867187504	-9.320e-04
14	[7.4316650391;7.4319213867]	7.4317932128906250	-3.453e-04
15	[7.4316650391;7.4317932129]	7.4317291259765623	-5.200e-05
16	[7.4316650391;7.4317291260]	7.4316970825195305	9.466e-05
17	[7.4316970825;7.4317291260]	7.4317131042480469	2.133e-05
18	[7.4317131042;7.4317291260]	7.4317211151123050	-1.533e-05
19	[7.4317131042;7.4317211151]	7.4317171096801760	3.001e-06
20	[7.4317171097;7.4317211151]	7.4317191123962409	-6.165e-06
21	[7.4317171097;7.4317191124]	7.4317181110382080	-1.582e-06
22	[7.4317171097;7.4317181110]	7.4317176103591915	7.095e-07
23	[7.4317176104;7.4317181110]	7.4317178606986998	-4.363e-07
24	[7.4317176104;7.4317178607]	7.4317177355289452	1.366e-07
25	[7.4317177355;7.4317178607]	7.4317177981138229	-1.499e-07
26	[7.4317177355;7.4317177981]	7.4317177668213841	-6.633e-09
27	[7.4317177355;7.4317177668]	7.4317177511751646	6.498e-08
28	[7.4317177512;7.4317177668]	7.4317177589982748	2.917e-08
29	[7.4317177590;7.4317177668]	7.4317177629098294	1.127e-08
30	[7.4317177629;7.4317177668]	7.4317177648656063	2.319e-09
31	[7.4317177649;7.4317177668]	7.4317177658434952	-2.157e-09
32	[7.4317177649;7.4317177658]	7.4317177653545503	8.063e-11
33	[7.4317177654;7.4317177658]	7.4317177655990232	-1.038e-09
34	[7.4317177654;7.4317177656]	7.4317177654767868	-4.788e-10
35	[7.4317177654;7.4317177655]	7.4317177654156685	-1.991e-10
36	[7.4317177654;7.4317177654]	7.4317177653851090	-5.924e-11
37	[7.4317177654;7.4317177654]	7.4317177653698296	1.070e-11
38	[7.4317177654;7.4317177654]	7.4317177653774689	-2.427e-11
39	[7.4317177654;7.4317177654]	7.4317177653736497	-6.789e-12
40	[7.4317177654;7.4317177654]	7.4317177653717401	1.952e-12
41	[7.4317177654;7.4317177654]	7.4317177653726949	-2.419e-12
42	[7.4317177654;7.4317177654]	7.4317177653722180	-2.354e-13
43	[7.4317177654;7.4317177654]	7.4317177653719790	8.584e-13
44	[7.4317177654;7.4317177654]	7.4317177653720989	3.091e-13
45	[7.4317177654;7.4317177654]	7.4317177653721584	3.642e-14
46	[7.4317177654;7.4317177654]	7.4317177653721878	-9.726e-14
47	[7.4317177654;7.4317177654]	7.4317177653721735	-3.242e-14
48	[7.4317177654;7.4317177654]	7.4317177653721664	0.000e+00

1.6.2 Metoda siecznych

Pierwsze miejsce zerowe			
nr	przedzial	rozwiazanie	wartosc
1	[6.1000000000;5.2721231045]	5.2721231045481129	2.238e+00
2	[5.2721231045;5.2238264588]	5.2238264587735248	1.234e-01
3	[5.2238264588;5.2353558422]	5.2353558421649877	-3.869e-02
4	[5.2353558422;5.2353163517]	5.2353163517211909	1.330e-04
5	[5.2353163517;5.2353163129]	5.2353163129139766	1.306e-07
6	[5.2353163129;5.2353163129]	5.2353163129141116	-4.534e-13
7	[5.2353163129;5.2353163129]	5.2353163129141116	8.882e-16

Drugie miejsce zerowe			
nr	przedzial	rozwiazanie	wartosc
1	[9.0000000000;7.2966891015]	7.2966891015391813	-8.043e+00
2	[7.2966891015;7.4122825051]	7.4122825051309968	5.855e-01
3	[7.4122825051;7.4328117423]	7.4328117422994682	8.831e-02
4	[7.4328117423;7.4317097704]	7.4317097703803334	-5.009e-03
5	[7.4317097704;7.4317177621]	7.4317177621310275	3.659e-05
6	[7.4317177621;7.4317177654]	7.4317177653721762	1.483e-08
7	[7.4317177654;7.4317177654]	7.4317177653721664	-4.396e-14
8	[7.4317177654;7.4317177654]	7.4317177653721664	0.000e+00

