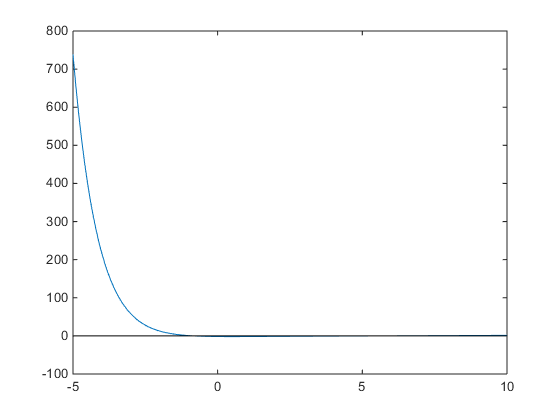
**MNUM – Projekt 3.15**

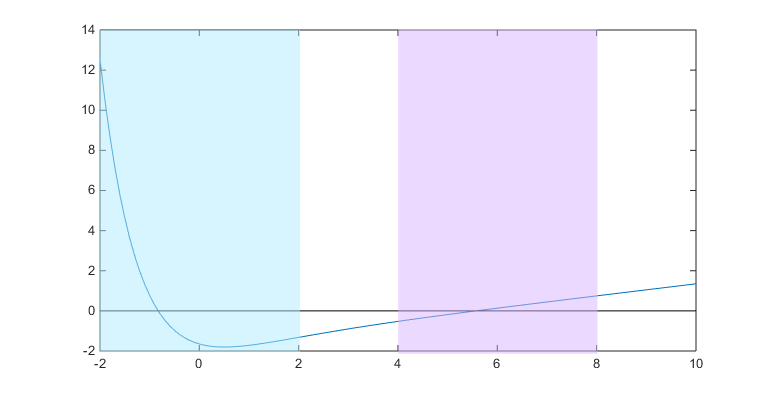
**Zadanie 1**  
Znajdowanie zer funkcji:

W przedziale [-5,10], używając metod:

1. bisekcji
2. siecznych.

Jako że funkcja ta przyjmuje duże wartości dla ujemnych x wykres na całym przedziale nie jest czytelny



Widać, że od pewnego miejsca wartości dla ujemnych x są znacznie większe od zera, więc obszar poszukiwań miejsc zerowych zawęzimy do przedziału [-2, 10].

Stąd przedziały startowe to: [-2,2] oraz [4,8].

1. Metoda bisekcji

Startujemy z początkowego przedziału izolacji pierwiastka. W metodzie bisekcji, w każdej iteracji:

1. Bieżący przedział zawierający zero funkcji, , jest dzielony na dwie połowy, punktem środkowym ,

,

i obliczana jest wartość funkcji w nowym punkcie .

1. Obliczane są iloczyny i , nowy przedział zawierający pierwiastek jest wybierany jako ten z dwóch podprzedziałów, któremu odpowiada iloczyn ujemny. Końce tego przedziału oznaczane są przez , .

Metoda jest powtarzana tak długo, aż np. , gdzie to założona dokładność rozwiązania. Test ten może być nieprecyzyjny dla funkcji, których pochodna jest bardzo mała w otoczeniu zera funkcji, dlatego też sprawdzana jest długość przedziału, , żądając aby była dostatecznie mała.

Dokładność rozwiązania uzyskanego metodą bisekcji zależy jedynie od ilości wykonanych iteracji, a nie zależy od dokładności obliczania wartości funkcji na krańcach kolejnych przedziałów izolowanego pierwiastka. Niech oznacza długość przedziału w   
-tym kroku iteracji,

Wówczas

Stąd metoda bisekcji jest zbieżna liniowo (), z ilorazem zbieżności .

