

Obliczenia naukowe

Laboratorium

Lista 3

Kinga Majcher
272354

Listopad 2024

1 Zadanie 1

1.1 Opis problemu

Napisać funkcję, która rozwiązuje równanie $f(x) = 0$ metodą bisekcji.
Funkcja przyjmuje parametry:

f - funkcja $f(x)$ zadana jako anonimowa funkcja

a, **b** - końce przedziału początkowego

delta, **epsilon** - dokładności obliczeń

Funkcja zwraca:

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 $[a, b]$

1.2 Opis metody

Idea metody bisekcji wywodzi się z twierdzenia Darboux. Twierdzenie to mówi o tym, że jeśli funkcja $f(x)$ jest ciągła na przedziale $[a, b]$ i na jego końcach przyjmuje różne wartości to przyjmuje również wszystkie wartości pośrednie znajdujące się między $f(a)$, a $f(b)$. W szczególności oznacza to, że dla przedziału $[a, b]$, takiego, że $f(a) \cdot f(b) < 0$, funkcja $f(x)$ posiada co najmniej jedno miejsce zerowe w tym przedziale.

Aby precyzyjnie wyznaczyć miejsce zerowe funkcji, konieczne jest stopniowe zawężanie badanego przedziału. W metodzie bisekcji przedział ten dzieli się na pół, a następnie sprawdza, na której połowie nadal zachodzi zmiana znaku funkcji. Tę połowę wybiera się jako nowy przedział. Proces ten powtarza się aż do momentu, w którym wartość funkcji w środku przedziału będzie dostatecznie bliska zera.

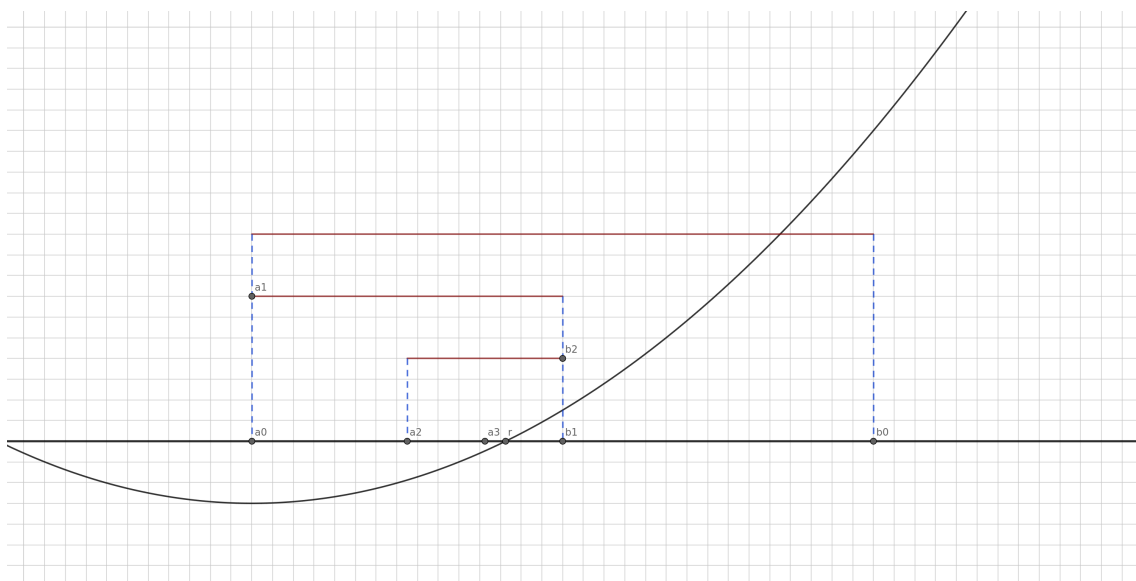
Usprawnienia metody

1. Sprawdzenie zmiany znaku: Zamiast obliczać iloczyn $f(a) \cdot f(b)$, co jest obciążone dużym błędem numerycznym wynikającym z ograniczonej precyzji arytmetyki, wykorzystuje się fakt, że w standardzie IEEE 754 jeden bit jest przeznaczony na reprezentację znaku liczby. W związku z tym korzystamy z funkcji $\text{sign}(x)$, która zwraca znak wartości x , umożliwiając dokładne sprawdzenie, czy na końcach przedziału występuje zmiana znaku.

2. Obliczanie środka przedziału: Klasyczne obliczanie środka przedziału za pomocą wzoru $c = \frac{a+b}{2}$ może prowadzić do błędów numerycznych, szczególnie gdy wartości a i b są bardzo bliskie sobie. W takich przypadkach, z powodu ograniczonej precyzji arytmetyki zmiennoprzecinkowej, wynik obliczeń może znajdować się poza zakresem przedziału $[a, b]$.

Bardziej niezawodnym podejściem jest obliczenie długości przedziału jako $e = b - a$, a następnie wyznaczenie środka za pomocą wzoru $c = a + \frac{e}{2}$. Po każdej iteracji wartość e jest dzielona przez 2, co pozwala precyzyjnie zawęzić przedział. Operacja dzielenia przez 2 w standardzie IEEE 754 modyfikuje jedynie wykładnik liczby zmiennoprzecinkowej, dzięki czemu nie jest obciążona błędami arytmetyki. Dodatkowo, dodawanie do a połowy długości przedziału jest bardziej stabilne numerycznie niż odejmowanie małych liczb, co istotnie redukuje potencjalne błędy związane z reprezentacją granic przedziału.

1.3 Interpretacja graficzna metody



Rysunek 1: Przedstawienie graficzne iteracyjnego wyznaczania $f(x) = 0$ metodą bisekcji

1.4 Pseudokod algorytmu

Implementacja metody bisekcji została napisana na podstawie poniższego pseudokodu:

Algorithm 1: Algorytm metody bisekcji

```
Input :  $f, a, b, \delta, \epsilon$   
Output:  $r, v, it, err$   
1  $fa \leftarrow f(a);$   
2  $fb \leftarrow f(b);$   
3  $e \leftarrow b - a;$   
4 if  $sgn(fa) = sgn(fb)$  then  
5   return Nothing, Nothing, Nothing, 1;  
6  $k \leftarrow 1;$   
7 while true do  
8    $e \leftarrow e/2;$   
9    $m \leftarrow a + e;$   
10   $fm \leftarrow f(m);$   
11  if  $|e| < \delta$  or  $|fm| < \epsilon$  then  
12    return  $m, fm, k, 0;$   
13  if  $sgn(fm) \neq sgn(fa)$  then  
14     $b \leftarrow m;$   
15     $fb \leftarrow fm;$   
16  else  
17     $a \leftarrow m;$   
18     $fa \leftarrow fm;$   
19   $k \leftarrow k + 1;$ 
```

2 Zadanie 2

2.1 Opis problemu

Napisać funkcję, która rozwiązuje równanie $f(x) = 0$ metodą Newtona.