1.6.3 Metoda Newtona

Pierwsze miejsce zerowe			
nr	przedzial	rozwiazanie	wartosc
1	2.291e+01	-1.891e+01	-3.440e+00
2	-2.907e+02	2.718e+02	-1.610e+01
3	6.998e+52	-6.998e+52	8.442e+25
4	Inf	-Inf	-Inf
5	NaN	NaN	NaN
6	NaN	NaN	NaN

Drugie miejsce zerowe			
nr	przedzial	rozwiazanie	wartosc
1	-10.6337977481	17.5337977481481566	1.873e+00
2	-5.4087425453	22.9425402934045692	1.768e-01
3	457.6247656451	-434.6822253516609180	-1.250e+01
4	-94071.6545455790	93636.9723202273016796	-1.325e+02
5	NaN	NaN	Inf
6	NaN	NaN	NaN
7	NaN	NaN	NaN
8	NaN	NaN	NaN

1.7 Wnioski

Dzieki graficznej analizie wykresu funkcji w prosty sposob mozemy wyznaczyc otoczenie miejsc zerowych i zachowanie funkcji. Zauwazylem, ze funkcja posiada dwa miejsca zerowe. Wyznaczylem wiec przedzialy "badania"miejsc zerowych za pomoca algorytmu przedstawionego w ksiazce prof. Piotra Tatjewskiego.
Jak wynika z otrzymanych rezultatow metoda bisekcji w obu przypadkach potrzebowala stosunkowo duzej ilosci iteracji (okolo 50), aby osiagac zadana dokladnosc. Jej zaletami jest niewrazliwosc na szczegolne zachowania funkcji wiec mimo slabej zbieznosci nadaje sie do wyznaczania miejsc zerowych funkcji ktorych przebiegu nie znamy w celu zabezpieczenia przez niezbieznoscia.
Metoda siecznych potrzebowala okolo 8 iteracji aby dojsc do wyniku. W przypadku badanej funkcji metoda ta okazala sie najlepsza. O wiele szybsza niz metoda bisekcji oraz zbiezna dla podanej funkcji. Wynika to z wiekszego wspolczynnika zbieznosci Niestety dla innych funkcji np. gdy ieczna przetnie wykres funkcji w wiekszej ilosci miejsc niz 2 razy moze okazac sie niezbiezna.
Metoda Newtona zupelnie zawiodla w moim rozwiazaniu. Rozpatrywany przedzial zawieral fragment funkcji z ujemna pochodna. Metoda okazala sie rozbiezna. Teorytycznie metoda ta powinna okazac sie najszybsza, poniewaz ma najwyzszy wspolczynnik zbieznosci.

2 Zadanie II

2.1 Tresc zadania

Uzywajac metody Mullera MM1, prosze znalezc wszystkie pierwiastki rzeczywiste i zespolone wielomianu

$$f(x) = a_4x^4 + a_3x^3 + a_2x^2 + a_1x + a_0, \begin{bmatrix} a_4 & a_3 & a_2 & a_1 & a_0 \end{bmatrix} = \begin{bmatrix} 1 & -7 & -4 & 2 & 9 \end{bmatrix}$$

2.2 Opis teorytyczny

Metoda Mullera : Powyzsze metody szukania zer funkcji jak najbardziej spelniaja swoje zadanie, lecz dzieki nim nie znajdziemy pierwiastkow zespolonych. Polecana metoda do znalezienia wszystkich pierwiastkow jest metoda Mullera zbiezna lokalnie. MM1 Pierwsza metoda Mullera polega na aproksymacji funkcji w otoczeniu rozwiazania funkcja kwadratowa. Moze byc traktowana jako uogolnienie metody siecznych, gdyz zamiast interpolacji funkcja liniowa w dwoch punktach interpolujemy funkcja kwadratowa w trzech. Wezmy trzy punkty x0, x1 i x2 oraz wartosci funkcji w tych punktach. Istnieje dokladnie jedna funkcja kwadratowa przechodzaca przez te trzy wartosci f(x0), f(x1) i f(x2) . Metoda polega na znajdowaniu pierwiastkow funkcji kwadrtatowej i potraktowaniu jednego z nich jako kolejnego przyblizonego pierwiastka wyjsciowej funkcji. Na przyklad mamy dwa pierwiastki w czwartej iteracji x4.1, x4.2 to wybieramy ten, ktory lezy blizej poprzedniego przyblizenia i tak dalej.

