

MNUM PROJEKT 3 - zestaw 3.1

Zadanie 1

Należy wyznaczyć wszystkie zera funkcji

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

w przedziale $[-5.5]$, używając dla każdego zera programu z implementacją:

- a) Metody bisekcji
- b) Metody siecznych
- c) Metody Newtona

Zero funkcji (lub używaną zamiennie nazwą: pierwiastkiem) funkcji $f(x)$ jest takie x , dla którego jest spełnione:

$$f(x) = 0$$

O ile znane są metody wyznaczania zer dla równań liniowych i kwadratowych, to dla funkcji nieliniowych sprawa nie jest taka oczywista.

Wyznaczanie zer funkcji nieliniowej: Aby wyznaczyć zera funkcji należy dla każdego zera wyznaczyć przedział, w którym znajduje się tylko dane jedno zero. Mając określony taki przedział można wykorzystując metody iteracyjne wyznaczyć przybliżenie danego zera.

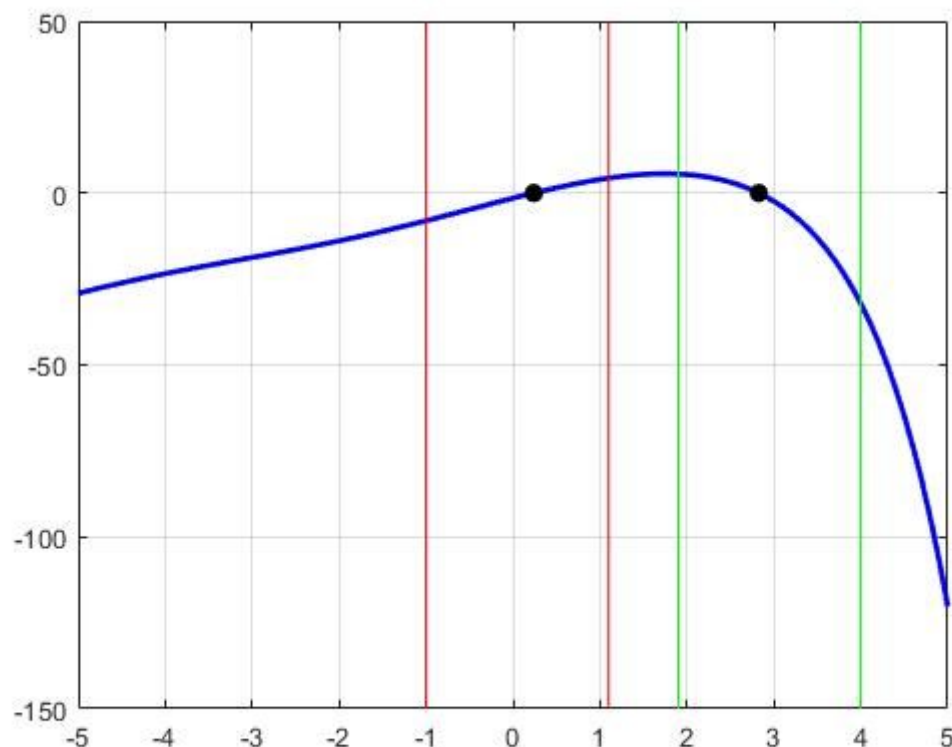
Wyznaczanie przedziału zawierającego pierwiastek:

- a) Można ręcznie oszacować przedział zawierający pierwiastek za pomocą wykresu, rysującego funkcję. Na potrzeby tego sprawozdania przygotowano funkcję matlabową generującą taki wykres, na którym można znaleźć zera funkcji.
- b) Drugą metodą jest użycie algorytmu, który wyznaczy przedział zawierający jeden pierwiastek. Zadając mu jakiś mały przedział początkowy algorytm bada iloczyn wartości 4 funkcji na krańcach przedziału i sukcesywnie go zmienia do momentu znalezienia zera funkcji. Warunkiem istnienia co najmniej jednego zera w przedziale jest prawdziwość:

$$f(a) * f(b) < 0$$

Proponowane rozwiązanie nie jest idealne, jednak pozwala na szybkie oszacowanie przedziału.

- c) 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.



