

Sprawozdanie

OBLICZENIA NAUKOWE, LISTA NR 3

ŁUKASZ KLEKOWSKI 229738

1. Zadania 1-3

1.1. Opis problemu

Polecenia polegały na napisaniu funkcji rozwiązujących równanie $f(x) = 0$ metodą bisekcji, Newtona, oraz siecznych zgodnie ze specyfikacją podaną w zadaniach. Należało także napisać programy testujące.

1.2. Rozwiązanie zadań

Napisałem funkcje opisane poniżej korzystając z pseudokodów podanych na prezentacji z wykładów. Zapakowałem wszystko w moduł ON_L3_Z1-3.jl. Następnie napisałem testy sprawdzające, które testują napisane metody przez obliczenie miejsc zerowych dla funkcji $y = 3x - 3$ oraz $y = \sin(3x - 1)$. Testy załączone są w pliku ON_L3_Z1-3_test.jl

1.2.1. Metoda bisekcji *mbisekcji()*:

I. Opis metody:

Metoda ta służy do rozwiązywania równań nieliniowych, która korzysta z twierdzenia Bolzana – Cauchy’ego:

*„Jeżeli funkcja ciągła f przyjmuje na końcach przedziału domkniętego $< a, b >$ wartości o znakach przeciwnych, tzn. $f(a) * f(b) < 0$, to wewnątrz tego przedziału istnieje co najmniej jeden pierwiastek równania $f(x) = 0$.”*

Aby tą funkcję można było zastosować muszą być spełnione następujące założenia:

1. Funkcja musi być ciągła w przedziale $[a, b]$
2. Funkcja na końców przedziału musi przyjmować różne znaki tzn. $f(a) * f(b) < 0$

II. Przyjmowane argumenty:

Nasz algorytm przyjmuje 5 parametrów:

f – funkcję $f(x)$

a, b – końce przedziału początkowego

$delta, epsilon$ – dokładność wyznaczania pierwiastka oraz dokładność porównania z zerem

III. Zwracane wartości:

Zwraca natomiast czwórkę (r, v, it, err) gdzie:

r – przybliżenie pierwiastka równania $f(x) = 0$

v – wartość $f(r)$

it – liczba wykonanych iteracji

err – sygnalizacja błędu

0 – brak błędu

1 – funkcja nie zmienia znaku w przedziale $[a, b]$

IV. Przebieg algorytmu:

Na początku nasz algorytm oblicza wartości funkcji na końcach przedziału: $f(a)$ oraz $f(b)$. Jeśli otrzymane wartości mają równe znaki oznacza to że nie ma w tym przedziale pierwiastka i zwracany jest błąd $err = 1$. W przeciwnym razie przechodzimy do głównej pętli.

Jest ona wykonywana *maxit* razy. Na początku obliczamy środek przedziału i obliczamy wartość funkcji w tym punkcie. Później sprawdzamy czy te rozwiązanie jest satysfakcjonujące (czy mieścimy się w naszych przybliżeniach). Jeśli tak zwracamy naszą czwórkę z błędem $err = 0$. W przeciwnym wypadku „wchodzimy” do tej połowy przedziału gdzie są różne znaki na końcach przedziału i powtarzamy poprzednie obliczenia.

Jeśli po przejściu pętli nie nadal nie otrzymaliśmy rozwiązania funkcja zwraca czwórkę z ostatnio obliczonymi rozwiązaniami. W specyfikacji nie ma wyróżnionego błędu który by to określał więc musimy patrzeć na ilość iteracji.

1.2.2. Metoda stycznych (Netwona) *mstycznych()*:

I. Opis metody:

Jest to algorytm stosowany do obliczenia przybliżonej wartości miejsca zerowego funkcji. Polega on na tym że w pierwszym kroku po wybraniu punktu startowego x_0 , prowadzona jest styczna w $f(x_0)$. Miejsce przecięcia się stycznej z osią OX jest pierwszym przybliżeniem rozwiązania. Jeśli to rozwiązanie nie jest wystarczające to te kroki są powtarzane dla nowego punktu x_1 . Algorytm jest powtarzany do momentu kiedy dostaniemy satysfakcjonujące rozwiązanie lub gdy skończą się iteracje.

