

# MNUM-PROJEKT, zadanie 3.2

Tomasz Jakubczyk

1. Znajdowanie zer funkcji  $f(x)=0.55*x*\sin(x)-\ln(x+2)$  metodami: bisekcji, siecznych i Newtona.

Wszystkie trzy metody dobrze działają tylko lokalnie i znajdują raczej jeden pierwiastek naraz. W celu zapewnienia poprawnego działania należy im podać punkt lub przedział startowy odpowiednio bliski miejscu zerowemu. Można podać ręcznie punkt lub przedział. Jeśli przedział jest zbyt mały można go rozszerzać aż znajdzie się przedział izolacji (a, b) taki że  $f(a)*f(b)<0$ , czyli w tym przedziale funkcja na pewno przechodzi przez zero. Postanowiłem na moim zadanym przedziale funkcji sprawdzać przedziały z krokiem 1 ze względu na małą dynamiczność funkcji w takim przedziale aby automatycznie sprawdzać, czy jest to przedział izolacji. Takie rozwiązanie jest proste i skuteczne jeśli wie się jak ta funkcja wygląda, ale ma też takie wady, że jeśli funkcja jest szybko zmienna, to w zbyt dużym przedziale może być parzysta liczba przejść przez zero i metoda odrzuci taki obszar. W moim przypadku najpierw wyrysowałem wykres funkcji na przedziale i odpowiednio dobrałem krok.

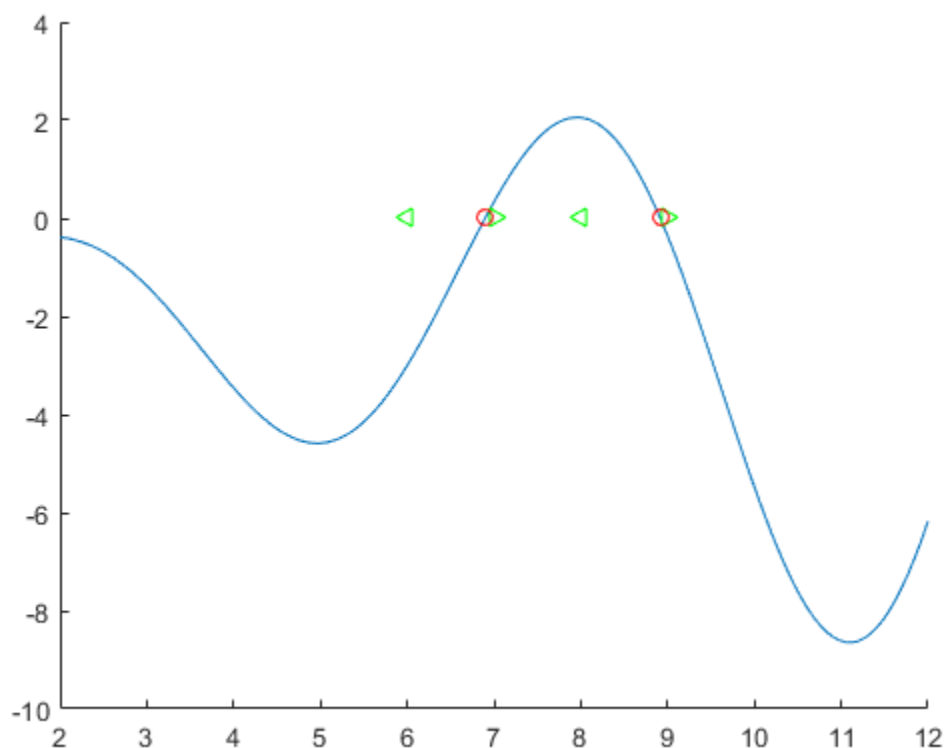
```
step=1;%krok szukania izolacji pierwiastka
jb=0;
for i=a:step:b-step
    p1=i;
    p2=i+step;
    if f(p1)*f(p2)<0 %izolacja
        plot(ax,p1,0,'<g');
        plot(ax,p2,0,'>g');
        [X0,Xi]=M_Bisekcji(p1,p2,f);
        jb=jb+1;
        XB(jb)=X0;
        plot(ax,X0,f(X0),'or');
    end
end
```