Funkcja przyjmuje parametry:

f, **pf** - funkcja $f(x)$ oraz jej pochodna $f'(x)$ zadane jako anonimowe funkcje

x0 - przybliżenie początkowe

delta, **epsilon** - dokładności obliczeń

maxit - maksymalna dopuszczalna liczba iteracji

Funkcja zwraca:

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

2.2 Opis metody

Metoda Newtona opiera się na wykorzystaniu rozwinięcia funkcji w szereg Taylora w punkcie x_n . Dla funkcji dwukrotnie różniczkowalnej oraz x_n , będącego obecnym przybliżeniem miejsca zerowego, mamy:

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

Celem jest znalezienie takiego \tilde{r} , że $f(\tilde{r}) = 0$. Podstawiając do wzoru stycznej:

$$f(\tilde{r}) = 0 \approx f(x_n) + (\tilde{r} - x_n) \cdot f'(x_n)$$

otrzymujemy:

$$f'(x_n) \cdot \tilde{r} = f'(x_n) \cdot x_n - f(x_n)$$

a stąd:

$$\tilde{r} = x_n - \frac{f(x_n)}{f'(x_n)}$$

Operację tę można powtórzyć kilkakrotnie dla większej dokładności uzyskanego przybliżenia. Równanie rekurencyjne opisujące metodę Newtona ma postać:

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

Problemy metody

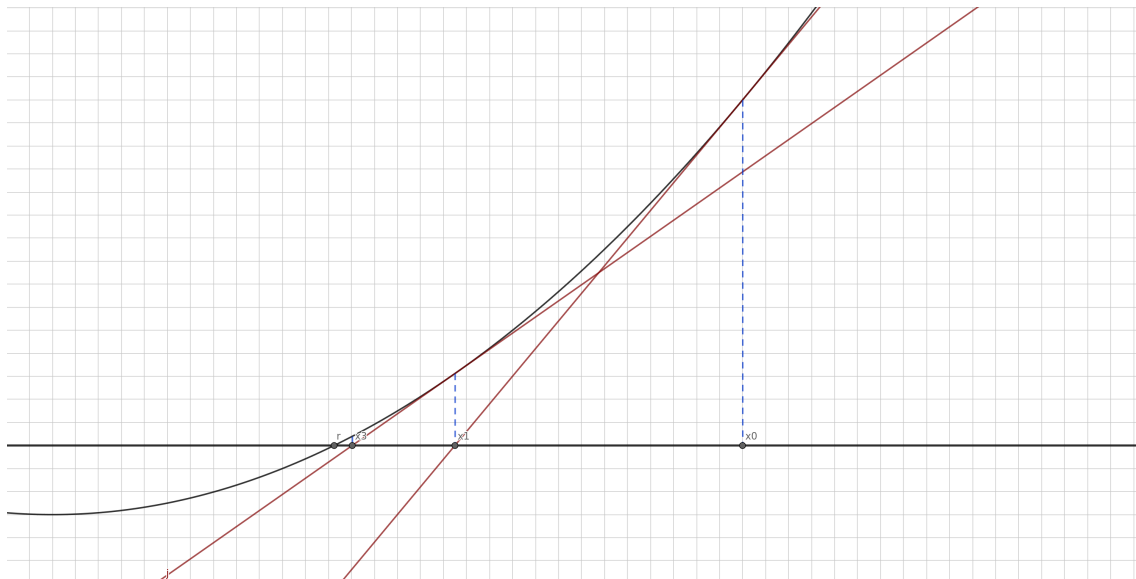
1. Fakt znajomości pochodnej: Korzystanie z metody Newtona wymaga znajomości pochodnej funkcji, co wiąże się z dodatkowymi obliczeniami. Dla złożonych funkcji obliczenie pochodnej może być trudne i czasochłonne.

2. Zbieżność: Metoda Newtona jest zbieżna lokalnie, co oznacza, że początkowe przybliżenie x_0 musi być wybrane z dostatecznie małego otoczenia pierwiastka. W przeciwnym wypadku kolejne iteracje mogą prowadzić do coraz większego oddalania się od rzeczywistego rozwiązania. Aby tego uniknąć, w implementacji metody Newtona dodana jest maksymalna liczba iteracji, którą chcemy wykonać. Jeżeli po ich wykonaniu rozwiązanie nie spełnia wymagań dokładności, algorytm zwraca uzyskaną wartość wraz z informacją o niespełnieniu warunków zbieżności.

3. Pochodna zbyt bliska zera: W algorytmie konieczne jest obliczenie ilorazu $\frac{f(x_n)}{f'(x_n)}$. Dla wartości $f'(x_n)$ bliskich zeru wynik dzielenia może stać się bardzo duży, co prowadzi do generowania kolejnych przybliżeń znacznie oddalonych od poprzednich. Jako że metoda Newtona jest zbieżna jedynie lokalnie, takie przybliżenia mogą skutkować rozbiegnięciem się algorytmu. Dodatkowo, gdy $f'(x_n) = 0$, będzie występować dzielenie przez 0, co jest błędem, którego chcemy uniknąć.

2.3 Interpretacja graficzna metody

Wzór $f(x) \approx f(x_n) + (x - x_n) \cdot f'(x_n)$ stanowi również równanie stycznej do wykresu funkcji $f(x)$ w punkcie x_n , czemu metoda ta zawdzięcza swoją drugą nazwę - metoda stycznych. Działanie metody Newtona można łatwo zrozumieć, odwołując się do jej interpretacji graficznej, zaprezentowanej poniżej. Tworzymy styczną do wykresu funkcji $f(x)$ w punkcie x_0 , a miejsce jej przecięcia z osią OX wyznacza wartość kolejnego przybliżenia x .