Ogólniej mówiąc: startując z pewnego przybliżenia początkowego x_0 , w kolejnych krokach tej metody k -te przybliżenie x_k jest punktem przecięcia się stycznej do wykresu $f(x)$ w punkcie x_{k-1} . Ponieważ równanie stycznej to:

$$y(x) = f(x_{k-1}) + f'(x_{k-1})(x - x_{k-1}),$$

i otrzymujemy następujący wzór dzięki któremu obliczamy kolejne punkty styczności prostej z funkcją:

$$x_k = x_{k-1} - \frac{f(x_{k-1})}{f'(x_{k-1})} \quad \text{dla } k = 1, 2, 3, 4, \dots \quad (*)$$

II. Przyjmowane argumenty:

Nasz algorytm przyjmuje 6 parametrów:

f, pf – funkcję $f(x)$ oraz jej pochodną $f'(x)$

$x0$ – przybliżenie początkowe

δ, ϵ – dokładność wyznaczania pierwiastka oraz dokładność porównania z zerem

$maxit$ – maksymalna ilość iteracji

III. Zwracane wartości:

Zwraca natomiast czwórkę (r, v, it, err) gdzie:

r – przybliżenie pierwiastka równania $f(x) = 0$

v – wartość $f(r)$

it – liczba wykonanych iteracji

err – sygnalizacja błędu

0 – metoda zbieżna

1 – nie osiągnięto wymaganej dokładności w $maxit$ iteracji

2 – pochodna bliska zeru

IV. Przebieg algorytmu:

Na początku obliczamy wartość funkcji w punkcie startowym i jeśli jej wartość bezwzględna jest mniejsza od ϵ to zwracamy czwórkę z aktualnymi wartościami oraz z $err = 0$.

Następnie wchodzimy do głównej pętli gdzie obliczamy wartość pochodnej funkcji w aktualnym punkcie. Jeśli jej wartość bezwzględna jest mniejsza od dokładności podanej w funkcji to zwracamy aktualną czwórkę z błędem 2.

Później obliczamy następny argument za pomocą wzoru opisanego wyżej (*), oraz liczymy wartość funkcji w tym punkcie.

Na końcu pętli sprawdzamy czy nasze rozwiązania zgadzają się z dokładnościami podanymi w parametrach funkcji. Jeśli tak to zwracamy czwórkę z $err = 0$, jeśli nie iterujemy dalej.

Jeśli iteracje się skończą funkcja zwróci aktualnie obliczony pierwiastek, jego wartość w funkcji, $maxit$ oraz $err = 1$, ponieważ nie osiągnięto odpowiedniej dokładności.

1.2.3. Metoda siecznych $msiecznych()$:

I. Opis metody:

Jest to metoda numeryczna, która służy do obliczania równań z jedną niewiadomą. Polega ona na przyjęciu że na dostatecznie małym odcinku $< x_0, x_1 >$ funkcja zmienia się w sposób liniowy. Wtedy na odcinku $< x_0, x_1 >$ zastępujemy sieczną krzywą $y = f(x)$. Za przybliżoną wartość pierwiastka funkcji przyjmujemy punkt przecięcia się siecznej w oś OX .

Ogólniej mówiąc: ta metoda wykorzystuje do konstrukcji x_k przybliżenia x_{k-1} oraz x_{k-2} . Wybieramy dwa punkty startowe x_0 oraz x_1 . Ponieważ równanie siecznej funkcji f dla punktów x_{k-1} oraz x_{k-2} ma postać:

$$y(x) = \frac{x - x_{k-2}}{x_{k-1} - x_{k-2}} f(x_{k-1}) + \frac{x - x_{k-1}}{x_{k-2} - x_{k-1}} f(x_{k-2})$$

możemy otrzymać możemy otrzymać to

$$x_k = x_{k-1} - f(x_{k-1}) \frac{x_{k-1} - x_{k-2}}{f(x_{k-1}) - f(x_{k-2})} \quad \text{dla } k = 1, 2, 3, 4, \dots \quad (**)$$

II. Przyjmowane argumenty:

Nasz algorytm przyjmuje 6 parametrów:

F – funkcję $f(x)$

x_0, x_1 – przybliżenia początkowe

$delta, epsilon$ – dokładność wyznaczania pierwiastka oraz dokładność porównania z zerem

$maxit$ – maksymalna ilość iteracji

III. Zwracane wartości:

Zwraca natomiast czwórkę (r, v, it, err) gdzie:

r – przybliżenie pierwiastka równania $f(x) = 0$

v – wartość $f(r)$

it – liczba wykonanych iteracji

err – sygnalizacja błędu

0 – metoda zbieżna

1 – nie osiągnięto wymaganej dokładności w $maxit$ iteracji

Przebieg algorytmu:

Na samym początku liczymy wartość funkcji dla argumentów x_0 i x_1 . Następnie wchodzimy do głównej pętli.

