

MNUM Projekt 3.

Autor:	Numer projektu:	Prowadzący:	Liczba punktów:
Mariusz Słapek	3.42	dr inż. Adam Krzemienowski	

Kody do poszczególnych programów zostały umieszczone na końcu sprawozdania.

Spis treści

Zadanie 1	3
Treść:	3
Ogólny zarys:	3
Wykres funkcji	3
Podstawowe informacje	4
Metoda bisekcji	4
Metoda regula falsi	5
Metoda siecznych	5
Metoda Newtona (stycznych)	6
Prezentacja otrzymanych wyników:	8
Przedziały liczbowe dość wąskie:	8
Przedziały liczbowe dość szerokie:	9
Sytuacja w której metoda stycznych zawodzi	12
Zadanie 2	14
Treść:	14
Ogólny zarys:	14
Podstawowe informacje	14
Metoda Müllera MM2	14
Deflacja czynnikiem liniowym	15
Prezentacja otrzymanych wyników:	15
Wnioski:	15
Kod do programów	16
Zadanie 1:	16
Metoda bisekcji	16
Metoda siecznych	16
Metoda stycznych	17
Funkcja	17
Pochodna funkcji	17
Zadanie 2:	17

MM2.....	17
Deflacja czynnikiem liniowym	18
Funkcja	18
Pierwsza pochodna funkcji.....	18
Druga pochodna funkcji	18

Zadanie 1

Treść:

Proszę znaleźć wszystkie zera funkcji w dane przydziale używając dla każdego zera programu z implementacją:

- a) metody bisekcji,
- b) metody siecznych,
- c) metody Newtona.

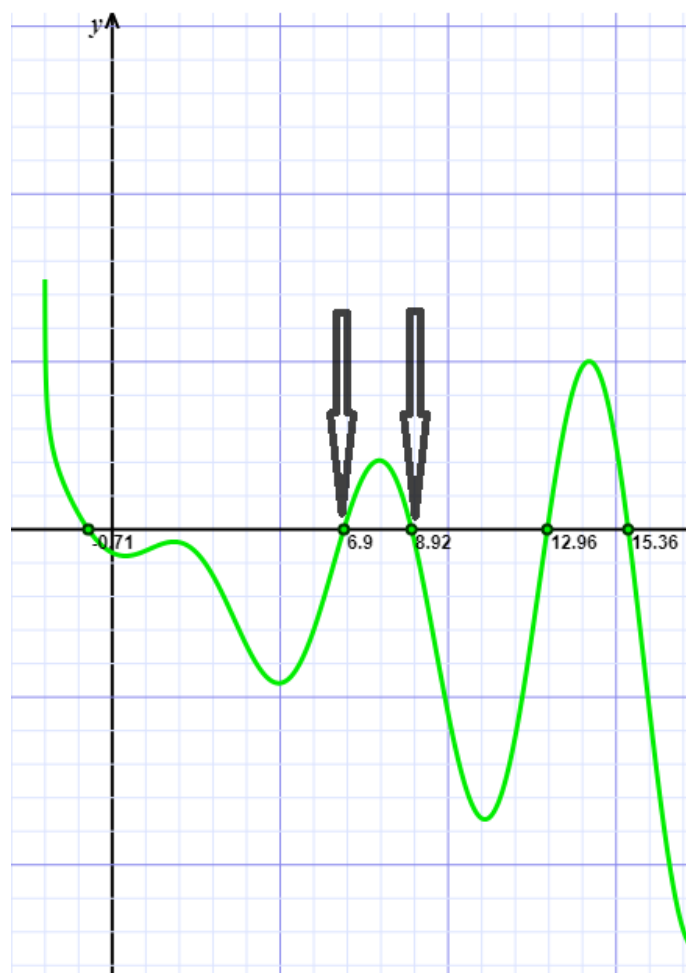
Ogólny zarys:

Wykres funkcji

$$f(x) = 0.55 \cdot x \cdot \sin(x) - \ln(x - 2)$$

UWAGA 1: strzałki wskazują gdzie znajdują się miejsca zerowe funkcji $f(x)$ w przedziale $[2, 12]$

UWAGA 2: liczby pod miejscami zerowymi są jedynie pewnym przybliżeniem miejsc zerowych.