Rysunek 2: Przedstawienie graficzne iteracyjnego wyznaczania $f(x) = 0$ metodą Newtona

2.4 Pseudokod algorytmu

Implementacja metody Newtona została napisana na podstawie poniższego pseudokodu:

Algorithm 2: Algorytm metody Newtona

Dane : $f, pf, x_0, \delta, \epsilon, maxit$

Wyniki: r, v, it, err

```

1  $fx \leftarrow f(x_0)$ ;
2 if  $|fx| < \epsilon$  then
3   return  $x_0, fx, 0, 0$ ;
4 for  $k \leftarrow 1$  to  $maxit$  do
5    $pdfx \leftarrow f'(x_0)$ ;
6   if  $|pdfx| < eps(Float64)$  then
7     return  $x_0, fx, k, 2$ ;
8    $x_1 \leftarrow x_0 - fx/pdfx$ ;
9    $fx \leftarrow f(x_1)$ ;
10  if  $|x_1 - x_0| < \delta$  or  $|fx| < \epsilon$  then
11    return  $x_1, fx, k, 0$ ;
12   $x_0 \leftarrow x_1$ ;
13 return  $x_0, fx, maxit, 1$ ;
```

3 Zadanie 3

3.1 Opis problemu

Napisać funkcję, która rozwiązuje równanie $f(x) = 0$ metodą siecznych. Funkcja przyjmuje parametry:

f - funkcja $f(x)$ zadana jako anonimowa funkcja

x_0, x_1 - przybliżenia początkowe

δ, ϵ - dokładności obliczeń

$maxit$ - maksymalna dopuszczalna liczba iteracji

Funkcja zwraca:

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

3.2 Opis metody

Metoda siecznych jest zbliżona w idei do metody Newtona, jednakże ma w pewnym sensie nad nią przewagę - nie musimy bowiem znać wartości funkcji pochodnej. Metoda siecznych bazuje na definicji pochodnej funkcji w punkcie:

$$f'(x_n) = \lim_{\Delta x \rightarrow 0} \frac{f(x_n + \Delta x) - f(x_n)}{\Delta x}$$

Co możemy zapisać jako:

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

Podstawmy otrzymany wzór na pochodną do wzoru na kolejny wyraz w metodzie Newtona:

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

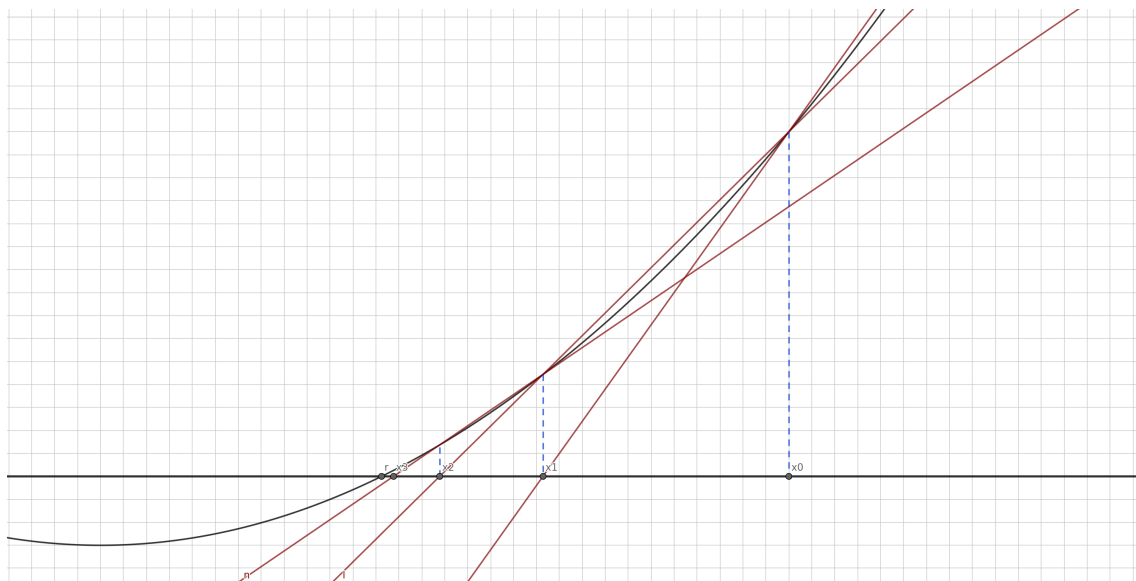
Problemy metody

1. Zbieżność: Ponieważ metoda siecznych korzysta w pewnym stopniu z metody Newtona to również jest zbieżna lokalnie, co oznacza, że początkowe przybliżenia x_0 oraz x_1 muszą być wybrane z dostatecznie małego otoczenia pierwiastka. W przeciwnym wypadku kolejne iteracje mogą prowadzić do coraz większego oddalania się od rzeczywistego rozwiązania. Aby tego uniknąć, w implementacji metody siecznych dodana jest maksymalna liczba iteracji, którą chcemy wykonać. Jeżeli po ich wykonaniu rozwiązanie nie spełnia wymagań dokładności, algorytm zwraca uzyskaną wartość wraz z informacją o niespełnieniu warunków zbieżności.

2. Warunek poprawności iteracji: Aby faktycznie zbliżać się wartością do pierwiastka funkcji, należy spełniać warunek $|f(x_n)| < |f(x_{n-1})|$. Jeśli ten warunek nie jest spełniony, wartości x_n oraz x_{n-1} należy zamienić miejscami.

3.3 Interpretacja graficzna metody

Działanie metody siecznych można łatwo zrozumieć, odwołując się do jej interpretacji graficznej, zaprezentowanej poniżej. Tworzymy sieczną przecinającą funkcję $f(x)$ w punktach $(x_0, f(x_0))$, $(x_1, f(x_1))$, a miejsce przecięcia jej z osią OX wyznacza wartość następnego przybliżenia x .



Rysunek 3: Przedstawienie graficzne iteracyjnego wyznaczania $f(x) = 0$ metodą siecznych

3.4 Pseudokod algorytmu

Implementacja metody siecznych została napisana na podstawie poniższego pseudokodu:

Algorithm 3: Algorytm metody siecznych

Data: $f, x_0, x_1, \delta, \epsilon, maxit$

Result: r, v, it, err

```

1  $f_{x0} \leftarrow f(x_0);$ 
2  $f_{x1} \leftarrow f(x_1);$ 
3 for  $k \leftarrow 1$  to  $maxit$  do
4   if  $|f_{x0}| > |f_{x1}|$  then
5      $x_0 \leftrightarrow x_1;$ 
6      $f_{x0} \leftrightarrow f_{x1};$ 
7    $s \leftarrow (x_1 - x_0) / (f_{x1} - f_{x0});$ 
8    $x_1 \leftarrow x_0;$ 
9    $f_{x1} \leftarrow f_{x0};$ 
10   $x_0 \leftarrow x_0 - f_{x0} * s;$ 
11   $f_{x0} \leftarrow f(x_0);$ 
12  if  $|x_1 - x_0| < \delta$  or  $|f_{x0}| < \epsilon$  then
13    return  $x_0, f_{x0}, k, 0;$ 
14 return  $x_0, f_{x0}, maxit, 1;$ 