Jeśli wartość bezwzględna dla $f(x_0)$ jest większa od $f(x_1)$ to zamieniamy te wartości ze sobą oraz zamieniamy argumenty miejscami.

Następnie do $f(x_1)$ oraz do x_1 przypisujemy odpowiednio $f(x_0)$ oraz x_0 a kolejne x_0 liczymy za pomocą wzoru podanego wyżej (**). Dla nowo policzonego x_0 liczymy wartość funkcji w tym punkcie $f(x_0)$

Na końcu pętli sprawdzamy czy nasze rozwiązania zgadzają się z warunkami podanymi w parametrach funkcji, czyli czy osiągnęliśmy odpowiednią dokładność wyznaczania pierwiastka oraz odpowiednie przybliżenie do zera

Jeśli warunki są spełnione metoda zwraca pierwiastek, wartość funkcji dla pierwiastka, ilość iteracji oraz $err = 0$ czyli znak że wszystko przebiegło po naszej myśli.

Jeśli iteracje się skończą funkcja zwróci aktualnie obliczony pierwiastek, jego wartość w funkcji $maxit$ oraz $err = 1$, ponieważ nie osiągnięto odpowiedniej dokładności w przeprowadzonych iteracjach.

2. Zadanie czwarte

2.1. Opis zadania

Polecenie polegało na obliczeniu pierwiastka równania $\sin(x) - \left(\frac{1}{2}x\right)^2 = 0$ za pomocą wcześniej napisanych metod.

2.2. Wyniki

2.2.1. Metoda bisekcji z przedziałem początkowym $[1.5, 2]$, $delta, epsilon = \frac{1}{2} * 10^{-5}$

Funkcja zwróciła następującą czwórkę:

$$(1.9337539672851562, -2.7027680138402843e-7, 16, 0)$$

2.2.2. Metoda Newtona z przybliżeniem początkowym $x_0 = 1.5$ i $delta, epsilon = \frac{1}{2} * 10^{-5}$

Funkcja zwróciła następującą czwórkę:

$$(1.933753779789742, -2.2423316314856834e-8, 4, 0)$$

2.2.3. Metoda siecznych z przybliżeniami początkowymi $x_0 = 1$, $x_1 = 2$ i $delta, epsilon = \frac{1}{2} * 10^{-5}$

Funkcja zwróciła następującą czwórkę:

$$(1.933753644474301, 1.564525129449379e-7, 4, 0)$$

2.3. Wnioski

Nazwa metody	r	v	Liczba iteracji
Metoda bisekcji	1.9337539672851562	-2.7027680138402843e-7	16
Metoda Newtona	1.933753779789742	-2.2423316314856834e-8	4
Metoda Siecznych	1.933753644474301	1.564525129449379e-7	4

Z powyższego porównania widać, że metody zwróciły podobne wyniki (zgadzają się do 6. miejsca po przecinku), dla których funkcja przyjmuje wartości bardzo bliskie zeru lecz nie są to faktycznie zera. Gdybyśmy podali funkcji lepsze przybliżenia zapewne otrzymalibyśmy wyniki bliższe prawdzie. Można zauważyć że funkcja bisekcji potrzebuje więcej iteracji, ponieważ nie posiada ona metod linearyzacji, a jedynie dzieli na połowy przeszukiwany przedział.