Rysunek 1 Wykres funkcji z zaznaczonymi miejscami zerowymi

Podstawowe informacje

Często istotnym zagadnieniem jest wyznaczanie zer równania nieliniowego $f(x) = 0$. W celu wyznaczenia takiego miejsca zerowego, najpierw musimy oszacować przedział, w którym znajduje się rozwiązanie (jest to tzw. *przedział izolacji pierwiastka*). Przedział ten możemy odczytać z uproszczonego wykresu funkcji (albo narysowanego w programie komputerowym) lub na podstawie metody wyznaczania przedziału izolacji. Podstawową metodą wyznaczenia tego przedziału jest badanie iloczynu wartości funkcji na końcach przedziału – jeśli ten iloczyn jest ujemny (a funkcja ta jest ciągła) wówczas w przedziale tym znajduje się co najmniej jeden pierwiastek.

Następnie szukamy wartości pierwiastka równania (mając dany przedział izolacji pierwiastka).

Rząd metody iteracyjnej (inaczej wykładnik zbieżności) największa liczba p (większa albo równa 0) taka, że:

$$\lim_{n \rightarrow +\infty} \frac{|x_{n+1} - \alpha|}{|x_n - \alpha|^p} = k < \infty$$

k – współczynnik lub iloraz zbieżności.

Gdy p :

- i. $p = 1$ – metoda zbieżna liniowo
- ii. $p = 2$ – metoda zbieżna kwadratowo

Metody iteracyjne dla problemów nieliniowych są (przeważnie) zbieżne tylko lokalnie.

Istnieje wiele metod szukania *zer* funkcji:

- i. metoda bisekcji (połowienia)
- ii. metoda *regula falsi*
- iii. metoda siecznych
- iv. metoda Newtona (stycznych)

Metoda bisekcji

Dość naturalna metoda obliczeniowa zer skalarnych funkcji ciągłych określonych na danym przedziale $[a, b]$ i zmieniających znak (tzn. funkcja przyjmuje na końcu przedziałów wartości przeciwnego znaku). Na mocy twierdzenia *Darboux* wiemy, że jest przynajmniej jedno zero funkcji.

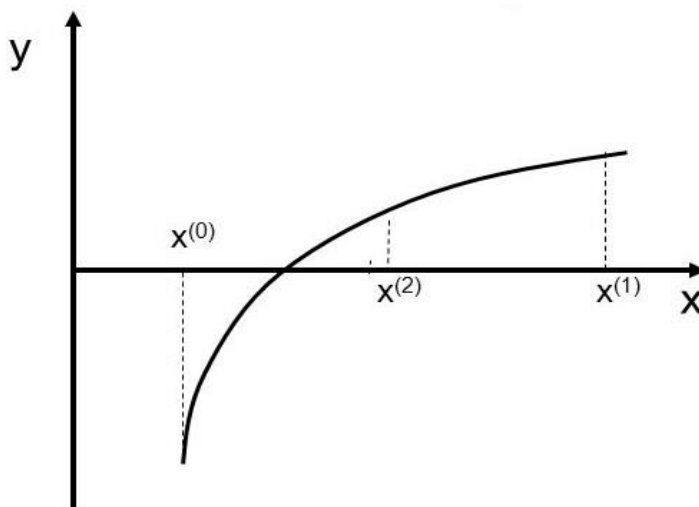
Ogólny algorytm wygląda następująco (i -ta iteracja):

- i. aktualny przedział zawierający zero funkcji $[a_i, b_i]$ jest dzielony na dwie połowy:

$$c_i = \frac{a_i + b_i}{2}$$

- ii. liczymy wartość funkcji w punkcie c_i
- iii. liczymy iloczyny $f(a_i) * f(c_i)$ i $f(c_i) * f(b_i)$
- iv. nowy przedziałem będzie ten podprzedział, gdzie odpowiada ujemna wartość funkcji

- na jego krańcach
- v. procedura jest postarzana tak długo, aż zostanie osiągnięta zakładana dokładność.



Rysunek 2 Metoda bisekcji

Dokładność rozwiązania zależy jedynie od ilości wykonanych iteracji, jest ona zbieżna liniowo). Jest to metoda zbieżna globalnie, co oznacza, że zawsze znajdziemy pierwiastek w danym przedziale, jeżeli ten tylko istnieje.

Metoda reguła falsi

Jest to metoda podobna do metody bisekcji. Różnica polega na tym, iż w metodzie tej przedziały dzielone nie zawsze są równe prostą – sieczną, która łączy na płaszczyźnie (f, x) punkty krańcowe przedziału.