1. Metoda siecznych jest bardzo podobna do metody bisekcji – różnica polega na tym, że aktualny przedział izolacji pierwiastka dzielony jest nie na dwa równe, ale na dwa najczęściej nierówne podprzedziały, prostą (sieczną) łączącą na płaszczyźnie punkty i , przecinającą oś rzędnych w punkcie .   
   W przeciwieństwie jednak do metody bisekcji nie dbamy o zachowanie przedziału izolacji pierwiastka (sieczną prowadzimy zawsze pomiędzy dwoma ostatnio wyznaczonymi punktami i te dwa ostatnie punkty wyznaczają nowy przedział). Z konstrukcji graficznej będziemy mieli:

Skąd

Rząd zbieżności metody siecznych , tak więc metoda ta jest szybsza od metody bisekcji. Jednakże, jest ona zbieżna jedynie lokalnie, stąd w praktyce może być niezbieżna - jeśli początkowy przedział izolacji pierwiastka nie jest dostatecznie mały.

Kod źródłowy:

Metoda bisekcji:  
function [ x, y ] = bisection(a,b,imax)

%sprawdzamy obszar izolacji

if(f(a)\*f(b)>0)

error('Brak obszaru izolacji');

end

for i = 1:imax

%obszar jest dzielony na pół

c = (a+b)/2;

%sprawdzamy warunki iloczynu i wybieramy kolejny przedział

if(f(c) \* f(a)<0)

b = c;

else

a = c;

end

x(i) = c;

y(i) = f(c);

end

end

Metoda siecznych:  
function [ x, y ] = secant( xnp, xn, imax )

%sprawdzamy obszar izolacji

if(f(xnp)\*f(xn)>0)

error('Brak obszaru izolacji');

end

for i = 1:imax+1

%obliczamy wartość kolenego przybliżenie miejsca zerowego

xnn = (xnp\*f(xn) - xn\*f(xnp))/ (f(xn) - f(xnp));

x(i) = xnn;

y(i) = f(xnn);

%Przechodzimy do kolejnego przedziału

xnp = xn;

xn = xnn;

end

end

Wyniki:

Dla pierwszego zaproponowanego przedziału [-2,2] metoda siecznych okazała się być niezbieżna, obszar poszukiwań wychodził poza przedział izolacji i algorytm znajdował zero z drugiego przedziału. W związku z tym obszar izolacji został zawężony do [-1,1].

Przedział [-1,1]:

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
|  | Metoda bisekcji | | Metoda siecznych | |
| Iteracja | x | y | x | y |
| 1 | 0,000000 | -1,650000 | -0,381953 | -1,204969 |
| 2 | -0,500000 | -0,975639 | -3,628549 | 133,905906 |
| 3 | -0,750000 | -0,287250 | -0,410908 | -1,153547 |
| 4 | -0,875000 | 0,186516 | -0,438390 | -1,101921 |
| 5 | -0,812500 | -0,062753 | -1,024977 | 0,899151 |
| 6 | -0,843750 | 0,058653 | -0,761403 | -0,248045 |
| 7 | -0,828125 | -0,002840 | -0,818393 | -0,040342 |
| 8 | -0,835938 | 0,027706 | -0,829462 | 0,002358 |
| 9 | -0,832031 | 0,012383 | -0,828850 | -2,06157E-05 |
| 10 | -0,830078 | 0,004759 | -0,828856 | -1,04023E-08 |

Przedział [4,8]:

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
|  | Metoda bisekcji | | Metoda siecznych | |
| Iteracja | x | y | x | y |
| 1 | 6,000000 | 0,135127 | 5,647320 | 0,024278 |
| 2 | 5,000000 | -0,183690 | 5,568322 | -0,000757 |
| 3 | 5,500000 | -0,022477 | 5,570711 | 1,22E-06 |
| 4 | 5,750000 | 0,056699 | 5,570707 | 6,25E-11 |
| 5 | 5,625000 | 0,017213 | 5,570707 | -2,78E-17 |
| 6 | 5,562500 | -0,002605 | 5,570707 | -2,78E-17 |
| 7 | 5,593750 | 0,007310 |  |  |
| 8 | 5,578125 | 0,002354 |  |  |
| 9 | 5,570313 | -0,000125 |  |  |
| 10 | 5,574219 | 0,001115 |  |  |