3. Zadanie trzecie

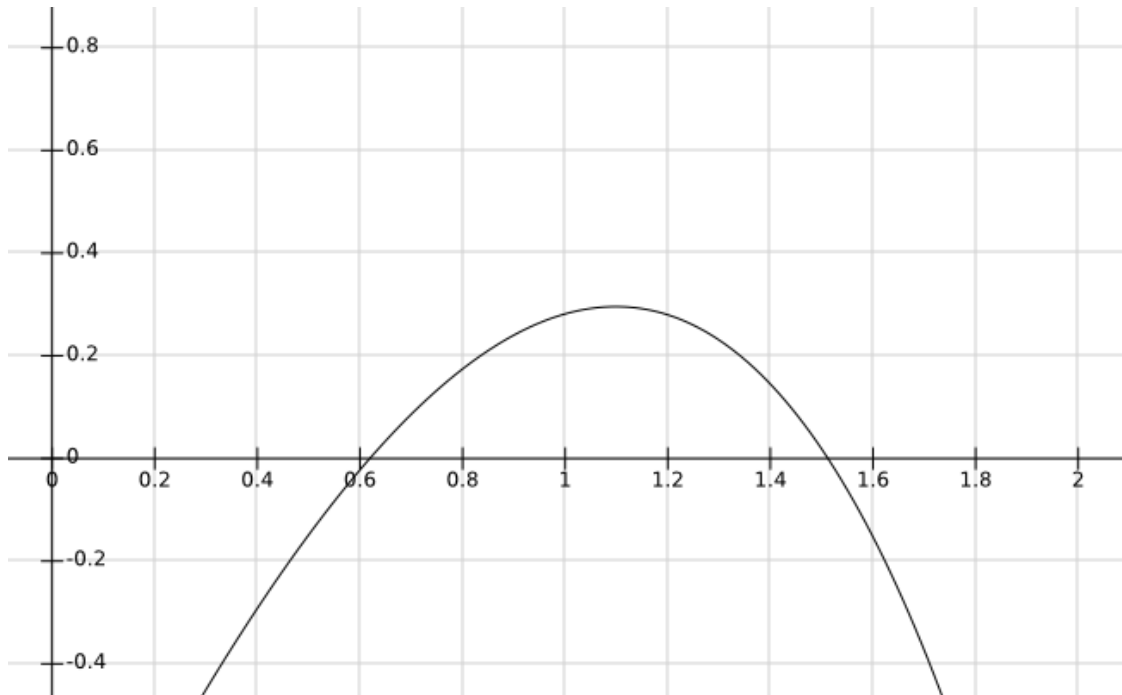
3.1.Opis zadania

Polecenie polegało na obliczeniu metodą bisekcji znaleźć taką wartość zmiennej x dla której przecinają się wykresy funkcji $y = 3x$ oraz $y = e^x$.

3.2.Opis rozwiązania

Aby obliczyć miejsca przecięcia się tych funkcji należy obliczyć miejsca zerowe dla funkcji $y = 3x - e^x$.

Narysujmy wykres tej funkcji:



Widzimy że te dwie funkcje przecinają się w 2 miejscach: gdzieś w przedziale $[0,1]$ oraz $[1,2]$.

Uruchomimy zatem naszą funkcję dwukrotnie: raz dla przedziału $[0,1]$, a drugi dla przedziału $[1,2]$.

3.3.Wyniki

Przedział	r	v	Liczba iteracji	err
$[0,1]$	0.619140625	9.066320343276146e-5	9	0
$[1,2]$	1.5120849609375	7.618578602741621e-5	13	0

3.4.Wnioski

Przyjmując znalezione wcześniej przedziały obliczyliśmy przybliżone miejsca zerowe. Jak wiadomo nie są one jednak rzeczywistymi rozwiązaniami tego równania. W tym zadaniu bardzo

ważne jest odpowiednie przeanalizowanie obliczanych funkcji, aby wskazać odpowiednie przedziały. W tym przypadku jeśli podalibyśmy przedział $[0,2]$ funkcja zwróciłaby błąd 1, czyli te same znaki funkcji na końcu przedziałów. Oznacza to że w danym przedziale musi być tylko i wyłącznie 1 pierwiastek.