```

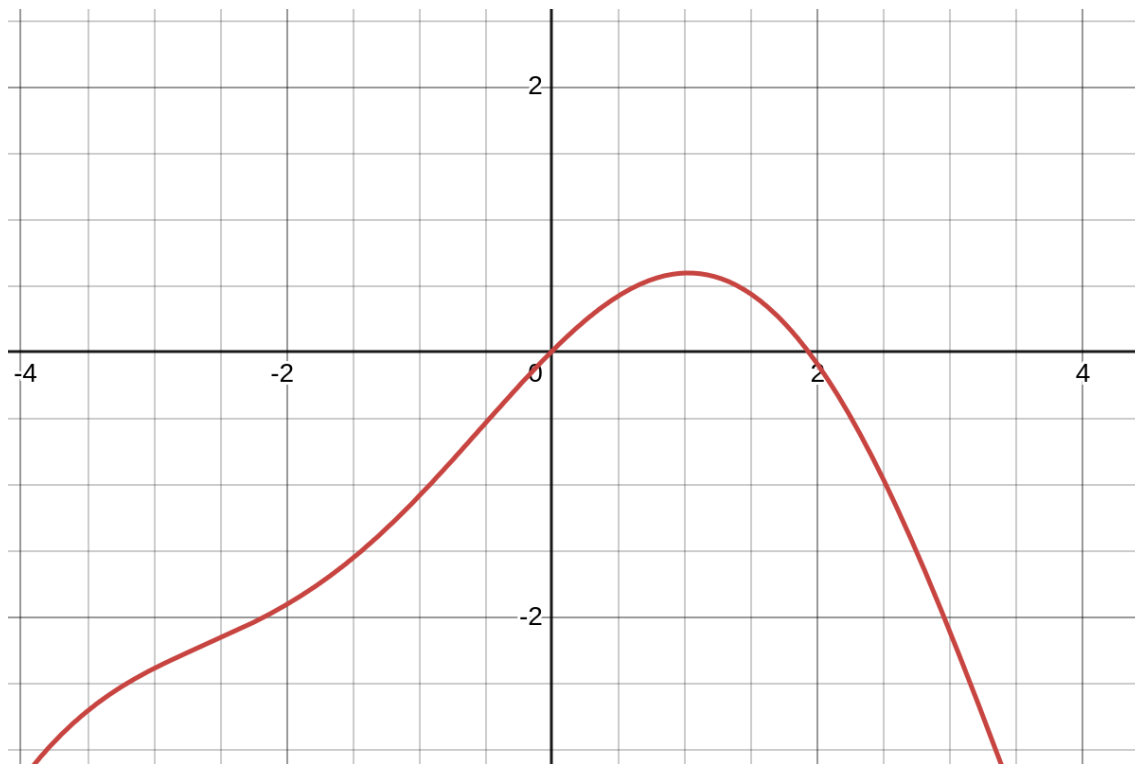
4 Zadanie 4

4.1 Opis problemu

Należy obliczyć pierwiastek równania $\sin(x) - (\frac{1}{2}x)^2$ trzema metodami:

1. bisekcji z przedziałem początkowym $[1.5, 2]$ i $\delta = \epsilon = \frac{1}{2}10^{-5}$
2. Newtona z przybliżeniem początkowym $x_0 = 1.5$ i $\delta = \epsilon = \frac{1}{2}10^{-5}$

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



Rysunek 4: Wykres funkcji $f(x) = \sin(x) - (\frac{1}{2}x)^2$

4.2 Rozwiązanie

Wykorzystane zostały wcześniej zaimplementowane metody dla danych:

```
f = x -> sin(x) - (0.5*x)^2
```

```
pf = x -> cos(x) - 0.5*x
```

```
delta = 0.5*10^(-5)
```

```
epsilon = 0.5*10^(-5)
```

Wywołania funkcji:

```
mbisekcji(f, 1.5, 2, delta, epsilon)
```

```
mstycznych(f, pf, 1.5, delta, epsilon, 20)
```

```
msiecznych(f, 1.0, 2.0, delta, epsilon, 20)
```

4.3 Wyniki

metoda	\tilde{r}	$f(\tilde{r})$	liczba iteracji	err
bisekcji	1.9337539672851562	-2.7027680138402843e-7	16	0
Newtona	1.933753779789742	-2.2423316314856834e-8	4	0
siecznych	1.933753644474301	1.564525129449379e-7	4	0

Tabela 1: Wartości przybliżenia pierwiastka funkcji dla podanych parametrów

4.4 Interpretacja wyników oraz wnioski

Jak można zauważyć, wszystkie trzy metody poprawnie przybliżyły wartość pierwiastka równania z oczekiwaną dokładnością dla podanych parametrów. Dla każdej z metod obliczona wartość pierwiastka jest identyczna do sześciu miejsc po przecinku. Najdokładniejsze \tilde{r} , dla którego wartość $f(\tilde{r})$ była najbliższa 0, uzyskała metoda Newtona.

Metoda bisekcji okazała się najwolniejsza, wymagając aż 16 iteracji, podczas gdy metody Newtona i siecznych potrzebowały jedynie 4 iteracji. Wynik ten jest zgodny z tym, że metoda bisekcji posiada liniowy wykładnik zbieżności, podczas gdy metoda Newtona ma wykładnik kwadratowy, a metoda siecznych – superliniowy.

Można więc dojść do wniosku, że jeśli posiadamy odpowiednią wiedzę o funkcji oraz możemy odpowiednio dobrać parametry metody, efektywniejszymi i szybszymi rozwiązaniami będą metoda Newtona lub metoda siecznych. Natomiast, jeśli ważne jest uzyskanie poprawnych wyników niezależnie od podanych parametrów, metoda bisekcji będzie lepszym i bardziej niezawodnym wyborem.

5 Zadanie 5

5.1 Opis problemu

Za pomocą metody bisekcji znaleźć wartość zmiennej x , dla której przecinają się wykresy funkcji $y = 3x$ i $y = e^x$. Wymagana dokładność obliczeń: $\delta = \epsilon = 10^{-4}$.