Wybór następnego przedziału izolacji jest dokonywany analogicznie jak w metodzie bisekcji (przedział, którego iloczyn wartości funkcji w punktach krańcowych jest ujemny).

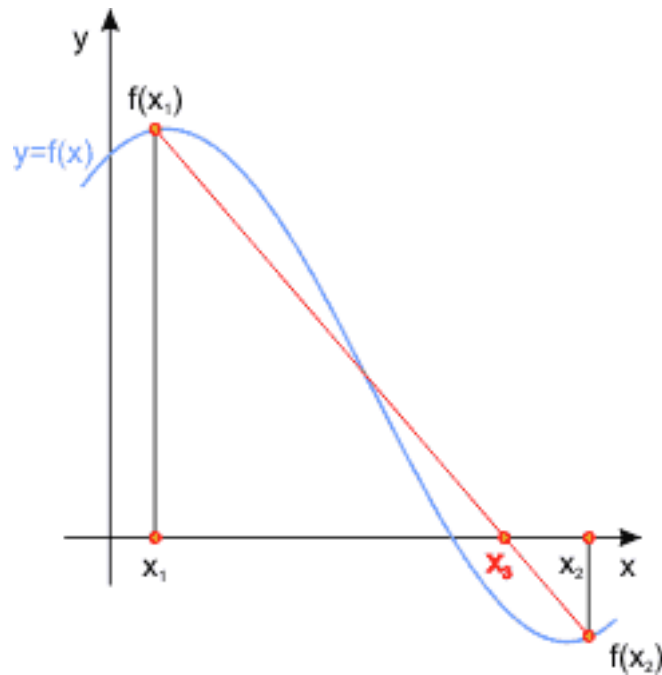
Metoda reguła falsi jest zawsze zbieżna liniowo, może być jednakże również wolno zbieżna (w ten celu wprowadza się pewną modyfikację – wówczas metoda zmodyfikowana reguła falsi – zbieżność superliniowa).

Wprowadziłem ją aby łatwiej było opisać metodę siecznych.

Metoda siecznych

Metoda ta różni się od metody reguła falsi, że sieczną prowadzimy zawsze między dwoma ostatnio wyznaczonymi punktami, tzn.:

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



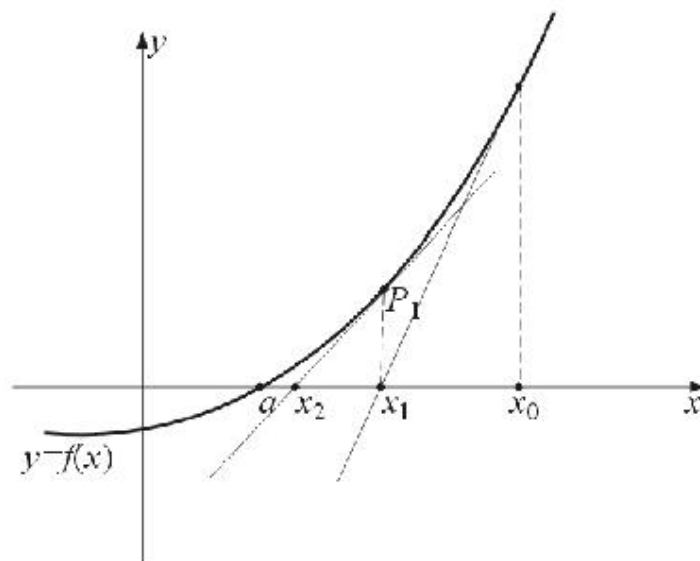
Rysunek 3 Metoda siecznych

Metoda ta jest szybsza od metody bisekcji i regula falsi. Jednakże, jest ona zbieżna tylko lokalnie, stąd w praktyce może być niezbieżna (przedział izolacji nie dostatecznie mały).