4. Zadanie szóste

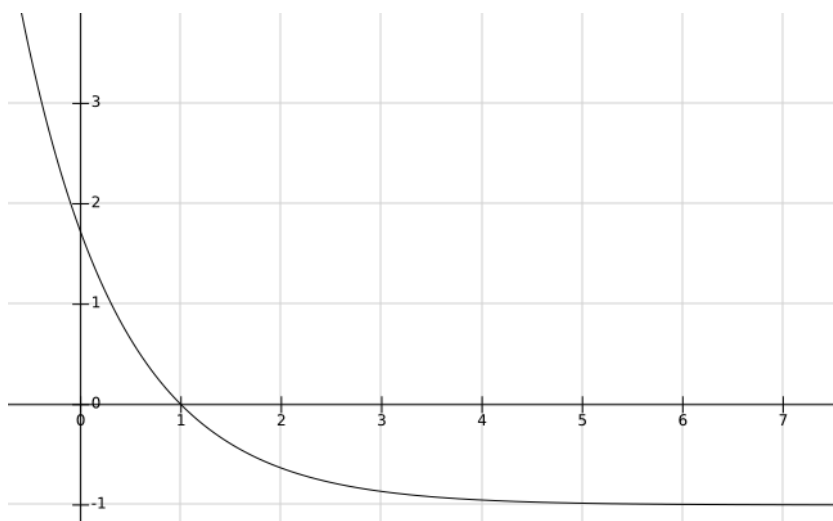
4.1. Opis zadania

Polecenie polegało na znalezieniu miejsc zerowych funkcji $f_1(x) = e^{1-x} - 1$ oraz $f_2(x) = xe^{-x}$ za pomocą wszystkich metod z dokładnością $\delta, \epsilon = 10^{-5}$. Następnie należało sprawdzić co się stanie jeśli w metodzie Newtona dla f_1 oraz f_2 wybierzemy $x \in (1, \infty)$, oraz czy możemy dla f_2 przyjąć $x_0 = 1$.

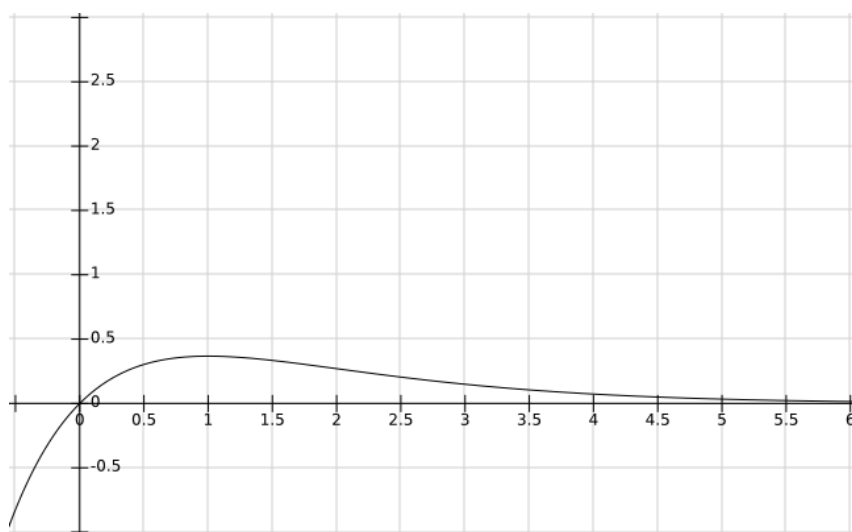
4.2. Interpretacja wyników

Narysujmy wykresy, które pomogą wybrać nam odpowiednie przedziały:

Funkcja pierwsza:



Funkcja druga:



4.2.1. Funkcja f_1

Łatwo policzyć to, że miejscem zerowym dla funkcji f_1 jest 1 (widzimy to również na wykresie). Skonfrontujmy to teraz z wynikami danych metod:

I. Metoda bisekcji:

Przedział	r	v	Liczba iteracji	err
[-1,5]	1.0	0	1	0
[1.41,5.31]	1.000001220703125	-1.2207023799559735e-6	18	0

Jak widać gdy daliśmy przedział który w połowie ma miejsce zerowe funkcja zwróciła dokładny wynik, a gdy ten przedział był już mniej przyjemny dla algorytmu to wyniki te nie są już dokładne oraz potrzebowaliśmy więcej iteracji, lecz wyniki są bliskie prawdy.