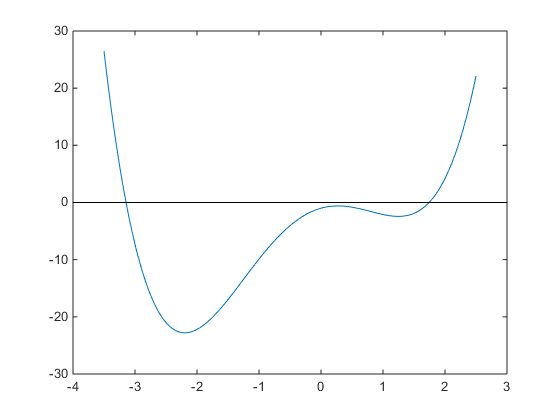
Wnioski:

W przypadku pierwszego inicjalnego przedziału [-2,2] metoda siecznych okazała się niezbieżna, wynika to z tego, że w tym przedziale pochodna funkcji przyjmuje wartości bliskie zeru (funkcja jest „płaska). Przez to sieczna może wyjść poza obszar izolacji, co doprowadzało do znajdowania drugiego zera funkcji. Jednakże po ograniczeniu przedziału do [-1,1], zarówno w nim jak i w drugim przedziale - [4,8], metoda siecznych była znacznie szybciej zbieżna, co potwierdza teoretyczne rozważania. Warto jednak zauważyć, że metoda bisekcji zwracała dobry wynik dla obszaru izolacji o dowolnym rozmiarze.

**Zadanie 2**

Znajdowanie pierwiastków wielomianu

1. Pierwiastków rzeczywistych metodą Newtona
2. Pierwiastków rzeczywistych i zespolonych metodami MM1 i MM2



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

Następny punkt, , wynika z przyrównania do zera sformułowanej lokalnej liniowej aproksymacji funkcji , tzn. z równania

co prowadzi do zależności iteracyjnej

Metoda Newtona jest zbieżna lokalnie - jeśli zaczynamy ją stosować w punkcie zbytnio oddalonym od rozwiązania (poza obszarem atrakcji pierwiastka), to może być ona rozbieżna. Natomiast metoda Newtona jest (lokalnie, asymptotycznie) bardzo szybka, jej zbieżność jest kwadratowa (tzn. z rzędem ).

Metoda Newtona jest szczególnie efektywna w przypadku, gdy krzywa jest bardzo stroma w otoczeniu danego pierwiastka, natomiast nie zaleca się jej stosowania w przypadku, gdy krzywa jest w pobliżu pierwiastka (tzn. punktu przecięcia z osią ) prawie pozioma, tzn. ma bardzo małą wartość.

1. Metoda Müllera

Metoda polega na aproksymacji wielomianu w otoczeniu rozwiązania funkcją kwadratową. Może być traktowana jako uogólnienie metody siecznych - zamiast interpolacji w dwóch punktach funkcją liniową (tzn. sieczną) wykonujemy interpolację w trzech punktach funkcją kwadratową. Istnieje również efektywna 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.

**MM1**. Rozważmy trzy punkty , , wraz z wartościami wielomianu w tych punktach , , . Skonstruujemy funkcję kwadratową przechodzącą przez te punkty, a następnie wyznaczymy pierwiastki tej funkcji i potraktujemy jeden z nich jako kolejne, poprawione przybliżenie rozwiązania (pierwiastka wielomianu).

