Lista nr 3

1. Problem znajdywania zer funkcji

Celem wszystkich algorytmów zawartych na tej liście jest numeryczne znajdywanie zer funkcji. Tego typu działanie jest często potrzebne w celu znajdywania rozwiązań równań nieliniowych, np. $3x = e^x$, $\sin x = \left(\frac{1}{2}x\right)^2$.

2. Metoda bisekcji

2.1. Pomysł

Działanie metody bisekcji wynika wprost ze znanego z analizy matematycznej *twierdzenia Darboux*.

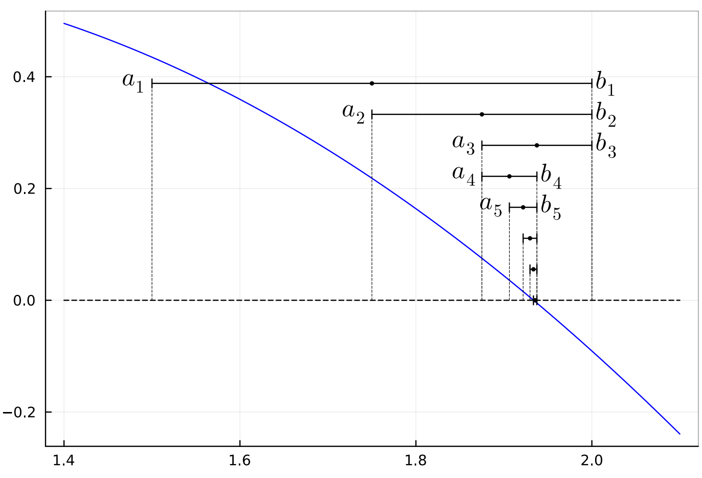
Twierdzenie Darboux Jeśli funkcja f jest ciągła na przedziale $[a, b]$, to dla każdej wartości $u \in [\min(f(a), f(b)), \max(f(a), f(b))]$ istnieje $c \in [a, b]$, że $f(c) = u$. Szczególnie jeśli $f(a) > 0 \wedge f(b) < 0$ albo $f(a) < 0 \wedge f(b) > 0$, to istnieje $c \in [a, b]$, że $f(c) = 0$.

Algorytm działa na zasadzie iteracyjnej - mając funkcję f i przedział $[a, b]$ na którego krańcach funkcja zmienia znak, wyznaczamy $c = \frac{a+b}{2}$, a następnie:

- Jeśli $f(c)$ jest wystarczająco blisko 0 lub a i b są wystarczająco blisko siebie - zwracamy c i kończymy
- Jeśli $f(c)$ i $f(a)$ mają różne znaki - nowy przedział to $[a, c]$
- Jeśli $f(c)$ i $f(b)$ mają różne znaki - nowy przedział to $[c, b]$

Tym samym w kolejnej iteracji algorytmu otrzymujemy kolejny przedział spełniający *twierdzenie Darboux*, ale który jest dwukrotnie węższy od poprzedniego. Algorytm kończymy, jeżeli przedział stanie się wystarczająco wąski lub c będzie wystarczająco blisko 0 względem żądanych dokładności obliczeń.

Metoda bisekcji charakteryzuje się liniową, ale globalną zbieżnością do miejsca zerowego funkcji.



Rysunek 1: Przykład użycia metody bisekcji dla znalezienia zer funkcji $f(x) = \sin(x) - \left(\frac{1}{2}x\right)^2$, z przedziałem początkowym $[1.5, 2.0]$

2.2. Możliwe problemy

W faktycznej implementacji algorytmu może pojawić się szereg problemów związanych z niedokładnością komputerowych obliczeń.

Nie jest dobrym pomysłem sprawdzać czy funkcja zmienia znak na krańcach przedziału $[a, b]$ poprzez sprawdzenie warunku $f(a) \cdot f(b) < 0$. W przypadku bardzo małych albo bardzo dużych wartości a i b może dojść do niedomiaru lub nadmiaru, co może spowodować błędne zawężenie przedziału. Lepszym pomysłem jest wykorzystanie szeroko dostępnej w językach programowania funkcji $\operatorname{sign}(x)$, która zwraca znak liczby x, i sprawdzanie warunku $\operatorname{sign}(a) \neq \operatorname{sign}(b)$.