II. Metoda newtona:

Posiadając wyniki metody bisekcji jako punkt startowy wybrałem 0.7. Jak widać poniżej do otrzymania wyniku potrzebowiałem tylko 3 iteracji.

x_0	r	v	Liczba iteracji	err
0.7	0.9999996623790672	3.376209898320326e-7	3	0

Następnie mieliśmy sprawdzić co się stanie gdy jako punkt startowy wybierzemy $x_0 \in (1, \infty)$.

x_0	r	v	Liczba iteracji	err
1	1.0	0.0	1	0
2	0.9999999810061002	1.8993900008368314e-8	5	0
3	0.9999999710783241	2.892167638712806e-8	9	0
4	0.999999995278234	4.721765201054495e-10	21	0
5	0.9999996427095682	3.572904956339329e-7	54	0
6	0.999999573590406	4.264096031825204e-8	147	0
7	0.999999484165362	5.15834650549607e-8	401	0
8	NaN	NaN	1000	1
9	NaN	NaN	1000	1
10	NaN	NaN	1000	1
11	NaN	NaN	1000	1
12	NaN	NaN	1000	1
13	13	-0.9999938557876467	0	2
14	14	-0.999997739670593	0	2
15	15	-0.9999991684712809	0	2
19	19	-0.9999999847700203	0	2
20	20	-0.9999999943972036	0	2
21	21	-0.9999999979388464	0	2
29	29	-0.9999999999993086	0	2
30	30	-0.9999999999997456	0	2
31	31	-0.9999999999999064	0	2
39	39	-1	0	2
40	40	-1	0	2

41	41	-1	0	2
48	48	-1	0	2
49	49	-1	0	2
50	50	-1	0	2

Jak widać wyżej dla punktów startowych od 1 do 7 funkcja znajdowała miejsca zerowe choć potrzebowała bardzo dużej ilości iteracji. Problem pojawił się gdy punktami startowymi stały się liczby od 8 do 12 otrzymywaliśmy wtedy błąd 1, czyli niewystarczającą ilość iteracji.

O dziwo jako wyniki otrzymywaliśmy *NaN* (*Not a Number*). Dzieje się tak ponieważ pochodna z tej funkcji jest bliska zeru, lecz niewystarczająco blisko, aby funkcja złapała błąd 2 (pochodna bliska zeru). Algorytm wchodzi następnie do pętli i dochodzimy do momentu gdzie wartość funkcji oraz jej pochodnej to nieskończoność, a dzielenie nieskończoności przez nią samą daje *NaN*. Aby ten algorytm był kompletny należałoby jeszcze sprawdzać czy pochodna funkcji znajduje się w nieskończoności.

Dla argumentów powyżej 12 było już wszystko w porządku. Algorytm obliczał pochodną gdzie była ona bliska zeru i zwracał $err = 2$.

III. Metoda siecznych:

Dla kolejnej metody wybrałem punkty startowe będące blisko oraz daleko miejsca zerowego.

x_0, x_1	r	v	Liczba iteracji	err
-1.1; 5.0	1.0000025776702943	-2.577666972092274e-6	4	0
-98.0; 98.0	-98.0	-1.0	1	0

Jak widać dla punktów startowych będących blisko miejsca zerowego wyniki są poprawne. Problem pojawił się dla drugiego zestawu danych. Na pierwszy rzut oka można stwierdzić, że wszystko jest w porządku ponieważ funkcja nie zwróciła błędów lecz jest to złudne, ponieważ -98 nie jest miejscem zerowym dla tej funkcji.

4.2.2. Funkcja f_2

Łatwo policzyć to, że miejscem zerowym dla funkcji f_2 jest 0 (widzimy to również na wykresie). Skonfrontujmy to teraz z wynikami danych metod:

I. Metoda bisekcji:

Przedział	r	v	Liczba iteracji	err
[-1,1]	0.0	0.0	1	0
[-7.71,6.32]	-2.861022883190946e-8	-2.861022965045467e-8	20	0

Tak jak w przypadku funkcji 1, gdy miejsce zerowe znajdowało się na środku wybranego przedziału to metoda zwróciła poprawny wynik w 1. iteracji. Dla bardziej skomplikowanych danych wejściowych funkcja potrzebowała więcej iteracji, lecz też sobie poradziła.

II. Metoda Newtona:

Sugerując się wynikami metody bisekcji jako punkt startowy wybrałem $x_0 = 0.2$

x_0	r	v	Liczba iteracji	err
0.7	-5.65546883836681e-6	-5.655500822785036e-6	3	0

Miejsce zerowe zostało odnalezione w tylko 3 iteracjach.