Metoda Newtona (stycznych)

Zakłada ona aproksymację funkcji jej liniowym przybliżeniem wynikającym z uciętego rozwinięcia w szereg Taylora w aktualnym punkcie x_i , a następnie przyrównania do zera sformułowanej lokalnej aproksymacji funkcji $f(x)$, co prowadzi do zależności iteracyjnej:

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



Rysunek 4 Metoda stycznych

Metoda Newtona jest zbieżna lokalnie (jeśli zaczniemy ją stosować w punkcie zbytnio oddalonym od rozwiązania, to może być ona rozbieżna). Jej zbieżność jest kwadratowa. Metoda stycznych jest szczególnie efektywna w przypadku, gdy krzywa jest bardzo stroma w otoczeniu danego pierwiastka (nie zaleca się stosowania, gdy krzywa jest w otoczeniu pierwiastka pozioma – innymi słowy pochodna w tym punkcie ma bardzo małą wartość).

Prezentacja otrzymanych wyników:

Przedziały liczbowe dość wąskie:



Rysunek 5 Zaznaczone punkty skrajne przedziału liczbowych dla przedziału wąskiego

Z zadaniu określę dwa przedziały (zaznaczone na wykresie powyżej):

- 1) $[2\pi; 2.5\pi]$
- 2) $[2.5\pi; 3\pi]$

ze względu na okresowość funkcji $\sin(x)$ i monotoniczność funkcji $\ln(x)$ (można odczytać z wykresu).

Przyjęta dokładność – $\epsilon = 0.00000000001$

	dla 1) przedziału			dla 2) przedziału		
Rodzaj metody:	bisekcja	siecznych	stycznych	bisekcji	siecznych	stycznych
Liczba iteracji:	38	6	4	40	7	4
Wynik:	6.897236756	6.897236756	6.897236756	8.91562544	8.915625441	8.915625441

Iteracje dla danej metody dla przedziału 1):

Iteracja	Metoda bisekcji	Metoda siecznych	Metoda stycznych
1	4.92699	7.0842	6.91713
2	6.39049	6.76826	6.89716
3	7.12223	6.90225	6.89724

4	6.75636	6.89734	6.89724
5	6.9393	6.89724	
6	6.84783	6.89724	
7	6.89356		
8	6.91643		
9	6.905		
10	6.89928		

Iteracje dla danej metody dla przedziału 2):

Iteracja	Metoda bisekcji	Metoda siecznych	Metoda stycznych
1	9.92699	8.56836	8.96268
2	8.89049	8.85142	8.91647
3	9.40874	8.92818	8.91563
4	9.14961	8.91528	8.91563
5	9.02005	8.91562	
6	8.95527	8.91563	
7	8.92288	8.91563	
8	8.90668		
9	8.91478		
10	9.92699		

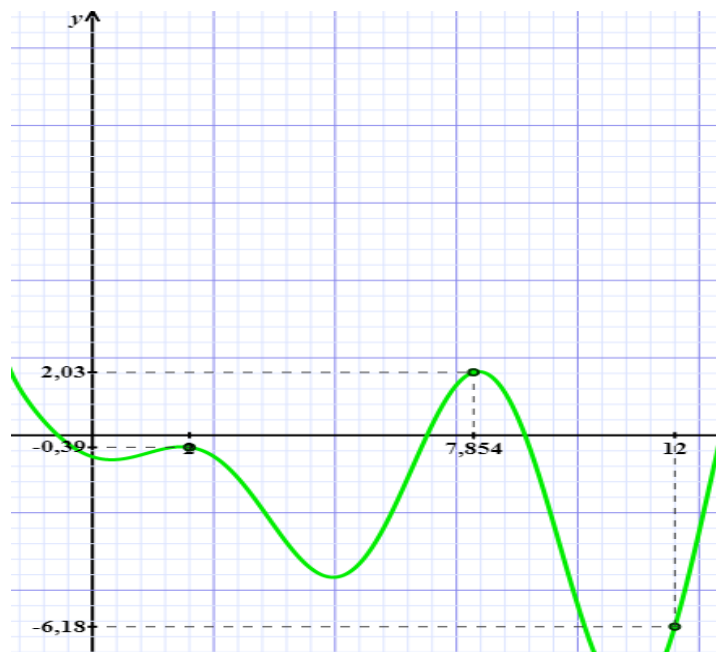
Dla $\epsilon = 0.0001$

Rodzaj metody:	dla 1) przedziału			dla 2) przedziału		
	bisekcja	siecznych	stycznych	bisekcji	siecznych	stycznych
Liczba iteracji:	14	5	3	15	5	3
Wynik:	6.89725699	6.8972366529	6.8972367548	8.915640149	8.915623713	8.915625728

Przedziały liczbowe dość szerokie:

Z zadaniu określę dwa przedziały dla metody bijekcji (zaznaczone na wykresie powyżej):