Wykres z zaznaczonymi przedziałami startowymi

Metoda bisekcji:

1) Opis metody:

Jest to metoda iteracyjna pozwalająca wyznaczyć zero funkcji. Mając dany przedział ,gdzie a i b spełniają równanie:

$$f(a) * f(b) < 0, \text{ war. 1}$$

dzielimy go w połowie aby otrzymać przedziały i . Sprawdzamy w, którym przedziale znajduje się zero za pomocą war. 1 i ten przedział staje się nowym przedziałem, wchodzącym w kolejną iterację algorytmu. Startujemy z początkowego przedziału $[a,b] = [a_0,b_0]$ izolacji pierwiastka. W metodzie bisekcji, w każdej n -tej iteracji:

- A. Bieżący przedział zawierający zero funkcji $[a_n, b_n]$ jest dzielony na dwie połowy, punktem środkowym c_n i obliczana jest wartość funkcji w nowym punkcie.
- B. Obliczane są iloczyny:

$$f(a) * f(c_n) \text{ oraz } f(c) * f(b_n)$$

nowy przedział zawierający pierwiastek wybierany jest jako ten z dwóch podprzedziałów, któremu odpowiada iloczyn ujemny. Procedura jest powtarzana tak długo, aż np. $f(c_n) < \delta$, gdzie δ to założona dokładność. Dla precyzji można również sprawdzać także długość przedziału a, b .

2) Dokładność metody:

Dokładność rozwiązania zależy jedynie od ilości wykonanych iteracji, a nie zależy od dokładności obliczania wartości funkcji $f(x)$ na krańcach kolejnych przedziałów izolacji pierwiastka.

3) Zbieżność metody:

Niech ε_i oznacza długość przedziału w i -tym kroku (iteracji). Mamy:

$$\begin{aligned}\varepsilon_{i+1} &= \frac{1}{2} * \varepsilon_i, \\ \varepsilon_n &= 2^{-n} * \varepsilon_0\end{aligned}$$

Stąd startując z przedziału o szerokości ε_0 i oznaczając przez n ilość iteracji potrzebną do uzyskania dokładności ε_n dostajemy:

$$n = \log_n\left(\frac{\varepsilon_0}{\varepsilon_n}\right)$$

Metoda bisekcji jest więc zbieżna liniowo ($p=1$), ze współczynnikiem $k = \frac{1}{2}$, co czyni ją stosunkowo wolno zbieżną w przypadku wyboru szerokiego przedziału początkowego.

Implementacja metody bisekcji:

```
%funkcja wyznaczajaca zera funkcji metoda bisekcji
%Dane wejsciowe:
%      l,r - lewa i prawa sterona przedzialu poszukiwan
%      fun - funkcja
%      iter - maksymalna liczba iteracji
%Dane wyjsciowe: zerospoint - wyznaczone miejsce zerowe
function bzeropoint = bisection(fun,l,r,iter)
    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    [%12.10f;%12.10f]    %12.16f    %12.3e\n',k,a,b,xm, 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
        if(fm == 0) %dodatkowy warunek zakonczenia wykonywania
            return
        end
    end
    bzeropoint = xm;
    return
end
```

4) Wyniki:

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

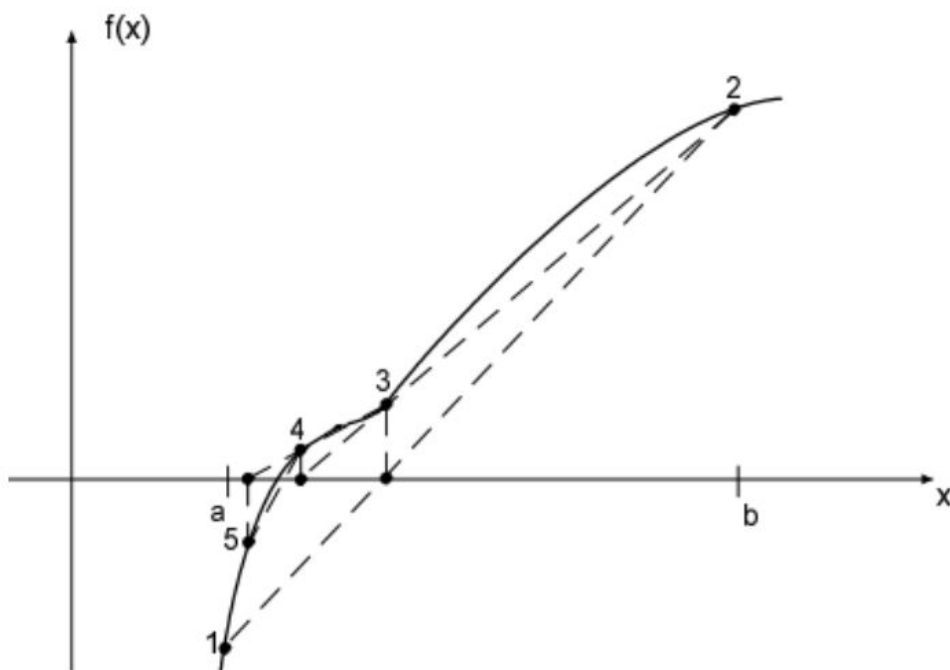
Metoda bisekcji drugie miejsce zerowe

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

Metoda siecznych:

1) Opis metody:

Metoda ta polega na przeprowadzeniu siecznych pomiędzy końcami przedziału ostatnio wyznaczonego. Jak na rysunku:



Jeżeli te dwa punkty oznaczmy przez x_{n-1} i x_n , to nowy punkt w metodzie tej jest zdefiniowany wzorem:

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

2) Zbieżność metody:

Rząd zbieżności: $p = (1 + \sqrt{5})/2 \approx 1.618$. Oznacza to, że metoda ta jest szybsza od metody bisekcji. Jest jednak ona zbieżna tylko lokalnie. Może być zawodna jeśli przedział izolacji zera funkcji jest zbyt mały. Podsumowując, jest ona dużo szybsza od metody bisekcji, jednak jest zbieżna jedynie lokalnie. Jeżeli nie zadbamy o wybór odpowiedniego przedziału początkowego może okazać się w ogóle nie zbieżna.

Implementacja metody siecznych:

```
%funkcja obliczająca zera funkcji metoda siecznych
%Dane wejściowe:
%l,r - lewa i prawa sterona przedziału poszukiwan
%      fun - funkcja
%      iter - maksymalna liczba uteracji
%Dane wyjściowe: zerospoint - wwyznaczone miejsce zerowe
function zerospoint = secant(fun,l,r,iter)
    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;
        zerospoint = b;
        fprintf('%3d      [%12.10f;%12.10f]      %12.16f      %12.3e\n',k,a,b,xm,fb);
        if(fb == 0) %dodatkowy warunek zakonczenia wykonywania
            return
        end
    end
end
```

3) 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:

1) Opis metody:

Metoda Newtona, zwana też metodą stycznych, zakłada aproksymację funkcji jej liniowym przybliżeniem wynikającym z uciętego rozwinięcia w szereg Taylora w aktualnym punkcie x_n (aktualnym przybliżeniem pierwiastka).

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

Następny punkt x_{n+1} wynika z przyrównania do zera sformułowanej lokalnej liniowej aproksymacji funkcji $f(x)$, tzn. z równania:

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

2) Zbieżność metody:

Metoda jest zbieżna lokalnie. Jeżeli zastosujemy ją w punkcie zbytnio oddalonym od rozwiązania, to może być ona rozbieżna. Natomiast metoda Newtona jest lokalnie bardzo szybka, jej zbieżność jest kwadratowa ($p=2$).

Implementacja metody Newtona:

```
%funkcja obliczająca zera funkcji metoda Newtona
%Dane wejściowe:
%           l - prawa strona przedziału poszukiwan
%           fun - funkcja
%           iter - maksymalna liczba iteracji
%Dane wyjściowe: zerospoint - wyznaczone miejsce zerowe
function nzeropoint = newton(fun,l,iter)
    x0 = l;
    for k = 1:iter
        [fold, fpold] = feval(fun,x0); |
        dx = fold / fpold; %wyznaczenie przyrostu funkcji
        x0 = x0 - dx;
        fprintf('%3d    %12.10f    %12.16f    %12.3e \n',k,dx,x0,fold);
        if(fold == 0)
            return
        end
        if fold==0 %dodatkowy warunek zatrzymania
            nzeropoint = x0;
            break;
        end
    end
end
end
```

3) Wyniki:

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

Wnioski:

Analizując wykres funkcji danej w zadaniu możemy wstępnie określić miejsca zerowe 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 osiągnąć 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 nie zbieżnością. 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 wada jest niezbieżność w przypadku gdy sieczna przecina wykres funkcji w większej ilości miejsc niż 2. Dlatego należy wybierać

wąskie przedziały startowe co zostało uczynione w rozwiązaniu. Ostatnia wypróbowana metoda 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 metoda 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.