5.2 Rozwiązanie

Szukamy takiego x , że:

$$3x = e^x$$

Przekształćmy to wyrażenie:

$$3x - e^x = 0$$

Teraz zapiszmy to jako funkcję, której miejsca zerowe szukamy:

$$f(x) = 3x - e^x$$

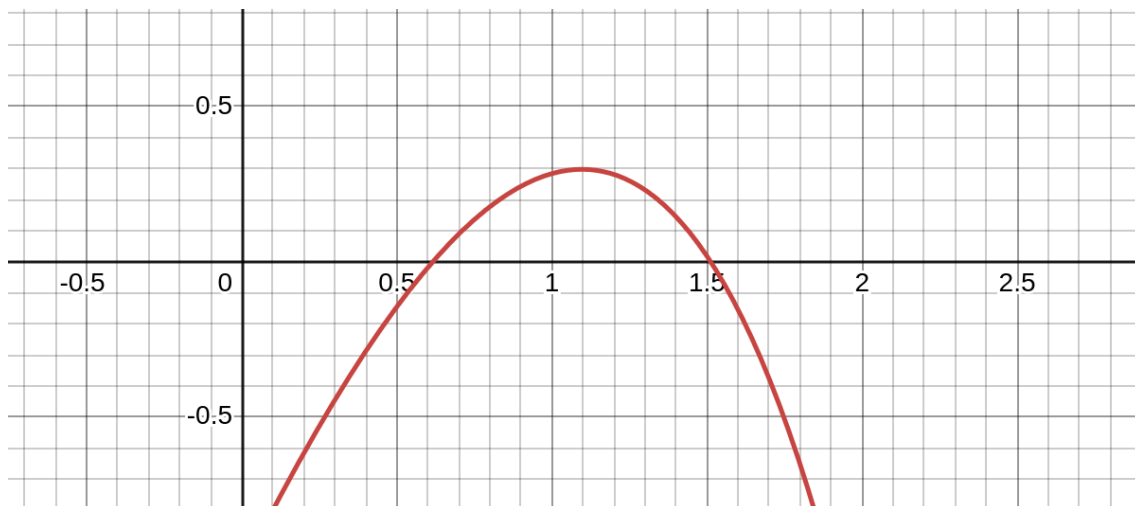
Sprawdźmy jak ta funkcja zachowuje się dla kilku wartości

$$\text{dla } x = 0: f(0) = 3 \cdot 0 - e^0 = 0 - 1 = -1$$

$$\text{dla } x = 1: f(1) = 3 \cdot 1 - e^1 = 3 - e > 0$$

$$\text{dla } x = 2: f(2) = 3 \cdot 2 - e^2 \approx 6 - (2.71828182846)^2 < 0$$

Można więc zauważyć, że zmiana znaku funkcji następuje na przedziałach $[0, 1]$ oraz $[1, 2]$.



Rysunek 5: Wykres funkcji $f(x) = 3x - e^x$

Obliczone wcześniej przedziały są zgodne z tym co widać na wykresie funkcji.

Do obliczeń została wykorzystana wcześniej zaimplementowana metoda bisekcji. Dane do znalezienia pierwszego miejsca zerowego:

```
f = x -> 3*x - (MathConstants.e)^x
a = 0.0
b = 1.0
delta = 10^(-4)
epsilon = 10^(-4)
```

Dane do znalezienia drugiego miejsca zerowego:

```
f = x -> 3*x - (MathConstants.e)^x
a = 1.0
b = 2.0
delta = 10^(-4)
epsilon = 10^(-4)
```

Wywołanie funkcji:

```
mbisekcji(f, a, b, delta, epsilon)
```

5.3 Wyniki

Otrzymane wyniki dla dobrze dobranych przedziałów $[0, 1]$ oraz $[1, 2]$:

a	b	\tilde{r}	$f(\tilde{r})$	liczba iteracji	err
0.0	1.0	0.619140625	9.066320343276146e-5	9	0
1.0	2.0	1.5120849609375	7.618578602741621e-5	13	0

Tabela 2: Wartości przybliżenia pierwiastka funkcji $f(x) = 3x - e^x$ dla przedziałów $[0, 1]$ oraz $[1, 2]$

Otrzymane wyniki dla źle dobranych przedziałów $[-10^{38}, 1]$ oraz $[1, 10^{38}]$:

a	b	\tilde{r}	$f(\tilde{r})$	liczba iteracji	err
-10^{38}	1.0	-7.174648137343063e-5	-1.000143495536464	140	0
1.0	10^{38}	1.512126384043548	1.2548601143969051e-5	139	0

Tabela 3: Wartości przybliżenia pierwiastka funkcji $f(x) = 3x - e^x$ dla przedziałów $[-10^{38}, 1]$ oraz $[1, 10^{38}]$

5.4 Interpretacja wyników i wnioski

Jak można zauważyć porównując otrzymane wyniki dla przedziału dobrze dobranego z tymi otrzymanymi przy użyciu dobranego niewłaściwie, dobranie odpowiednich parametrów jest kluczowe. Oczywiście jest, że wybór zbyt dużych przedziałów skutkuje znacznym wzrostem liczby iteracji potrzebnych do osiągnięcia wyniku. Jak się jednak okazuje, nieodpowiednie dobranie przedziału prowadzić do znaczącego błędu w wyznaczeniu pierwiastka.

Dla źle dobranego przedziału obliczona wartość pierwszego pierwiastka wynosi $-7.174648137343063 \cdot 10^{-5}$ zamiast oczekiwanej wartości w okolicach 0.619140625. Błąd ten jest spowodowany oczywiście ograniczoną precyzją arytmetyki. Jak wiadomo wartości o dużym module, takie jak na przykład 10^{38} , są przechowywane w standardzie IEEE 754 z bardzo ograniczoną dokładnością, a jako, że wykonujemy na nich różne działania to błędy związane z precyzją są tutaj nieuniknione.

Przykład ten jednoznacznie wskazuje, jak ważny jest właściwy dobór parametrów, takich jak przedział początkowy, w celu uzyskania poprawnych wyników oraz minimalizacji błędów związanych z ograniczeniami numerycznymi.