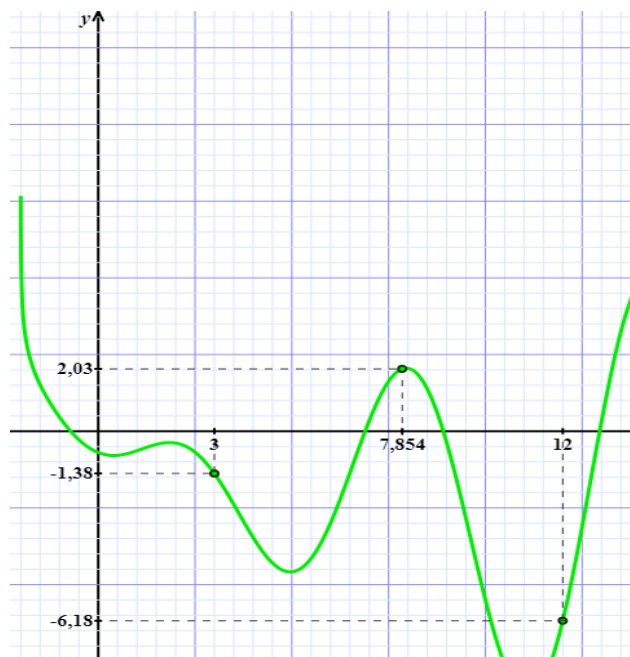
- 1) $[2; 2.5\pi]$
- 2) $[2.5\pi; 12]$



Rysunek 6 Zaznaczone punkty skrajne przedziałów metody bisekcji dla przedziału szerokiego

Z zadaniu określę dwa przedziały dla metody siecznych (zaznaczone na wykresie powyżej):

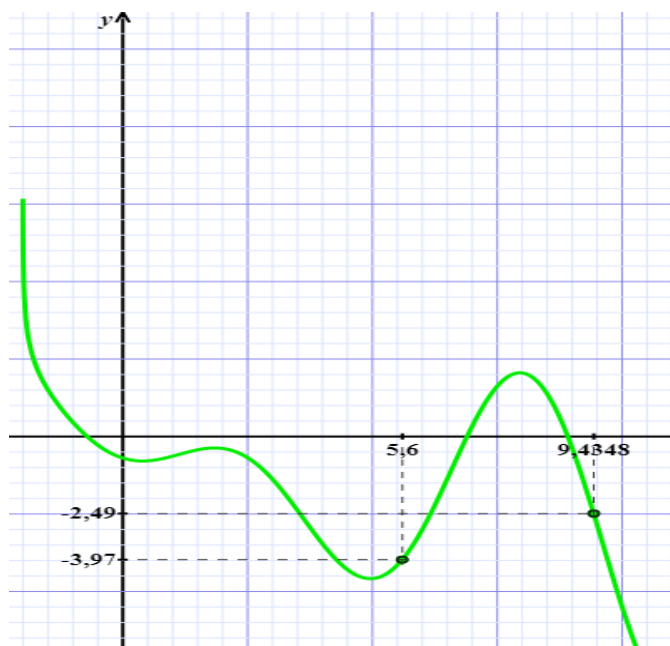
- 1) $[3; 2.5\pi]$
- 2) $[2.5\pi; 12]$



Rysunek 7 Zaznaczone punkty skrajne przedziałów metody siecznych dla przedziału szerokiego

Z zadaniu określę dwa przedziały dla metody stycznych (zaznaczone na wykresie powyżej):

- 1) 5
- 2) 3π



Rysunek 8 Zaznaczone punkty skrajne przedziałów metody stycznych dla przedziału szerokiego

Przyjęta dokładność – $\epsilon = 0.00000000001$

	dla 1) przedziału			dla 2) przedziału		
Rodzaj metody:	bisekcja	siecznych	stycznych	bisekcji	siecznych	stycznych
Liczba iteracji:	37	7	6	35	6	4
Wynik:	6.897236756	6.897236756	6.897236756	8.915625441	8.91562544	8.91562544

Iteracje dla danej metody dla przedziału 1):

Iteracja	Metoda bisekcji	Metoda siecznych	Metoda stycznych
1	7.06858	4.96043	7.67985
2	6.67588	6.96546	6.06471
3	6.87223	6.87274	6.99746
4	6.97041	6.89758	6.89478
5	6.92132	6.89724	6.89724
6	6.89678	6.89724	6.89724
7	6.90905	6.89724	
8	6.90291		
9	6.89985		
10	6.89831		