Kolejnym problemem okazuje się szukanie środka przedziału $[a, b]$. Najprostszym sposobem jest wykonanie działania $\frac{a+b}{2}$. Jednak po wzięciu pod uwagę błędów obliczeń, okazuje się, że nie ma gwarancji, żeby wynik tego działania leżał między a i b. Zamiast tego liczymy początkową długość przedziału i w każdej kolejnej iteracji ją dwukrotnie zmniejszamy.

2.3. Pseudokod

Algorytm 1:

input funkcja $f(x)$, przedział $[a, b]$, żądane dokładności δ, ε
output przybliżenie r rozwiązania równania $f(x) = 0$

interval $= b - a$
if $\operatorname{sign}(f(a)) = \operatorname{sign}(f(b))$
    return „Przedział na krańcach nie zmienia znaku"
end
for $i \in [1, \text{max iterations}]$
    interval $:= \frac{\text{interval}}{2}$
    middle $:= a +$ interval
    if $|\text{interval}| < \delta$ **or** $|f(\text{middle})| < \varepsilon$, ▷ Czy osiągnięto żądaną dokładność?
        return middle
    end
    if $\operatorname{sign}(f(\text{middle})) \neq \operatorname{sign}(f(a))$
        $b :=$ middle ▷ $f(a)f(\text{middle}) < 0$
    else
        $a :=$ middle ▷ $f(b)f(\text{middle}) < 0$
    end
end

3. Metoda Newtona

3.1. Pomysł

Dla funkcji $f(x)$, która jest różniczkowalna, możemy znaleźć równanie prostej do niej stycznej w danym punkcie x_0.

$$y(x) = f'(x_0)(x - x_0) + f(x_0)$$

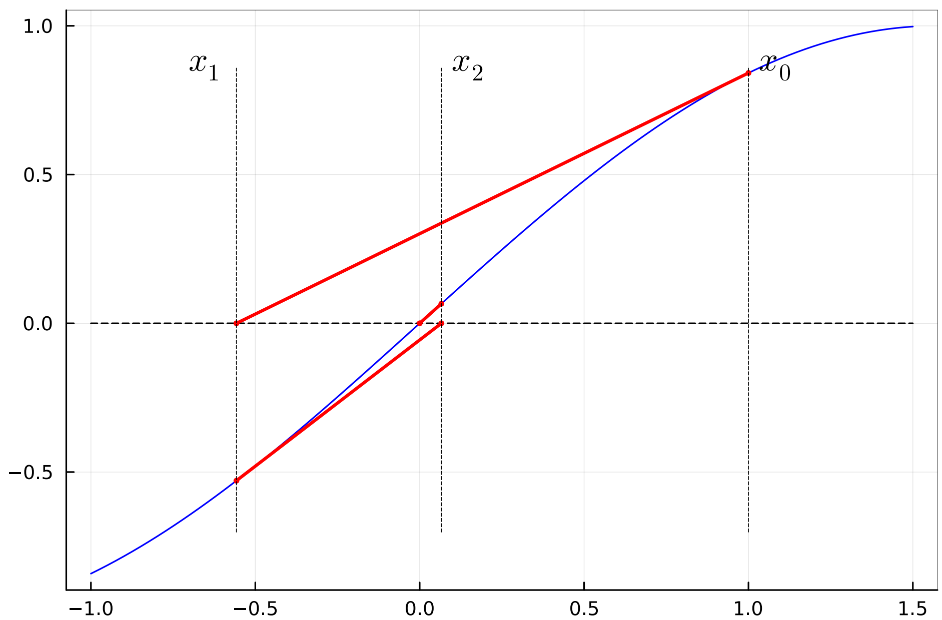
Ta styczna jest pewnym przybliżeniem funkcji $f(x)$ w pobliżu punktu x_0. Bardzo łatwo jest też znaleźć x, dla którego ta prosta przyjmuje wartość 0.

$$0 = f'(x_0)(x - x_0) + f(x_0) \longrightarrow x = x_0 - \frac{f(x_0)}{f'(x_0)}$$

W metodzie Newtona, nazywanej też metodą stycznych, zaczynamy z pewnym początkowym przybliżeniem rozwiązania x_0, które poprawiamy poprzez iteracyjne wyznaczanie stycznej do funkcji w tym punkcie i szukanie jej zera, które będzie kolejnym przybliżeniem rozwiązania.

Działanie algorytmu jest zakończone, gdy kolejne iteracje będą znajdywały się wystarczająco blisko siebie albo gdy wartość funkcji będzie bliska 0.