a) Metoda bisekcji – podaje się jej przedział startowy, w którym funkcja przechodzi przez zero tylko raz. Polega na dzieleniu przedziału na pół i wybieraniu jako nowego przedziału tej połówki, w której funkcja przechodzi przez zero (sprawdzenie  $f(a)*f(b)<0$ ). Warunkiem stopu jest znalezienie punktu, gdzie wartość funkcji jest odpowiednio bliska zeru, warunek ten jest uzupełniony warunkiem sprawdzającym, że przedział jest odpowiednio wąski, na wypadek gdyby funkcja miała małą pochodną (była płaska). Metoda bisekcji jest zbieżna liniowo.

```

function [ X, Xiteracje ] = M_Bisekcji( a0, b0, f )
%M_Bisekcji znajdowanie pierwiastków metodą bisekcji
%   a0, b0 - przedział
%   f - funkcja
%   X - miejsce zerowe
%   Xiteracje - punkty w kolejnych iteracjach
a=a0;
b=b0;
delta=0.0001;%dokładność rozwiązania
epsilon=0.0001;%długość przedziału
fn=f(a);
c=0;
j=0;
while abs(fn)>delta || abs(b-a)>epsilon
    c=(a+b)/2;
    fn=f(c);
    j=j+1;
    Xiteracje(j,1)=j;
    Xiteracje(j,2)=c;
    Xiteracje(j,3)=fn;
    fac=f(a)*f(c);
    fcb=f(c)*f(b);
    if fac<0
        b=c;
    elseif fcb<0
        a=c;
    else
        disp('metoda nie znajduje pierwiastka w przedziale');
    end
end
X=c;
end

```



Wykres funkcji  $f(x)$  z p. I. zielone trójkąty - wybrane przedziały, czerwone kółka - pierwiastki

Tabela punktów w iteracjach metody bisekcji dla przedziału [6, 7]

Iteracja	x	f(x)
1	6,500000000000000	-1,37101220608233
2	6,750000000000000	-0,498265076458980
3	6,875000000000000	-0,0737999297676115
4	6,937500000000000	0,131993062801295
5	6,906250000000000	0,0297364807756746
6	6,890625000000000	-0,0218794740025672
7	6,898437500000000	0,00396753297081443
8	6,894531250000000	-0,00894633393628519
9	6,896484375000000	-0,00248697623224325
10	6,897460937500000	0,000740886319805068
11	6,896972656250000	-0,000872893204562519
12	6,897216796875000	-6,59654749668448e-05
13	6,897338867187500	0,000337469917959243

14	6,89727783203125	0,000135754594920101
15	6,89724731445313	3,48951532753716e-05

b) Metoda siecznych – polega na znajdowaniu kolejnych punktów przez wyznaczenie siecznej między dwoma ostatnio wybranymi punktami i wzięciu jako nowy punkt przecięcie siecznej z zerem. Dwa ostatnie punkty nie muszą być przedziałem izolacji pierwiastka. Rząd zbieżności tej metody to 1.618. Metoda jest zbieżna tylko lokalnie. Może być niebezpieczna, jeśli początkowy przedział izolacji jest niewystarczająco mały.

```
function [ X, Xiteracje ] = M_siecznych( a0, b0, f )
%M_siecznych znajdowanie pierwiastków metodą siecznych
% a0, b0 - przedział
% f - funkcja
% X - miejsce zerowe
% Xiteracje - punkty w kolejnych iteracjach
delta=0.0001;%dokładność rozwiązania
epsilon=0.0001;%długość przedziału
fn=0;
j=0;
xnm1=a0;
xn=b0;
while abs(fn)>delta || abs(xn-xnm1)>epsilon
    xnp1=(xnm1*f(xn)-xn*f(xnm1))/(f(xn)-f(xnm1));
    xnm1=xn;
    xn=xnp1;
    fn=f(xn);
    j=j+1;
    Xiteracje(j,1)=j;
    Xiteracje(j,2)=xn;
    Xiteracje(j,3)=fn;
end
X=xn;
end
```