Iteracje dla danej metody dla przedziału 2):

Iteracja	Metoda bisekcji	Metoda siecznych	Metoda stycznych
1	8.63938	8.87976	8.96268
2	9.03208	8.95131	8.91647
3	8.83573	8.9151	8.91563

4	8.9339	8.91562	8.91563
5	8.88482	8.91563	
6	8.90936	8.91563	
7	8.92163		
8	8.9155		
9	8.91856		
10	8.91703		

Sytuacja w której metoda stycznych zawodzi

Gdy w metodzie stycznych dobierzemy punkt startowy dla którego pochodna jest bliska zeru, wówczas widzimy, że program nas zawodzi i otrzymujemy wartość pierwiastka, który nie jest położony w obszarze poszukiwań.

Dla punktu startowego 2.5π mamy:

Numer iteracji:	Wartość funkcji:
1	3.32392
2	2.36311
3	1.63359
4	3.44429
5	2.39633
6	1.68319
7	3.92544
8	2.34183
9	1.59938
10	3.22598
11	2.32694
12	1.57414
13	3.10028
14	2.26979
15	1.46528
16	2.75118
17	2.04959
18	0.618276
19	4.19551
20	2.04371
21	0.573311
22	5.00647
23	40.5213
24	40.6759
25	40.6721
26	40.6721

Widzimy, że funkcja znalazła całkowicie inny pierwiastek dla tak zadanego punktu startowego.

Wnioski:

Możemy zauważyć, że metody siecznych oraz Newtona są o wiele szybsze niż metoda bisekcji przy danej dokładności (liczba iteracji przy metodzie bisekcji jest zdecydowanie większa niż przy innych metodach).

Warto zauważyć, że metoda Newtona wymaga dość dobrze dobranego punktu startowego. Jeżeli funkcja przyjmuje w punkcie startowym pochodną bliską zeru, wówczas odnaleziony pierwiastek leży poza przeszukiwanym w zadaniu zakresem (doprowadzona przez mnie sytuacja).

Metoda siecznych ma również swoją wadę - jeżeli przedział izolacji pierwiastka jest zbyt mały, wówczas metoda nie jest często zbieżna.

Metoda bisekcji:

Wadą metody jest to, iż zbieżność ta nie jest imponująca.

Ma ona parę zalet jest: ona w pewien sposób uniwersalna, ma ona zbieżność globalną, wystarczy dla niej jedynie ciągłość funkcji.

Metoda siecznych:

Metoda siecznych może zawieść. Jeśli jest ona jedynie lokalni, stąd w praktyce może być niezbieżna – jeśli początkowy przedział izolacji pierwiastka nie jest dostatecznie mały. Ponadto, gdy żądanie przez użytkownika dokładności są bardzo wielkie, a sama funkcja „złośliwa”, metoda siecznych może cierpieć z powodu redukcji cyfr przy odejmowaniu.

Zaletą jest to metoda szybsza niż metoda bisekcji i reguła fałsi (rząd zbieżności to 1.618). Inną zaletą jest to, iż nie wymaga ona obliczenia pochodnej funkcji.

Metoda stycznych (Newtona):

Metoda stycznych może zawieść. Dzieje się to, gdy dla jednego z rozwiązań pochodna funkcji w tym punkcie istnieje i jest to bardzo mała liczba (pochodna bardzo bliska zera).

Wadą metody jest to, iż jest zbieżna lokalnie. Jeśli stosujemy ją w punkcie zbyt oddalonym od rozwiązania, to może być ona rozbieżna.

Istotną zaletą metody jest bardzo szybka (szczególnie efektywna, gdy krzywa jest bardzo stroma w otoczeniu danego pierwiastka).

Zadanie 2

Treść:

Używając metody Müllera *MM2*, proszę znaleźć wszystkie pierwiastki rzeczywiste i zespolone danego wielomianu

Ogólny zarys:

Podstawowe informacje

Wielomian stopnia n posiada dokładnie n pierwiastków (zer wielomianu):

- i. pierwiastki mogą być zarówno rzeczywiste oraz zespolone
- ii. pierwiastki mogą być pojedyncze lub wielokrotne

Do poszukiwania pierwiastków rzeczywistych możemy korzystać z metod wyznaczania przeszukiwania zer funkcji nieliniowej.