6 Zadanie 6

6.1 Opis problemu

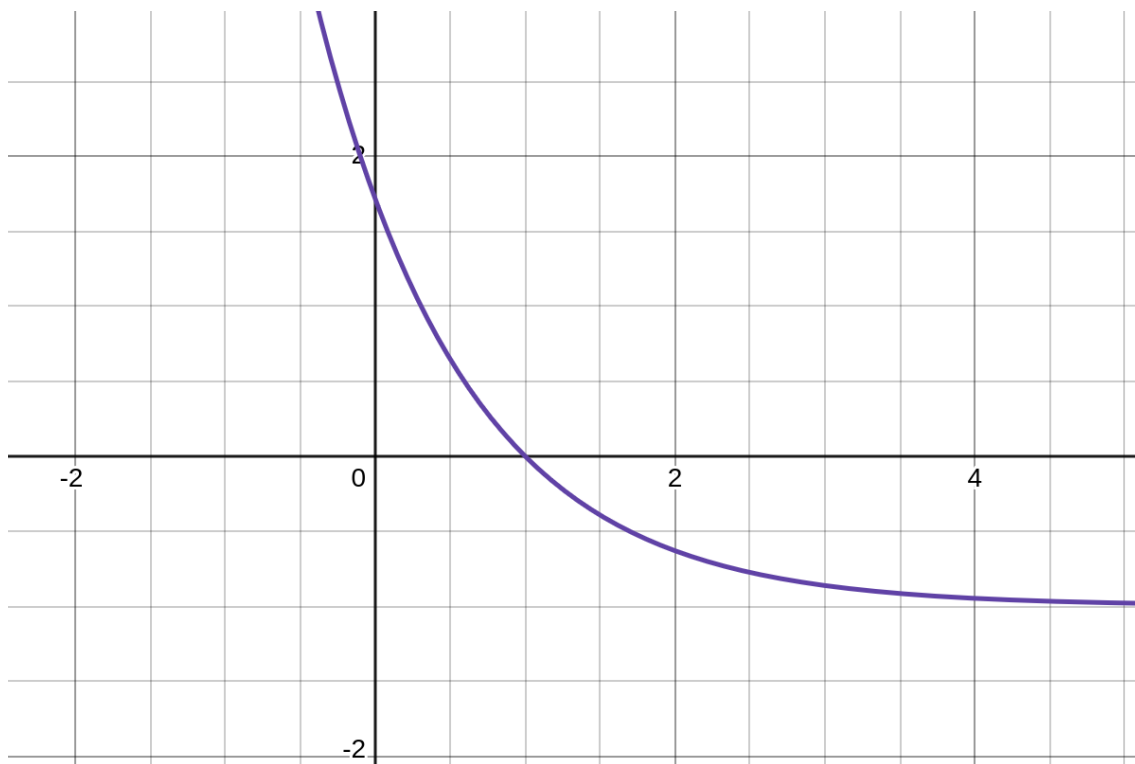
Znaleźć miejsce zerowe funkcji $f_1(x) = e^{1-x} - 1$ oraz $f_2(x) = xe^{-x}$ za pomocą metod bisekcji, Newtona i siecznych. Wymagana dokładność obliczeń: $\delta = \epsilon = 10^{-5}$. Dobrać odpowiednio przedział i przybliżenia początkowe.

Sprawdzić co się stanie, gdy w metodzie Newtona dla f_1 wybierzemy $x_0 \in (1, \infty]$, a dla f_2 wybierzemy $x_0 > 1$. Czy dla f_2 można wybrać $x_0 = 1$.

6.2 Rozwiązanie

6.2.1 Funkcja $f_1(x) = e^{1-x} - 1$

Oszacowanie miejsca zerowego: Funkcja $f_1(x)$ jest funkcją odwróconą eksponencjalną, na której zostało zastosowane przesunięcie o wektor $[1, -1]$. Można więc łatwo zauważyć, że miejscem zerowym tej funkcji jest $(1, 0)$.



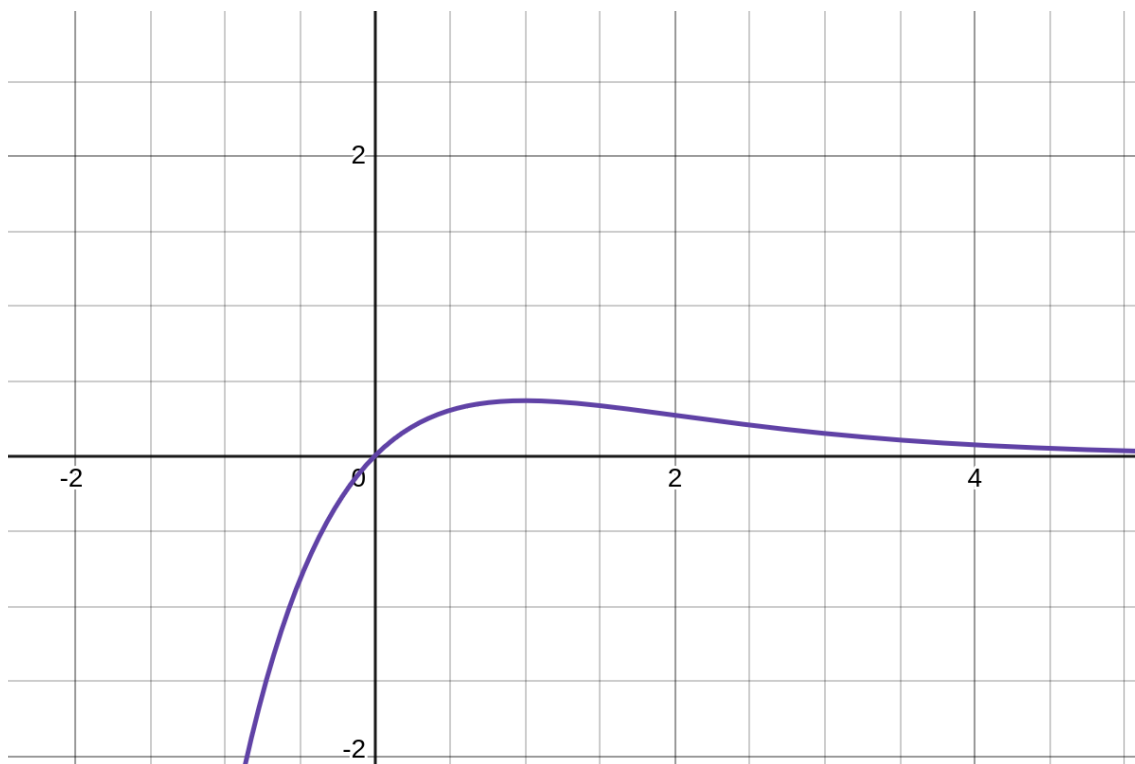
Rysunek 6: Wykres funkcji $f_1(x) = e^{1-x} - 1$

Obliczone miejsce zerowe jest zgodne z tym co widać na wykresie.

Wartość pochodnej funkcji: $f'_1(x) = -e^{1-x}$

6.2.2 Funkcja $f_2(x) = xe^{-x}$

Oszacowanie miejsca zerowego: W funkcji $f_2(x)$ mamy do czynienia z mnożeniem dwóch czynników: x oraz e^{-x} . Aby $f_2(x) = 0$ któryś z czynników musi mieć wartość 0. Jako, że dla każdego x : $e^{-x} > 0$ to jedynym miejscem zerowym jest $x = 0$.