Tabela punktów w iteracjach metody siecznych dla przedziału [6, 7]

Iteracja	x	f(x)
1	6,90035840910979	0,0103108195418917
2	6,89716640991636	-0,000232497023642697
3	6,89723679861306	1,40424634587077e-07

Dla dużego przedziału, którego normalnie udaje mi się uniknąć za pomocą odpowiedniego wyszukiwania przedziałów izolacji pierwiastków, metoda siecznych zawiodła wychodząc poza zadany przedział i zaczęła zwracać liczby zespolone.

$[X_0, X_i] = M\_siecznych(2, 12, f);$

Iteracja	x	f(x)
1,000000000000000 + 0,000000000000000i	1,33372032068152 + 0,000000000000000i	-0,491060838344121 + 0,000000000000000i
2,000000000000000 + 0,000000000000000i	0,413093765119997 + 0,000000000000000i	-0,789700761354086 + 0,000000000000000i
3,000000000000000 + 0,000000000000000i	2,84752880154165 + 0,000000000000000i	-1,12453250107323 + 0,000000000000000i
4,000000000000000 + 0,000000000000000i	-5,32852201880128 + 0,000000000000000i	-3,59431840653317 - 3,14159265358979i
5,000000000000000 + 0,000000000000000i	4,26948110640401 - 1,80873771480963i	-7,25738466698135 + 6,08549028020200i
6,000000000000000 + 0,000000000000000i	-4,60758701722518 - 4,64084146674700i	-118,439639151070 - 143,196764923741i
7,000000000000000 + 0,000000000000000i	4,38732528974562 - 2,26798088385003i	-11,1729497708077 + 9,79395863598575i
8,000000000000000 + 0,000000000000000i	4,49764007604491 - 2,99943895710214i	-22,7669598584172 + 21,9284120495968i
9,000000000000000 + 0,000000000000000i	4,23284530974811 - 1,63158048489573i	-6,32655214139196 + 5,00917626879413i
10,000000000000000 + 0,000000000000000i	4,08235974172648 - 1,17939396209867i	-4,49030632677124 + 3,07231056727665i
11,000000000000000 + 0,000000000000000i	3,58848535982172 - 0,342781542535645i	-2,56671366764430 + 0,769685094950975i
12,000000000000000 + 0,000000000000000i	2,80873085269891 + 0,0377510450726292i	-1,06460067947827 - 0,0561911876334027i
13,000000000000000 + 0,000000000000000i	2,27189284150440 - 0,0168799062393888i	-0,497009222430974 + 0,0104624178869327i
14,000000000000000 + 0,000000000000000i	1,80281883952784 + 0,000262723983877388i	-0,370762451484475 + 1,16375338155165e-05i
15,000000000000000 + 0,000000000000000i	0,430473450195371 - 0,0629529741418101i	-0,791407332680411 - 0,00213552476049003i
16,000000000000000 + 0,000000000000000i	3,01267826201982 + 0,0497683114545174i	-1,39738772867633 - 0,0882190751361746i
17,000000000000000 + 0,000000000000000i	-2,89654160841544 + 0,253358337878704i	0,504378464823996 - 2,50530739888338i