Przyjmijmy, bez utraty ogólności, że jest aktualną aproksymacją rozwiązania (pierwiastka wielomianu). Wprowadzimy zmienną przyrostową

i wykorzystamy różnice

oznaczając poszukiwaną parabolę przez

Biorąc pod uwagę trzy dane punkty, mamy

Stąd, do wyznaczenia i należy rozwiązać układ równań liniowych

Ponieważ interesuje nas pierwiastek paraboli o najmniejszym module (tzn. położony jak najbliżej ), więc do numerycznego wyznaczenia tego pierwiastka najlepiej wykorzystać wzory:

Do kolejnego przybliżenia rozwiązania bierzemy pierwiastek położony jak najbliżej ,   
tj. o mniejszym module,

Gdzie:

, gdy |

, w przeciwnym przypadku

Przed przejściem do następnej iteracji odrzucamy spośród , , punkt położony najdalej od ostatnio wyznaczonego przybliżenia rozwiązania, tj. punktu .

Algorytm działa prawidłowo również w przypadku, gdy , prowadzi to do wyznaczenia zera zespolonego.

**MM2**. Wersja metody wykorzystująca informację nie o wartości wielomianu w kolejnych trzech punktach, ale o wartości wielomianu i jego pochodnych, pierwszego i drugiego rzędu w aktualnym punkcie (przybliżeniu zera) - wersja nieco efektywniejsza obliczeniowo z tego powodu, że obliczenie wartości wielomianu w punktach jest nieco kosztowniejsze niż obliczenie wartości wielomianu i jego kolejnych pochodnych w jednym punkcie.

Oznaczając szukaną parabolę tak samo jak w MM1:

Dla , w punkcie bezpośrednio wynika:

Co prowadzi do wzoru na pierwiastki:

Do przybliżenia zera α bierzemy pierwiastek paraboli o mniejszym module:

Gdzie jest wybierany spośród w taki sam sposób jak w wersji MM1.

Podobnie jak MM1, MM2 znajduje zespolone pierwiastki wielomianu.

Metoda Müllera jest zbieżna lokalnie, z rzędem zbieżności 1.84. 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).

Kod źródłowy:

Metoda Newtona:   
function [ x, y ] = newton( xn, imax )

for i = 1:imax

%obliczamy kolejne przybliżenie miejsca zerowego

xn = xn - (f2(xn))/(f2p(xn));

x(i) = fn;

y(i) = f2(xn);

end

end

MM1:   
function [ x, y ] = mm1( x0, x1, x2, imax )

for i = 1:imax

%tworzymy zmienne przyrostowe i obliczamy c

z0 = x0 - x2;

z1 = x1 - x2;

c = f2(x2);

%tworzymy układ równań do obliczenia a,b i rozwiązujemy go

A = [z0^2 , z0; z1^2, z1];

B = [f2(x0) - c; f2(x1) - c];

[w] = linsolve(A,B);

a = w(1,1);

b = w(2,1);

%wybieramy zmin jako ten o najmniejszym module

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

%obliczamy kolejne przybliżenie miejsca zerowego

x3 = x2 + zmin;

x(i) = x3;

y(i) = f2(x3);

d0 = abs(x3-x0);

d1 = abs(x3-x1);

d2 = abs(x3-x2);

%Odrzucamy spośród x0,x1,x3 to przybliżenie które jest

% najbardziej oddalone of x3

if(d1 < d0)

u = x1;

x1 = x0;

x0 = u;

end

if(d2<d1)

u = x2;

x2 = x1;

x1 = u;

end

%Przygotowujemy się do kolejnej iteracji

x2 = x3;

end

end

MM2:   
function [ x, y ] = mm2( xn, imax )

for i = 1:imax

pier = sqrt((f2p(xn))^2 - 2\* f2(xn)\*f2b(xn));

%zgodnie z teorią wybiermay pierwiastek o mniejszym module

if(abs(f2p(xn) + pier) > abs(f2p(xn) - pier))

z = (-2\*f2(xn))/ (f2p(xn) + pier);

else

z = (-2\*f2(xn))/ (f2p(xn) - pier);

end

%obliczamy kolejne przybliżenie miejsca zerowego

xn = xn + z;

x(i) = xn;

y(i) = f2(xn);

end

end

Wyniki:

Metoda Newtona

|  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- |
|  | -3 | | -2 | | -1 | | 0 | |
| iteracje | x | y | x | y | x | y | x | y |
| 1 | -3,4475 | 21,7253 | -3,1633 | 1,1132 | 1,8276 | 1,092249 | -0,2774 | -2,3070 |
| 2 | -3,1990 | 3,2637 | -3,1443 | 0,0163 | 1,7523 | 0,104131 | 0,0803 | -0,7974 |
| 3 | -3,1463 | 0,1281 | -3,1440 | 3,68E-06 | 1,7434 | 0,001334 | 0,4680 | -0,7699 |
| 4 | -3,1440 | 0,0002 | -3,1440 | 1,99E-13 | 1,7433 | 2,29E-07 | -0,0088 | -1,0269 |
| 5 | -3,1440 | 7,07E-10 | -3,1440 | -1,42E-14 | 1,7433 | 7,11E-15 | 0,3218 | -0,6152 |
| 6 | -3,1440 | -1,42E-14 | -3,1440 | 1,42E-14 | 1,7433 | -1,78E-15 | -1,0490 | -10,5768 |
| 7 | -3,1440 | 1,42E-14 | -3,1440 | -1,42E-14 | 1,7433 | 3,55E-15 | -0,2903 | -2,3915 |
| 8 | -3,1440 | -1,42E-14 | -3,1440 | 1,42E-14 | 1,7433 | -1,78E-15 | 0,0713 | -0,8162 |
| 9 | -3,1440 | 1,42E-14 | -3,1440 | -1,42E-14 | 1,7433 | 3,55E-15 | 0,4493 | -0,7409 |
| 10 | -3,1440 | -1,42E-14 | -3,1440 | 1,42E-14 | 1,7433 | -1,78E-15 | -0,0501 | -1,1655 |

|  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- |
|  | 1 | | 2 | | 3 | | 4 | |
| iteracje | x | y | x | y | x | y | x | y |
| 1 | 0,3333 | -0,6210 | 0,0870 | -0,7839 | 1,8073 | 0,8063 | 2,4028 | 17,3865 |
| 2 | -0,7919 | -7,1926 | 0,4829 | -0,7948 | 1,7487 | 0,0619 | 2,0186 | 4,6126 |
| 3 | -0,2029 | -1,8614 | 0,0195 | -0,9438 | 1,7434 | 0,0005 | 1,8152 | 0,9149 |
| 4 | 0,1348 | -0,7021 | 0,3605 | -0,6392 | 1,7433 | 2,97E-08 | 1,7500 | 0,0771 |
| 5 | 0,6220 | -1,0890 | -0,4506 | -3,6108 | 1,7433 | -1,78E-15 | 1,7434 | 0,0007 |
| 6 | 0,1787 | -0,6493 | -0,0302 | -1,0960 | 1,7433 | 3,55E-15 | 1,7433 | 7,02E-08 |
| 7 | 0,8520 | -1,7158 | 0,2956 | -0,6066 | 1,7433 | -1,78E-15 | 1,7433 | -3,55E-15 |
| 8 | 0,2370 | -0,6109 | -2,6208 | -19,0983 | 1,7433 | 3,55E-15 | 1,7433 | -1,78E-15 |
| 9 | 1,9316 | 2,8154 | -3,6255 | 39,1427 | 1,7433 | -1,78E-15 | 1,7433 | 3,55E-15 |
| 10 | 1,7812 | 0,4597 | -3,2652 | 7,5710 | 1,7433 | 3,55E-15 | 1,7433 | -1,78E-15 |