Metoda Newtona charakteryzuje się kwadratową zbieżnością do miejsca zerowego funkcji.

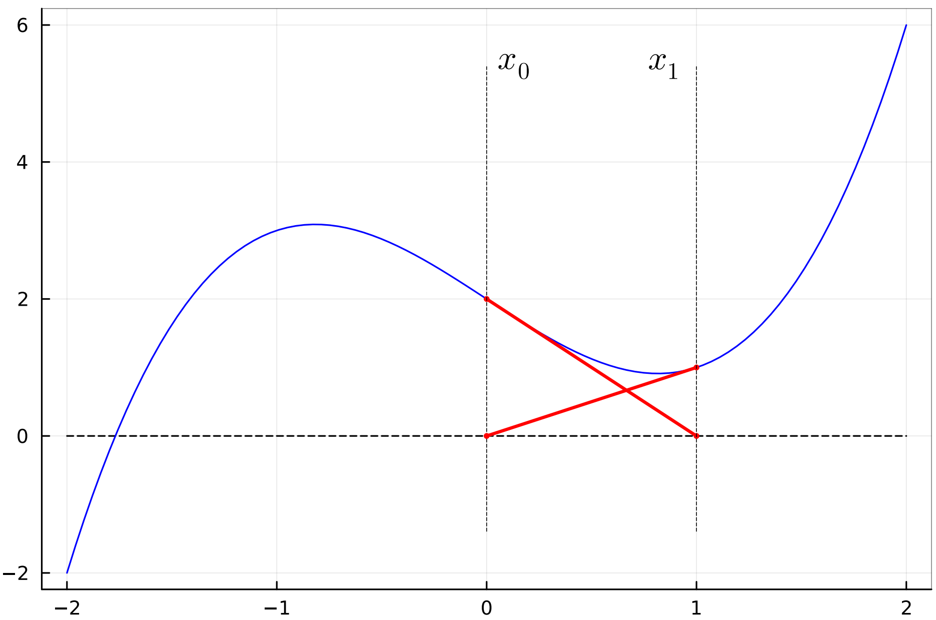


Rysunek 2: Przykład użycia metody stycznych dla znalezienia zer funkcji $f(x) = \sin(x)$, z przybliżeniem początkowym $x_0 = 1.0$

3.2. Możliwe problemy

Największym problemem tej metody jest to, że nie zawsze jest ona zbieżna. Zbieżność bardzo mocno zależy od wartości początkowego przybliżenia x_0.

Kolejnym problemem może być obliczanie pochodnej. Kiedy $f'(x)$ jest zerem, styczna jest równoległa do osi OX i jej nie przecina. Nawet gdy pochodna nie jest zerem, a jest jedynie bliska zeru, to istnieje ryzyko wprowadzenia bardzo dużego błędu.



Rysunek 3: Dla $f(x) = x^3 - 2x + 2$ i $x_0 = 0$ metoda stycznych wpada w cykl

3.3. Pseudokod

Algorytm 2:

input funkcja $f(x)$ i jej pochodna $f'(x)$, przybliżenie początkowe x_0, żądane dokładności δ, ε

output przybliżenie r rozwiązania równania $f(x) = 0$

if $|f(x_0)| < \varepsilon$
    return x_0
end
for $i \in [1, \text{max iterations}]$
    if $|f'(x_0)| < \varepsilon$
        return „Pochodna bliska zeru”
    end
    $x_1 := x_0 - \frac{f(x_0)}{f'(x_0)}$ ▷ Zero prostej stycznej do funkcji
    if $|x_1 - x_0| < \delta$ **or** $|f(x_1)| < \varepsilon$ ▷ Czy osiągnięto żądaną dokładność?
        return x_1
    end
    $x_0 := x_1$
end
return „Metoda nie zbiegła”

4. Metoda siecznych

4.1. Pomysł

Metoda Newtona wymagała, żeby dla danej funkcji osobno policzyć i podać jej pochodną. Metoda siecznych bierze się z tych samych obserwacji, co metoda Newtona, ale zamiast podawać pochodnej, jest ona w pewien sposób przybliżana.