18,00000000000000 + 0,000000000000000i	1,40939116636676 - 2,11248393580467i	1,05977079644697 - 4,76328895711271i
19,00000000000000 + 0,000000000000000i	-7,51414485143798 + 3,05228457307707i	33,5741255923999 - 33,9666643082699i
20,00000000000000 + 0,000000000000000i	1,88516336201477 - 3,16078083761550i	16,3681754147475 - 15,0765629872380i
21,00000000000000 + 0,000000000000000i	10,5136584917657 - 7,83431869764705i	-3951,63541161169 + 8207,17777209899i
22,00000000000000 + 0,000000000000000i	1,90891150549774 - 3,16363417730076i	16,9358383643647 - 14,6581230717231i
23,00000000000000 + 0,000000000000000i	1,93283538665714 - 3,16578436511589i	17,4832262924894 - 14,2028913442357i
24,00000000000000 + 0,000000000000000i	1,85304776896055 - 2,41000891125097i	6,01792374323296 - 5,01785595401848i
25,00000000000000 + 0,000000000000000i	1,81840214553195 - 2,00615209860414i	3,19494862196169 - 2,67005402534010i
26,00000000000000 + 0,000000000000000i	1,78020372412744 - 1,54808035856574i	1,34368879702104 - 1,20114192310719i
27,00000000000000 + 0,000000000000000i	1,77161635039917 - 1,19763063523102i	0,547649007990295 - 0,566379063630098i
28,00000000000000 + 0,000000000000000i	1,79992709813991 - 0,927847935703747i	0,169435237643234 - 0,247332314658573i
29,00000000000000 + 0,000000000000000i	1,85997159280054 - 0,774851037150794i	0,0233424963667552 - 0,0899041675263488i
30,00000000000000 + 0,000000000000000i	1,91421098808745 - 0,728908393563316i	-0,00775547867656523 - 0,0162951910919623i
31,00000000000000 + 0,000000000000000i	1,93010450894031 - 0,731167148404588i	-0,00137741027608396 + 0,000185950803130636i
32,00000000000000 + 0,000000000000000i	1,93022286618894 - 0,732424160277518i	1,29170964733305e-05 + 1,72919537762828e-05i
33,00000000000000 + 0,000000000000000i	1,93020515520247 - 0,732416102318522i	-1,93488316480028e-08 - 1,66289001735098e-08i

c) Metoda Newtona – funkcję aproksymuje się liniowo w punkcie za pomocą uciętego rozwinięcia w szereg Taylora.

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

Po przekształceniu tego wzoru wychodzi iteracyjna zależność:

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

punkt ten jest przecięciem z zerem poprzedniej liniowej aproksymacji.

Metoda Newtona jest zbieżna lokalnie, w punkcie niezbyt oddalonym od pierwiastka może być rozbieżna. Metoda Newtona jest zbieżna kwadratowo. Możliwe testy stopu to:

1. odpowiednia bliskość zera

$$|f(x_k)| \leq \text{eps}$$

2. odpowiednio mała odległość między kolejnymi przybliżeniami

$$|x_{k+1} - x_k| \leq \text{eps}$$

3. szacowany błąd odpowiednio mały

4. kryterium mieszane 1 i 2 (to realizuję)

W metodzie Newtona potrzebna jest pochodna funkcji w punkcie, postanowiłem w metodzie podawać analitycznie wyliczoną pochodną, można to zrobić ręcznie lub korzystając z toolboxa Symbolic, oraz na wiele innych sposobów.

```
function [ X, Xiteracje ] = M_Newtona( a0, b0, f, df )
%M_Newtona znajdowanie pierwiastków metodą Newtona
% a0, b0 - przedział
% f - funkcja
% df - pochodna funkcji
% X - miejsce zerowe
% Xiteracje - punkty w kolejnych iteracjach
epsilon=0.0001;%długość przedziału
xk=b0;
xkml=a0;
j=0;
while abs(f(xk))>epsilon || abs(xk-xkml)>epsilon
    xkp1=xk-(f(xk)/df(xk));
    xkml=xk;
    xk=xkp1;
    fn=f(xk);
    j=j+1;
    Xiteracje(j,1)=j;
    Xiteracje(j,2)=xk;
    Xiteracje(j,3)=fn;
end
X=xk;
end
```

Tabela punktów w iteracjach metody Newtona dla przedziału [6, 7] (punkt x=7)