Rysunek 7: Wykres funkcji $f_2(x) = xe^{-x}$

Jak można zauważyć, obliczone miejsce zerowe jest zgodne z tym co widać na wykresie.

Wartość pochodnej funkcji: $f'_2(x) = -e^{-x}(x - 1)$

6.3 Wyniki, ich interpretacja i wnioski

6.3.1 Funkcja $f_1(x) = e^{1-x} - 1$

Wyniki dla metody bisekcji:

a	b	\tilde{r}	$f(\tilde{r})$	liczba iteracji	err
0.0	2.0	1.0	0.0	1	0
0.1	2.0	1.0000038146972656	-3.814689989667386e-6	17	0
-1e38	1e38	1.0000024573828763	-2.457379856934949e-6	140	0
0.0	1000.0	1.0000020265579224	-2.026555868894775e-6	26	0
-11.0	2222.0	1.0000077337026596	-7.733672754639542e-6	26	0

Tabela 4: Wartości przybliżenia pierwiastka funkcji $f_1(x) = e^{1-x} - 1$ dla różnych przedziałów dla metody bisekcji

Jak widać, najlepsze, dokładne rozwiązanie otrzymaliśmy dla przedziału, dla którego rozwiązanie było jego dokładnym środkiem. Wybieranie takiego przedziału jest dobrą praktyką, jednakże gdy znamy dokładną wartość pierwiastka, to liczenie go za pomocą metod iteracyjnych jest z natury pozbawione sensu.

Gdy nie znamy dokładniej wartości pierwiastka dobrą praktyką jest więc wybieranie stosunkowo małego przedziału, w którym wiemy, że się on znajduje. Dla wszystkich testowanych przedziałów, także tych dużych, zwrócone wartości mieszczą się w przewidzianej dokładności, jednakże dla

mniejszych przedziałów wymagają one mniejszej liczby iteracji.

Wyniki dla metody Newtona:

x_0	\tilde{r}	$f(\tilde{r})$	liczba iteracji	err
-1000.0	NaN	NaN	20	1
-100.0	-80.0	1.5060973145850306e35	20	1
-10.0	0.999999998781014	1.2189871334555846e-10	15	0
-1.0	0.9999922654776594	7.734552252003368e-6	5	0
1.0	1.0	0.0	0	0
2.0	0.9999999810061002	1.8993900008368314e-8	5	0
10.0	NaN	NaN	20	1
100.0	100.0	-1.0	1	2
1000.0	1000.0	-1.0	1	2

Tabela 5: Wartości przybliżenia pierwiastka funkcji $f_1(x) = e^{1-x} - 1$ dla różnych przedziałów dla metody Newtona

Dla $x_0 \ll r$ np. $x_0 = -1000$: pochodna przyjmuje bardzo duże wartości przez co metoda staje się rozbieżna i nie potrafi znaleźć pierwiastka nawet po maksymalnej liczbie iteracji i zwraca wartości NaN.

Dla $x_0 \ll r$ np. $x_0 = -100$: metoda zaczyna stawać się zbieżna, aczkolwiek bardzo powoli. Dla zadanej maksymalnej liczby iteracji nie osiąga wyniku o wymaganej precyzji, ale po dalszym sprawdzeniu okazuje się że uzyskuje go dla 105 iteracji.

Dla $x_0 < r$: metoda jest zbieżna, im bliższe początkowe przybliżenie tym szybciej.

Dla $x_0 = r$: metoda nie wykonuje żadnej iteracji, bo od razu znajduje się w pierwiastku.

Dla $x_0 > r$ np. $x_0 = 2$: metoda nadal jest zbieżna.

Dla $x_0 > r$ np. $x_0 = 10$: wartość pochodnej jest bardzo mała co w efekcie powoduje błędy obliczeń i zwracanie wartości NaN.

Dla $x_0 \gg r$ np. $x_0 = 1000$: wartość pochodnej jest bardzo zbliżona do zera co w efekcie skutkuje zwróceniem od razu takiego kodu o błędzie.

Wyniki dla metody siecznych:

x_0	x_1	\tilde{r}	$f(\tilde{r})$	liczba iteracji	err
0.0	0.5	0.9999998133327657	1.8666725165594755e-7	5	0
0.0	2.0	1.0000017597132702	-1.7597117218937086e-6	6	0
0.1	2.0	1.0000008829446039	-8.829442140756427e-7	6	0
1.1	10000.0	1.1	-0.09516258196404048	2	0
0.8	100000.0	1.0000002623783855	-2.623783510458111e-7	11	0
-0.5	1000.0	NaN	NaN	20	1
-100.0	-10.0	-10.0	59873.14171519782	1	0
-100.0	1000.0	1000.0	-1.0	1	0

Tabela 6: Wartości przybliżenia pierwiastka funkcji $f_1(x) = e^{1-x} - 1$ dla różnych przedziałów dla metody siecznych

Jak widać, metoda siecznych jest nieco bardziej odporna na podawanie niewłaściwie dobranych wartości początkowych, niż metoda Newtona. Zwraca ona względnie poprawne wartości nawet dla dużych przedziałów.

Problemy pojawiają się tutaj w przypadku, gdy jedna z wartości początkowych jest dużą co do modułu liczbą ujemną. Wówczas przez to, że powstała sieczna jest prostą niemalże pionową, to zwracany jest wynik otrzymany po pierwszej iteracji, gdyż różnica kolejnych przybliżeń jest mniejsza od wymaganej dokładności.

Podobny problem pojawia się w sytuacji, gdy dla obu podanych przybliżeń funkcja przyjmuje podobną wartość - wówczas powstała styczna jest prawie pozioma, a metoda staje się rozbieżna i zwracany zostaje taki kod błędu.

6.3.2 Funkcja $f_2(x) = xe^{-x}$

Wyniki dla metody bisekcji:

a	b	\tilde{r}	$f(\tilde{r})$	liczba iteracji	err
-1.0	1.0	0.0	0.0	1	0
-0.9	1.0	3.8146972655607238e-6	3.8146827136732514e-6	17	0
-1e37	10.0	-7.174648137343063e-6	-7.174699613103618e-6	140	0
-1100.0	1000.0	475.0	2.4367709331973727e-204	2	0
-11.0	2222.0	1105.5	0.0	1	0

Tabela 7: Wartości przybliżenia pierwiastka funkcji $f_2(x) = xe^{-x}$ dla różnych przedziałów dla metody bisekcji

Mogło by się wydawać, że i w tym przypadku metoda bisekcji będzie wyjątkowo odporna na podawane wartości przedziałów. W ogólności jest to prawdą, jednakże w tym przypadku mamy do czynienia z funkcją, która od pewnego momentu zaczyna przyjmować wartości bardzo bliskie zera, aczkolwiek nigdy nie równe. W arytymetyce IEEE 754 nie da się jednak zapisywać coraz to mniejszych wartości w nieskończoność i od pewnego momentu są one zaokrąglane do zera. Stąd właśnie dla niektórych przedziałów zwracane są wartości pierwiastka bardzo oddalone od tej realnej. Problemem nie jest więc tutaj sama implementacja metody, tylko ograniczona precyzja arytmetyki.