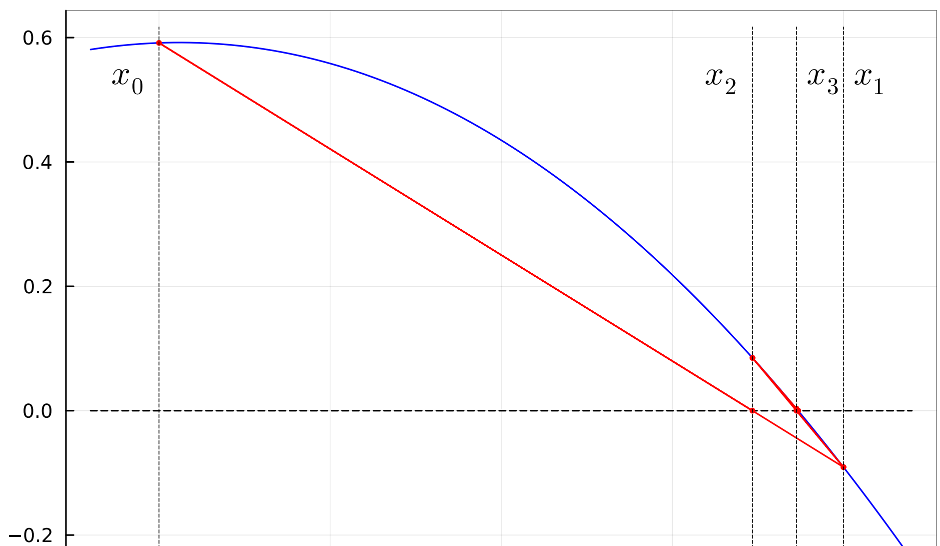
Dla tej metody zaczynamy z dwoma punktami x_0,x_1 będącymi początkowymi przybliżeniami.

$$f'(x) \approx \frac{f(x_1 + h) - f(x_1)}{h} \overset{x_0 = x_1 + h}{\longrightarrow} f'(x) \approx \frac{f(x_0) - f(x_1)}{x_0 - x_1}$$

Dodatkowo zawsze będziemy zakładali, że $|f(x_1)| \leq |f(x_0)|$ - w przeciwnym wypadku możemy je zamienić. To nam pozwoli zagwarantować, że dla ciągu przybliżeń x_n wartości $|f(x_n)|$ będą nierosnące.

Do następnej iteracji algorytmu bierzemy nowo otrzymane przybliżenie x_2 razem z x_1 z poprzedniej iteracji. Warunek końca pozostaje taki sam, jak w przypadku metody Newtona.

Metoda siecznych ma wykładnik zbieżności równy $\varphi \approx 1.618$.



Rysunek 4: Przykład użycia metody siecznych dla znalezienia zer funkcji $f(x) = \sin(x) - \left(\frac{1}{2}x\right)^2$, z przybliżeniami początkowymi $x_0 = 1.0$ i $x_1 = 2.0$

Graficznie algorytm jest realizowany poprzez narysowanie siecznej wykresu funkcji $f(x)$ przechodzącej przez punkty x_0 i x_1 i ustalenie kolejnego przybliżenia x_2 na przecięcie prostej z osią OX.

4.2. Pseudokod

Algorytm 3:

input funkcja $f(x)$, przybliżenia początkowe x_0, x_1, żądane dokładności δ, ε
output przybliżenie r rozwiązania równania $f(x) = 0$

for $i \in [1, \max \text{ iterations}]$
  if $|f(x_1)| > |f(x_0)|$
    swap x_0, x_1
  end
  $d := f(x_1) \cdot \frac{x_0 - x_1}{(f(x_0) - f(x_1))}$
  $x_0 := x_1$
  $x_1 := x_1 - d$
  if $|d| < \delta$ **or** $|f(x_1)| < \varepsilon$ ▷ Czy osiągnięto żądaną dokładność?
    return x_1
  end
end
return „Metoda nie zbiegła”

5. Zadanie 4

5.1. Opis

W tym zadaniu wykorzystuję zaimplementowane algorytmu do wyznaczenia pierwiastka równania $\sin(x) - \left(\frac{1}{2}x\right)^2 = 0$ z dokładnościami $\delta = \frac{1}{2}10^{-5}$, $\varepsilon = \frac{1}{2}10^{-5}$ oraz parametrami:

- dla metody bisekcji - przedział początkowy $[1.5, 2]$
- dla metody Newtona - przybliżenie początkowe $x_0 = 1.5$
- dla metody siecznych - przybliżenia początkowe $x_0 = 1$, $x_1 = 2$