|  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- |
|  | -3 | | -2 | | -1 | | 0 | |
| iteracje | x | y | x | y | x | y | x | y |
| 1 | -3.5224 - 0.6721i | 2.1789 +60.5538i | -3.2159 | 4.3236 | -1.5155 | 17.1841 | -0.6737 | -5.8133 |
| 2 | -2.8134 - 0.6029i | -26.5099 +16.3872i | -3.1459 | 0.1086 | -0.9179 | -8.7955 | -0.2143 | -1.9255 |
| 3 | -3.2248 - 0.0203i | 4.8714 + 1.3072i | -3.1440 | 0.0002 | -0.4227 | -3.3765 | 0.3750 + 0.1590i | -0.5462 - 0.1542i |
| 4 | -3.1518 - 0.0028i | 0.4413 + 0.1622i | -3.1440 | -0.0000 | -0.0209 | -1.0652 | 0.2679 + 0.3257i | -0.0791 - 0.0501i |
| 5 | -3.1441 | 0.0049 - 0.0021i | -3.1440 | 0.0000 | 0.2751 + 0.3106i | -0.1316 - 0.0640i | 0.2514 + 0.3475i | 0.0045 - 0.0045i |
| 6 | -3.1440 | -0.0000 | -3.1440 | -0.0000 | 0.2542 + 0.3440i | -0.0092 - 0.0126i | 0.2504 + 0.3461i | -0.0000 |
| 7 | -3.1440 | 0.0000 | -3.1440 | 0.0000 | 0.2504 + 0.3461i | 0.0002 | 0.2504 + 0.3461i | -0.0000 |
| 8 | -3.1440 | -0.0000 |  |  | 0.2504 + 0.3461i | -0.0000 | 0.2504 + 0.3461i | -0.0000 |
| 9 | -3.1440 | 0.0000 |  |  | 0.2504 + 0.3461i | -0.0000 | 0.2504 + 0.3461i | 0.0000 |
| 10 | -3.1440 | -0.0000 |  |  | 0.2504 + 0.3461i | -0.0000 | 0.2504 + 0.3461i | 0.0000 |

|  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- |
|  | 1 | | 2 | | 3 | | 4 | |
| iteracje | x | y | x | y | x | y | x | y |
| 1 | 0.1438 | -0.6896 | 0.3900 + 0.2189i | -0.4705 - 0.2517i | 1.4799 | -1.9873 | 1.8438 | 1.3325 |
| 2 | 0.2550 + 0.3235i | -0.0802 - 0.0063i | 0.2712 + 0.3404i | -0.0318 - 0.0703i | 2.6395 | 30.2054 | 1.7540 | 0.1248 |
| 3 | 0.2519 + 0.3457i | -0.0022 - 0.0054i | 0.2499 + 0.3470i | 0.0034 + 0.0013i | 1.6565 | -0.8745 | 1.7431 | -0.0023 |
| 4 | 0.2504 + 0.3461i | 0.0000 | 0.2504 + 0.3461i | 0.0000 | 1.7553 | 0.1396 | 1.7433 | 0.0000 |
| 5 | 0.2504 + 0.3461i | -0.0000 | 0.2504 + 0.3461i | 0.0000 | 1.7431 | -0.0021 | 1.7433 | 0.0000 |
| 6 | 0.2504 + 0.3461i | -0.0000 | 0.2504 + 0.3461i | -0.0000 | 1.7433 | -0.0000 | 1.7433 | -0.0000 |
| 7 | 0.2504 + 0.3461i | 0.0000 | 0.2504 + 0.3461i | 0.0000 | 1.7433 | -0.0000 | 1.7433 | 0.0000 |
| 8 | 0.2504 + 0.3461i | 0.0000 | 0.2504 + 0.3461i | 0.0000 | 1.7433 | 0.0000 | 1.7433 | -0.0000 |
| 9 |  |  |  |  | 1.7433 | -0.0000 |  |  |
| 10 |  |  |  |  | 1.7433 | -0.0000 |  |  |