Iteracja	x	f(x)
----------	---	------

1	6,89464014586213	-0,00858606739340928
2	6,89723546499602	-4,26717416956990e-06
3	6,89723675612413	-1,06181730075150e-12

Metoda bisekcji jest bardzo solidną i prostą metodą, zawsze jest zbieżna dla przedziału izolacji pierwiastka, nawet jeśli w przedziale było by więcej niż jeden pierwiastek, to na pewno jakiś znajdzie. Niestety jest wolniej zbieżna niż metoda siecznych, która jest wolniej zbieżna niż metoda Newtona. Metoda Newtona może mieć punkt początkowy poza kulą zbieżności i dodatkowo potrzebuje pochodnej co stanowi utrudnienie.

2. Znajdowanie pierwiastków rzeczywistych i zespolonych metodą Müllera MM2 wielomianu  $f(x)=a_4x^4+a_3x^3+a_2x^2+a_1x+a_0$   $a=[-1 \ 1.5 \ 1.5 \ 0.5 \ 1]$

Metoda Müllera polega na aproksymacji wielomianu funkcją kwadratową w otoczeniu rozwiązania. MM2 wykorzystuje wartość, pierwszą i drugą pochodną w punkcie. Z pochodnych można bezpośrednio uzyskać wartości współczynników funkcji kwadratowej,

$$\begin{aligned}y''(0) &= 2a = f''(x_k), \\y'(0) &= b = f'(x_k), \\y(0) &= c = f(x_k),\end{aligned}$$

a ta daje wzór na pierwiastki.

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

W następnym kroku wybiera się pierwiastek o mniejszym module

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

Metoda Müllera jest zbieżna lokalnie i ma rząd zbieżności 1.84.

```
function [ X, Xiteracje ] = MM2( x,f,df1,df2 )
%MM2 metoda Mulllera MM2
% x - punkt startowy
% f - funkcja
% df1 - pierwsza pochodna funkcji
% df2 - druga pochodna funkcji
xn=x;
j=0;
for i=1:10%ograniczenie na liczbe iteracji
p=sqrt((df1(xn))^2 - 2* f(xn)*df2(xn));
if abs(df1(xn)+p) > abs(df1(xn)-p)%wybór pierwiastka o mniejszym module
z=(-2*f(xn))/(df1(xn)+p);
else
z=(-2*f(xn))/(df1(xn)-p);
```



```

end
xn=xn+z;
fn=f(xn);
j=j+1;
Xiteracje(j,1)=j;
Xiteracje(j,2)=xn;
Xiteracje(j,3)=fn;
if abs(fn)<0.0001
    break;
end
end
X=xn;
end

```

Znalezione rozwiązania, to:

$$X_0 = -0.9035$$

$$X_0 = 2.3197$$

$$X_0 = 0.0419 + 0.6895i$$

$$X_0 = 0.0419 - 0.6895i$$

Rozwiązania bardzo zależą od punktu początkowego, może się zdarzać, że metoda w poszukiwaniu rozwiązań od jednego punktu startowego znajdzie więcej niż jedno rozwiązanie. Z tego powodu istotne są kryteria ocenianie, czy wartość funkcji jest odpowiednio bliska zeru. Ułatwieniem może być informacja o stopniu wielomianu, dzięki temu można np. wybrać odpowiednią liczbę najlepszych rozwiązań. Zespólna dziedzina utrudnia przeszukiwanie rozwiązań po kolejnych punktach.

Tabela iteracji algorytmu MM2 dla punktu startowego  $0.04+0.6i$

Iteracja	x	f(x)
1,0000000000000000 + 0,0000000000000000i	0,0418654087770215 + 0,690009344082850i	-0,00187410336001692 - 0,000861668647784808i
2,0000000000000000 + 0,0000000000000000i	0,0419160907831130 + 0,689474956327550i	3,52068596498611e-10 + 3,17315507203375e-10i