Wyniki dla metody Newtona:

x_0	\tilde{r}	$f(\tilde{r})$	liczba iteracji	err
-1000.0	NaN	NaN	20	1
-100.0	-80.21919260199935	-5.533895786593711e36	20	1
-10.0	-3.784424932490619e-7	-3.784426364678097e-7	16	0
-1.0	-3.0642493416461764e-7	-3.0642502806087233e-7	5	0
-0.9	-4.179471505818752e-8	-4.179471680498576e-8	5	0
-0.1	-6.707074105306854e-9	-6.7070741502916976e-9	3	0
0.0	0.0	0.0	0	0
0.5	-3.0642493416461764e-7	-3.0642502806087233e-7	5	0
1.0	1.0	0.36787944117144233	1	2
10.0	14.380524159896261	8.173205649825554e-6	4	0
100.0	100.0	3.7200759760208363e-42	0	0
1000.0	1000.0	0.0	0	0

Tabela 8: Wartości przybliżenia pierwiastka funkcji $f_2(x) = xe^{-x}$ dla różnych przedziałów dla metody Newtona

Jak zauważyliśmy wcześniej, dla $x \gg 0$ wartości funkcji dążą do 0. Wraz z wzrostem x do zera dążą również wartości pochodnej.

Dla $x_0 \ll r$ np. $x_0 = -1000$: pochodna przyjmuje bardzo duże wartości przez co metoda

staje się rozbieżna i nie potrafi znaleźć pierwiastka nawet po maksymalnej liczbie iteracji i zwraca wartości NaN.

Dla $x_0 \ll r$ np. $x_0 = -100$: metoda zaczyna stawać się zbieżna, aczkolwiek bardzo powoli. Dla zadanej maksymalnej liczby iteracji nie osiąga wyniku o wymaganej precyzji, ale po dalszym sprawdzeniu okazuje się że uzyskuje go dla 108 iteracji.

Dla $x_0 < r$: metoda jest zbieżna, im bliższe początkowe przybliżenie tym szybciej.

Dla $x_0 = r$: metoda nie wykonuje żadnej iteracji, bo od razu znajduje się w pierwiastku.

Dla $x_0 > r$ np. $x_0 = 0.5$: metoda nadal jest zbieżna.

Dla $x_0 > r$ np. $x_0 = 1$: pochodna jest równa 0, stąd otrzymujemy odpowiedni błąd już w pierwszej iteracji.

Dla $x_0 > r$ np. $x_0 = 10$: metoda jest rozbieżna, a przez to, że zarówno funkcja jak i jej pochodna są bardzo bliskie 0 to zwracane wyniki są dalekie od poprawnych.

Dla $x_0 \gg r$ np. $x_0 = 1000$: wartość funkcji przez błędy arytmetyki wynosi 0, przez co obecna wartość x_0 jest od razu zwracana.

Wyniki dla metody siecznych:

x_0	x_1	\tilde{r}	$f(\tilde{r})$	liczba iteracji	err
1.0	0.5	8.76690178927691e-8	8.766901020691273e-8	9	0
0.0	100.0	0.0	0.0	1	0
-1.0	1.0	1.744165849924562e-8	1.7441658195034172e-8	18	0
-1.0	0.5	-1.1737426154042664e-6	-1.1737439930768023e-6	7	0
-1.0	100.0	100.0	3.7200759760208363e-42	1	0
-100.0	-90.0	-76.76062071385324	-1.6666809881044752e35	20	1
-100.0	-10.0	-10.0	-220264.65794806718	1	0
-100.0	1000.0	1000.0	0.0	1	0

Tabela 9: Wartości przybliżenia pierwiastka funkcji $f_2(x) = xe^{-x}$ dla różnych przedziałów dla metody siecznych

Dla przybliżeń początkowych znajdujących się stosunkowo blisko realnej wartości pierwiastka, metoda zachowuje się prawidłowo i zwraca poprawne wyniki.

Gdy jednym z przybliżeń początkowych jest realna wartość pierwiastka to metoda również od razu to wykrywa i zwraca tę wartość.

Problem pojawia się gdy podajemy wartość $x \gg r$. Wówczas, przez to, że w ograniczonej precyzji wartość funkcji wynosi wtedy 0 to jej wartość jest zwracana. Ponownie nie jest to problem implementacyjny tylko wynikający z ograniczonej precyzji arytmetyki.

Problemy pojawiają się tutaj w przypadku, gdy jedna z wartości początkowych jest dużą co do modułu liczbą ujemną. Wówczas przez to, że powstała sieczna jest prostą niemalże pionową, to zwracany jest wynik otrzymany po pierwszej iteracji, gdyż różnica kolejnych przybliżeń jest mniejsza od wymaganej dokładności.

Gdy podamy dwie duże pod względem modułu liczby ujemne jako pierwsze przybliżenia to metoda ta jest zbieżna, aczkolwiek bardzo wolno. Dla przykładu $x_0 = -100$ i $x_1 = -90$ metoda zbiega się dopiero po 142 iteracjach, co znacząco przekracza maksymalną wartość iteracji, dla której szukanie pierwiastka było testowane.

6.4 Ogólne wnioski

Jak można zauważyć, żadna z testowanych metod iteracyjnych nie jest całkowicie niezawodna. Nawet dla najbezpieczniejszej z metod - metody Newtona da się dobrać takie parametry i taką funkcję, aby zwracane wyniki były nieprawidłowe, jednakże zdarza się to zdecydowanie rzadziej.

Kluczową kwestią jest więc dobieranie odpowiednich parametrów, zwłaszcza dla metod Newtona i siecznych, które są zbieżne tylko lokalnie. Dobrą praktyką jest także zbadanie przebiegu funkcji i jej pochodnej, tego czy, w którymś momencie nie zaczyna ona dążyć do zera i na podstawie tego dobranie odpowiednich stosunkowo małych przedziałów. Należy mieć świadomość, że przez ograniczoną precyzję arytmetyki błędy w obliczeniach mogą wystąpić i nie jest to wina zastosowanej metody.