2.3 Rozwiazanie

Listing 5: Implementacja funkcji MM1

```
% implementacja algorytmu Mullera MM1
% Funkcja zwraca rozwiazanie oraz
% macierz wartosci i punktow
function [x,mm]= MM1(fun,x0,x1,x2,pre)
  i=1;
  x=100;
  counter = 0;
  while abs(feval(fun, x))>pre
      counter = counter +1;
    z0=x0-x2;
    z1=x1-x2;
    c=feval(fun, x2);% Rozwiazujemy uklad rownan liniowych
% by otrzymac wszystkie wspolczynniki paraboli
    A=[z0^2,z0;z1^2,z1];
    B=[feval(fun, x0)-c;feval(fun, x1)-c];
    r=A\B;
    a=r(1);
    b=r(2);
    % Wybor warunka minimalnego dla z
    if abs(b+sqrt(b.^2-4*a*c))>=abs(b-sqrt(b.^2-4*a*c))
      zmin=((-2)*c)/(b+sqrt(b.^2-4*a*c));
    else
```

```
      zmin=((-2)*c)/(b-sqrt(b.^2-4*a*c));
    end
    % obliczenie szukanego pierwiastka x
    x=x2+zmin;
    mm(i,1)=x;
    mm(i,2)=feval(fun,x);
    %szukamy najbliszego pierwiastka
    %ktory jest oddalony od x
    p0=abs(x-x0);
    p1=abs(x-x1);
    p2=abs(x-x2);
    if p0>p1 %gdy x0 jest dalej
      m=x1;
      x1=x0;
      x0=m;
    end
    if p1>p2 %gdy x1 jest dalej niz x2
    m=x2;
    x2=x1;
    x1=m;
    end
    x2=x;
    if isnan(c)
      break;
    end
    i = i +1;
  end
  counter
end
```

Listing 6: Skrypt rozwiazujacy zadanie nr 2

```
%skrypt generujacy wyniki do zadania 2
clear;
x = -10: .1 : 20;
plot(x, md_fun_2(x))
grid on
% axis([-5 2 -100 50])
hold on
plot(1.0420, 0, '.','MarkerSize', 24, 'MarkerEdge', 'k');
plot(7.4776, 0, '.','MarkerSize', 24, 'MarkerEdge', 'k');
n=100;
```

```
x1 = -8;
x2 = -7;
% x1 = 4
% x2=5
% algorytmiczne wyznaczanie przedzialow izolacji
for k=1:10
    for j=1:n
        if md_fun_1(x1)*md_fun_1(x2)<0
            a = x1;
            b = x2;
            fprintf('Wyniki dla %d miejsca zerowego w przedziale [%d,%d]\n',k,a,
            md_MM1('md_fun_2', x1, (x1+x2)/2, x2, 0.001)
            x1 = x1+1;
            x2 = x2 +2;
            break;
        elseif abs(md_fun_1(x1))<abs(md_fun_1(x2))
            x1 = x1+1.1*(x1-x2);
        else
            x2 = x2+1.1*(x2-x1);
        end
        %wyjscie z petli po przekroczeniu przedzialu
        if(x1>11)&&(x2<(2))
            break;
        end
    end
end
```

2.4 Wyniki

Wyniki		
nr	cz. rzeczywista	cz. urojona
1	1.0420	0i
2	7.4776	0i
3	-0.75982	0.76009i
4	-0.75982	-0.76009i

2.5 Wnioski

Metoda Mullera dla wielomianu danego w zadaniu okazala sie bardzo skuteczna. Dzieki aproksymacji kwadratowej metoda zawsze zbiegala do blizszego od wierzcholka paraboli zera, co uniemozliwilo pominiecie rozwiazania podczas badania kolejnych przedzialow. Jezeli chodzi o pierwiastki zespolone spelniaja one warunek wzajemnego sprzezenia. Wszystkich rozwiazan wyszlo tyle ile wynosi stopien wielomianu co rowniez jest zgodne z teoria.