|  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- |
|  | -3 | | -2 | | -1 | | 0 | |
| iteracje | x | y | x | y | x | y | x | y |
| 1 | -2.9785 + 0.2922i | -11.5805 -12.2917i | -3.1446 | 0.0340 | -0.8830 | -8.3386 | -0.1273 | -1.4806 |
| 2 | -3.1368 | -0.4112 - 0.0015i | -3.1440 | -0.0000 | -0.0313 | -1.0998 | 0.2380 - 0.3219i | -0.0798 - 0.0513i |
| 3 | -3.1440 | 0.0000 | -3.1440 | -0.0000 | 0.2466 - 0.3221i | -0.0817 - 0.0224i | 0.2504 - 0.3461i | -0.0000 |
| 4 | -3.1440 | -0.0000 | -3.1440 | 0.0000 | 0.2504 - 0.3461i | 0.0000 | 0.2504 - 0.3461i | -0.0000 |
| 5 | -3.1440 | 0.0000 | -3.1440 | -0.0000 | 0.2504 - 0.3461i | -0.0000 | 0.2504 - 0.3461i | 0.0000 |
| 6 | -3.1440 | -0.0000 | -3.1440 | 0.0000 | 0.2504 - 0.3461i | 0.0000 | 0.2504 - 0.3461i | 0.0000 |
| 7 | -3.1440 | 0.0000 | -3.1440 | -0.0000 | 0.2504 - 0.3461i | 0.0000 | 0.2504 - 0.3461i | 0.0000 |
| 8 | -3.1440 | -0.0000 | -3.1440 | 0.0000 | 0.2504 - 0.3461i | 0.0000 | 0.2504 - 0.3461i | 0.0000 |
| 9 | -3.1440 | 0.0000 | -3.1440 | -0.0000 | 0.2504 - 0.3461i | 0.0000 | 0.2504 - 0.3461i | 0.0000 |
| 10 | -3.1440 | -0.0000 | -3.1440 | 0.0000 | 0.2504 - 0.3461i | 0.0000 | 0.2504 - 0.3461i | 0.0000 |

|  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- |
|  | 1 | | 2 | | 3 | | 4 | |
| iteracje | x | y | x | y | x | y | x | y |
| 1 | 0.2500 - 0.3227i | -0.0806 - 0.0108i | 0.4465 | -0.7369 | 1.7278 | -0.1740 | 2.1150 + 0.5232i | -0.1639 +13.0689i |
| 2 | 0.2504 - 0.3461i | 0.0000 | 0.2431 - 0.4042i | 0.2331 + 0.0025i | 1.7433 | 0.0000 | 1.8572 + 0.0301i | 1.5212 + 0.4705i |
| 3 | 0.2504 - 0.3461i | 0.0000 | 0.2503 - 0.3459i | -0.0004 - 0.0002i | 1.7433 | -0.0000 | 1.7425 - 0.0009i | -0.0096 - 0.0099i |
| 4 | 0.2504 - 0.3461i | -0.0000 | 0.2504 - 0.3461i | -0.0000 | 1.7433 | 0.0000 | 1.7433 | -0.0000 |
| 5 | 0.2504 - 0.3461i | 0.0000 | 0.2504 - 0.3461i | -0.0000 | 1.7433 | -0.0000 | 1.7433 | -0.0000 |
| 6 | 0.2504 - 0.3461i | 0.0000 | 0.2504 - 0.3461i | 0.0000 | 1.7433 | 0.0000 | 1.7433 | 0.0000 |
| 7 | 0.2504 - 0.3461i | 0.0000 | 0.2504 - 0.3461i | 0.0000 | 1.7433 | -0.0000 | 1.7433 | -0.0000 |
| 8 | 0.2504 - 0.3461i | 0.0000 | 0.2504 - 0.3461i | 0.0000 | 1.7433 | 0.0000 | 1.7433 | 0.0000 |
| 9 | 0.2504 - 0.3461i | 0.0000 | 0.2504 - 0.3461i | 0.0000 | 1.7433 | -0.0000 | 1.7433 | -0.0000 |
| 10 | 0.2504 - 0.3461i | 0.0000 | 0.2504 - 0.3461i | 0.0000 | 1.7433 | 0.0000 | 1.7433 | 0.0000 |