Jednak istnieją metody bardziej złożone, które są opracowane specjalnie dla wielomianów (wykorzystują właściwość – wielokrotna różniczkowalność).

Do metod tych należą:

- i. metoda Müllera (aproksymacja wielomianu funkcją kwadratową w otoczeniu rozwiązania – uogólniona metoda siecznych – *MM1*, wykorzystanie informacji o wielomianie jedynie w jednym punkcie, tzn. wykorzystująca do wyznaczenia funkcji kwadratowej wartości wielomianu i jego pierwszej i drugiej pochodnej w danym punkcie – *MM2*)
- ii. metoda Laguerre’a.

Metoda Müllera *MM2*

Ta wersja metody wykorzystuje informację o wartości wielomianu i jego pochodnych, pierwszego i drugiego w aktualnym punkcie (przybliżeniu zera). Wersja nieco efektywniejsza obliczeniowo z powodu, iż obliczenie wartości wielomianu w $k+1$ punktach jest kosztowniejsze niż obliczanie wartości wielomianów i jego k kolejnych pochodnych w jednym punkcie.

Wiemy, iż:

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

co prowadzi do wzoru na pierwiastki:

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

Do przybliżenia zera α bierzemy pierwiastek paraboli o mniejszym module:

$$x_{k+1} = x_k + Z_{\min}$$

Jest więc bardziej efektywna niż metoda siecznych (niewiele wolniejsza niż metoda stycznych).

Deflacja czynnikiem liniowym

Po znalezieniu pojedynczego pierwiastka α należy, przed wyznaczeniem następnego, uprościć wielomian dzieląc go przez czynnik $(x - \alpha)$ zawierający znaleziony pierwiastek. Uzyskany wielomian niższego rzędu jest już prostszy i nie wyznaczymy ponownie tego samego pierwiastka.

W zadaniu wykorzystałem deflację prostym schematem Hornera (zakładam, że α różna od 0):

$$q_{n+1} = 0$$
$$q_i = a_i + q_{i+1}\alpha$$

Prezentacja otrzymanych wyników:

Pierwszy pierwiastek	-0,903510312100570 + 0,000000000000000i
Drugi pierwiastek	0,0419160907353486 - 0,689474956441362i
Trzeci pierwiastek	0,0419160907353486 + 0,689474956441362i
Czwarty pierwiastek	2,31967813062987 + 0,000000000000000i

Wnioski:

Program po wykonaniu dał nam cztery miejsca zerowe wielomianu. Po sprawdzenie wyników w programie okazuje się, że uzyskane przeze mnie wyniki są poprawne.

Cechą wyróżniającą metodę MM2 jest to, iż znajduje ona również pierwiastki zespolone. Innym z plusów metody jest to, iż niezależnie od oddalenia od lokalnego punktu umożliwia ona odnalezienie pierwiastka.

Metoda Müllera jest zbieżna lokalnie, z rzędem 1.84. Jest więc (lokalnie) bardziej efektywna niż metoda siecznych, jest niewiele wolniejsza od metod Newtona. Z konstrukcji metody wynika, że może ona być stosowana do poszukiwania zer rzeczywistych i zespolonych nie tylko wielomianów, ale i innych funkcji nieliniowych.

Jedną z wad metody MM2 brak względnej skuteczności.

Kod do programów

Zadanie 1:

Metoda bisekcji

```
function result = bisection_method(a, b, eps)
    x = a;
    y = b;
    iter = 0; %zmienna, która powie nam ile zostało wykonanych iteracji
    war_z = 1;
    war_x = 0;
    war_y = 0;

    while (abs(war_z) > eps) %algorytm zatrzyma się, gdy zostanie uzyskana
określona dokładność
        war_x = fun(x); %obliczamy wartosc funkcji w punkcie x
        war_y = fun(y); %obliczamy wartosc funkcji w punkcie y
        z = (x+y)/2; %obliczamy srodek przedziału
        war_z = fun(z); %obliczamy wartosc funkcji dla srodka przedziału
        if (war_y * war_z < 0) %jesli iloczyn wartosci funkcji dla srodka
przedziału i wartosci w punkcie x jest ujemna wówczas...
            x = z; %przypisujemy x wartosc zmiennej z
        end
        if (war_x * war_z < 0) %jesli iloczyn wartosci funkcji dla srodka
przedziału i wartosci w punkcie y jest ujemna wówczas...
            y = z; %przypisujemy y wartosc zmiennej z
        end

        iter = iter + 1; %inkrementujemy liczbe iteracji

        fprintf(1, '%g \n', z);
    end
    result = z; %przypisujemy jako wynik wartosc zmiennej z
    fprintf(1, 'Liczba iteracji: %d \n', iter);
end
```