Kolejną częścią zadania było to, aby sprawdzić co się stanie gdy punktami startowymi będą argumenty większe niż 1 oraz co się stanie gdy za punkt startowy przyjmiemy 1.

x_0	r	v	Liczba iteracji	err
1	0.0	0.36787944117144233	1	2
2	14.398662765680003	8.036415344217211e-6	10	0
3	14.787436802837927	5.594878975694858e-6	10	0
4	14.398662765680003	8.036415344217211e-6	9	0
5	14.118053159563352	3.572904956339329e-7	9	2
6	14.97432014974184	4.699833827208111e-6	8	0
13	15.159766454352443	3.95266121872815e-6	2	0
14	15.076923076923077	4.270593381508261e-6	1	0
15	15.0	4.588534807527386e-6	0	0
16	16.0	1.8005627955081459e-6	0	0
19	19.0	1.0645313231320808e-7	0	0
20	20.0	4.122307244877116e-8	0	0
21	21.0	1.5923376898615004e-8	0	0
29	29.0	7.376630377393076e-12	0	0
30	30.0	2.8072868906520526e-12	0	0
31	31.0	1.0671679036256928e-12	0	0
39	39.0	4.503807427476156e-16	0	0
40	40.0	1.6993417021166355e-16	0	0
41	41.0	6.407816976273453e-17	0	0
48	48.0	6.840787597156489e-20	0	0
49	49.0	2.5690139750480974e-20	0	0
50	50.0	9.643749239819589e-21	0	0

Zacznijmy od argumentu 1. Nie można go dać jako punkt startowy ponieważ pochodna funkcji w tym punkcie wynosi 0.

Dla punktów startowych większych od 1 metoda zwracała już błędne wyniki choć błąd wynosił 0, czyli wszystko powinno być w porządku.

Dla argumentów od 2 do 14 funkcja zwracała błędne wyniki, ponieważ funkcja ta jest zbieżna lokalnie i jeśli nie zadamy o to aby punkt startowy był wystarczająco blisko miejsca zerowego to funkcja zwróci nam złe wyniki.

Dla argumentów większych niż 14 funkcja zwracała miejsca zerowe takie same jak punkt startowy. Jest to spowodowane tym, że omawiana funkcja jest zbieżna do zera i od pewnego momentu wartość funkcji jest mniejsza od *epsilona* podawanego w funkcji i algorytm przyjmuje podawany punkt startowy jako miejsce zerowe.

I. Metoda siecznych:

x_0, x_1	r	v	Liczba iteracji	err
-1.1; 1.6	14.53024892277346	-2.577666972092274e-6	15	0
-0.3; 0.3	-2.6954217796065577e-7	-2.695421053076994e-7	5	0

Tak jak w funkcji 1. Dla zestawu danych które są blisko miejsca zerowego funkcja zwróciła poprawny wynik. Gdy oddaliliśmy miejsca startowe funkcja zwróciła zły wynik z błędem 0. Ten eksperyment pokazuje zbieżność lokalną tej metody.

4.3. Wnioski

Eksperymenty te pokazują plusy oraz minusy używanych metod.

Metoda bisekcji jest zbieżna globalnie. Oznacza to że jest ona stabilniejsza przez to że nie musimy dbać o przedział początkowy. Ważne jest jedynie to abyśmy mieli pewność, że jest tam pierwiastek funkcji. Minusem tej metody jest to, że nie posiada ona metod linearyzacji przez to potrzebujemy więcej iteracji, aby dotrzeć do rozwiązania.

Metody Newtona oraz siecznych są zbieżne lokalnie, czyli są wrażliwe na dane początkowe. Jeśli punkty startowe będą za daleko od miejsca zerowego to funkcja nie zwróci nic lub poda błędne wyniki. Plusem natomiast jest ich szybkość. Przez to że korzystają z linearyzacji są w stanie obliczyć pierwiastek o wiele szybciej.

Przy obliczaniu pierwiastków często najpierw wykorzystywana jest metoda bisekcji ze względu na zbieżność globalną, a gdy dojdziemy do pewnego przybliżenia to obliczenia są przełączane na pozostałe metody ze względu na ich szybkość.