Zadanie 2

Używając metody Müllera MM2 znaleźć wszystkie pierwiastki rzeczywiste i zespolone wielomianu:

$$f(x) = a_4x^4 + a_3x^3 + a_2x^2 + a_1x^1 + a_0,$$

$$[a_4 \ a_3 \ a_2 \ a_1 \ a_0] = [1 \ 3 \ -8 \ 4 \ 2]$$

Metoda Müllera:

1) Opis metody:

Metoda Müllera polega na aproksymacji wielomianu w otoczeniu rozwiązania funkcją kwadratową. Może być traktowana jako uogólnienie metod siecznych – zamiast interpolacji w dwóch punktach funkcją liniową wykonujemy interpolację w trzech punktach funkcją kwadratową. Metoda MM2 to realizacja oparta na wykorzystaniu informacji o wielomianie jedynie w jednym punkcie, tzn. wykorzystująca do wyznaczenia funkcji kwadratowej wartości wielomianu i jego pierwszej i drugiej pochodnej w aktualnym punkcie. Wersja MM2 jest efektywniejsza obliczeniowo z powodu, że obliczanie wartości wielomianu w $k+1$ punktach jest kosztowniejsze niż obliczenie wartości wielomianu i jego k kolejnych pochodnych w jednym punkcie.

Z definicji paraboli $y(x)$ dla $z = x - x_k$, w punkcie $z = 0$ wynika:

$$y(0) = c = f(x_k)$$

$$y'(0) = b = f'(x_k)$$

$$y''(0) = 2a = f''(x_k)$$

co prowadzi do wzoru:

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

Do przybliżenia zera α bierzemy pierwiastek paraboli o mniejszym module:

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

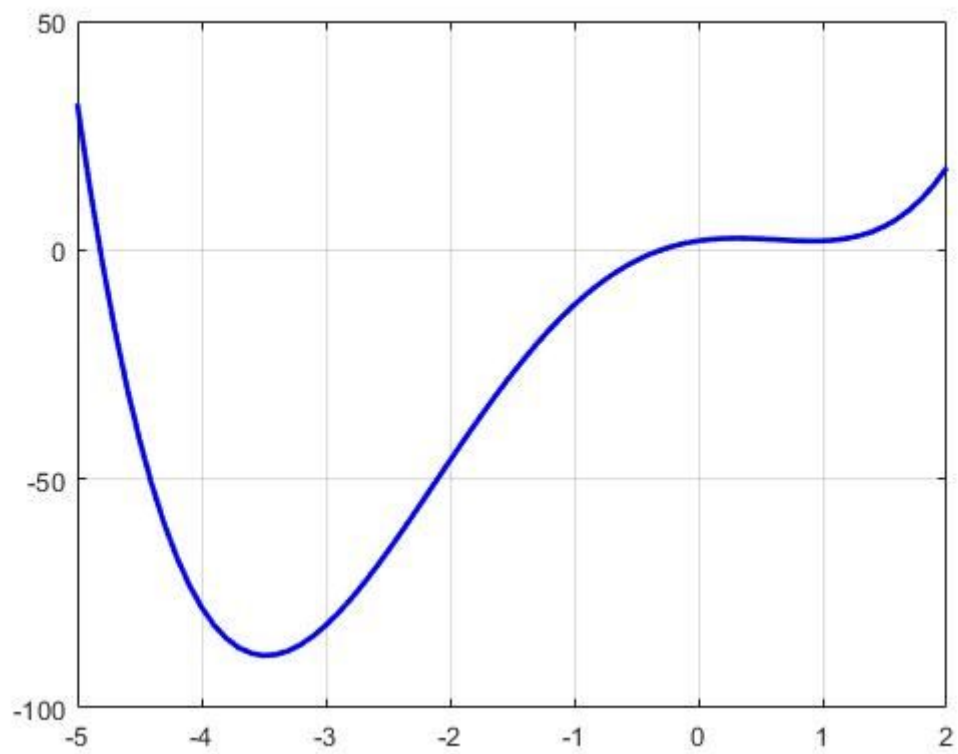
```
%Funkcja zwracająca wektor pierwiastków rzeczywistych wielomianu
%danego w zadaniu
%x - punkt startowy
%n - ilość iteracji
function [z] = muller(x,n)
    %z = (1:n);
    for i = 1:n
        %obliczamy mianowniki punktów w celu ich porównania
        z1 = df(x,1)+sqrt(df(x,1)^2-2*df(x,0)*df(x,2));
        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;
        fprintf('%d      %f      %f\n ',i,x,zmin);
    end
    z = x;
end
```

Metoda Mullera jest zbieżna lokalnie. Jest więc lokalnie bardziej efektywna niż metoda siecznych, jest niewiele wolniejsza od metody Newtona. Z konstrukcji metody wynika, że może ona być stosowana do poszukiwania zer rzeczywistych i zespolonych nie tylko wielomianów, ale i innych funkcji nieliniowych (analitycznych).

2) *Wyniki:*

Wykres wielomianu:



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

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.