Metoda siecznych

```
function result = secant_method(a, b, eps)

    x = a;
    y = b;
    war_z = 2 * eps;
    iter = 0; %zmienna, która powie nam ile zostało wykonanych iteracji

    while (abs(war_z) > eps) %algorytm zatrzyma się, gdy zostanie uzyskana
określona dokładność
        war_x = fun(x); %obliczamy wartosc funkcji w punkcie x
        war_y = fun(y); %obliczamy wartosc funkcji w punkcie y

        z = (x*war_y-y*war_x)/(war_y - war_x); %obliczamy wartosc z z
danego wzoryu
    end
```



```

        war_z = fun(z); %obliczamy wartość funkcji dla obliczonej wartość
z
        x = y; %przypisujemy wartość y zmiennej x
        y = z; %przypisujemy wartość zmiennej y zmiennej z
        iter = iter + 1; %inkrementujemy wartość zmiennej iter
        fprintf(1, '%g\n', z);
    end
    result = z; %przypisujemy jako wynik wartość zmiennej z
    fprintf(1, 'Liczba iteracji jest równa: %d\n', iter);

end

```

Metoda stycznych

```

function result = newton_method(a,b, eps)
    x = a;
    y = b;
    iter = 0; %zmienna, która powie nam ile zostało wykonanych iteracji
    war_z = 1;

    while(abs(war_z) > eps) %algorytm zatrzyma się, gdy zostanie uzyskana
określona dokładność
        z = x - fun(x)/derivative(x); %obliczamy zmienną z z gotowego wzoru
        war_z = fun(z); %obliczamy wartość funkcji w punkcie z
        x = z; %przypisujemy pod zmienną x wartość zmiennej z
        iter = iter + 1; %inkrementujemy iter
        fprintf(1, '%g\n', z);
    end

    result = z; %przypisujemy jako wynik wartość zmiennej z
    fprintf(1, 'Liczba iteracji jest równa: %d\n', iter);

end

```

Funkcja

```

function y = fun(x)
    y = 0.55*x*sin(x) - log(x+2);
end

```

Pochodna funkcji

```

function y = derivative(x)
    y = 0.55*sin(x)+0.55*x*cos(x)- (1/(x+2));
end

```

Zadanie 2:

```

function [solutions] = MM2(x, n, degree)
    solutions = zeros(degree, 1);
    for k = 1:degree

        for i = 1:n
            z1 = -2*fun(x(k))/(der1_fun(x(k))+sqrt(der1_fun(x(k))^2-
2*fun(x(k))*der2_fun(x(k))));
            z2 = -2*fun(x(k))/(der1_fun(x(k))-sqrt(der1_fun(x(k))^2-
2*fun(x(k))*der2_fun(x(k))));

            if abs(z1) > abs(z2)
                z_min = z2;
            else
                z_min = z1;
            end
            x(k) = x(k)+z_min;
        end
        solutions(k) = x(k)+z_min;
        degree = degree - 1;
    end
end

```

Deflacja czynnikiem liniowym

```

function new_parameters = deflaction(parameters, degree, solution)
    %deflacja liniowa czynnikiem liniowym - prosty schemat Hornera
    new_parameters = zeros(1, degree-1);
    q = 0;
    for (i = 0:degree-1)
        new_parameters(1,degree-1-i) = parameters(1, degree - 1 -i) +
q*solution;
        q = new_parameters(1, degree-1-i);
    end
end

```

Funkcja

```

function y = fun(x)
    y = -1 * x^4 + 1.5 * x^3 + 1.5 * x^2 + 0.5 * x + 1;
end

```

Pierwsza pochodna funkcji

```

function y= der1_fun(x)
    y = -4 * x^3 + 4.5 * x^2 + 3 * x + 0.5;
end

```

Druga pochodna funkcji

```

function y= der2_fun(x)

```

```
y = -12 * x^2 + 9 * x + 3;  
end
```