5.2. Wyniki

n	mbisekcji	mstycznych	msiecznych
0	1.75	1.5	2.0
1	1.875	2.1403927723880054	1.8670388611329274
2	1.9375	1.952008946405626	1.9313545683871074
3	1.90625	1.933930573929843	1.9338445267485187
4	1.921875	1.933753779789742	1.933753644474301
5	1.9296875	...	...
6	1.93359375	...	...
7	1.935546875	...	...
8	1.9345703125	...	...
9	1.93408203125	...	...
10	1.933837890625	...	...
11	1.9337158203125	...	...
12	1.93377685546875	...	...
13	1.933746337890625	...	...
14	1.9337615966796875	...	...
15	1.9337539672851562	1.933753779789742	1.933753644474301

Tabela 1: Wartości x_n dla kolejnych iteracji

n	mbisekcji	mstycznych	msiecznych
0	0.218361	0.434995	0.0907026
1	0.0751795	0.303202	0.0849816
2	0.00496228	0.0243706	0.00316741
3	0.0358138	0.000233752	0.000119988
4	0.0156014	$2.24233 \cdot 10^{-8}$	$1.56453 \cdot 10^{-7}$
5	0.0053634	...	...
6	0.000211505	...	...
7	0.00237265	...	...
8	0.00107989	...	...
9	0.000434021	...	...
10	0.000111215	...	...
11	$5.01558 \cdot 10^{-5}$	...	...
12	$3.05269 \cdot 10^{-5}$	...	...
13	$9.81511 \cdot 10^{-6}$	...	...
14	$1.03558 \cdot 10^{-5}$	...	...
15	$2.70277 \cdot 10^{-7}$	$2.24233 \cdot 10^{-8}$	$1.56453 \cdot 10^{-7}$

Tabela 2: Wartości $|f(x_n)|$ dla kolejnych iteracji

5.3. Wnioski

Wszystkie metody z zadaną dokładnością odnalazły rozwiązanie tego równania. Różnica natomiast pojawiła się w szybkości zbiegania. Metoda bisekcji potrzebowała łącznie 15 iteracji, żeby osiągnąć żądaną dokładność, podczas gdy metody stycznych i siecznych potrzebowały jedynie 4, pomimo zaczynania w tym samym obszarze. Pokazuje to, że metoda bisekcji jest jedynie liniowo zbieżna, a metoda stycznych jest zbieżna kwadratowo.

6. Zadanie 5

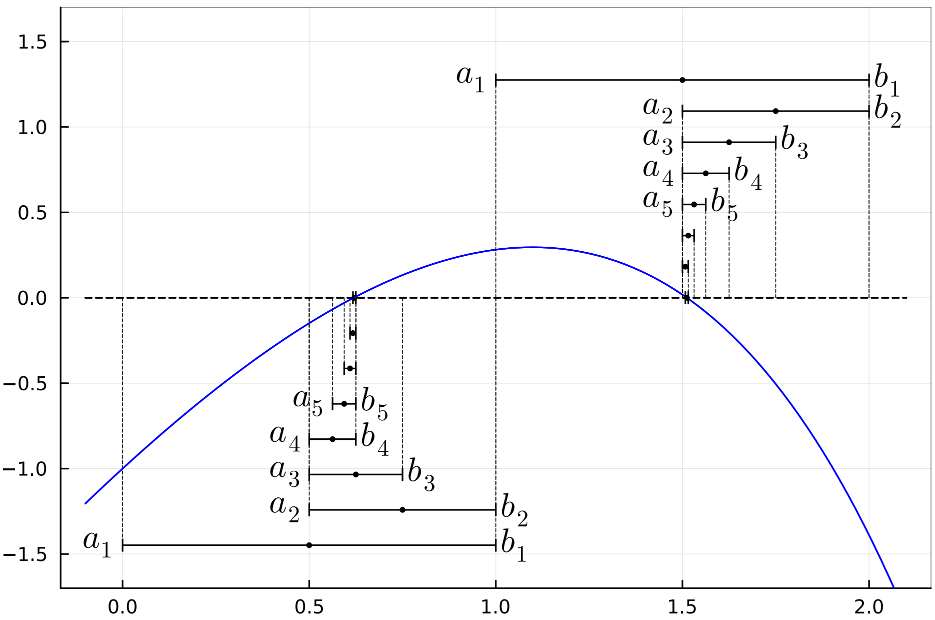
6.1. Opis

W tym zadaniu szukamy wartości zmiennej x, dla której przecinają się wykresy funkcji $y = 3x$ i $y = e^x$. Wymagane dokładności to $\delta = 10^{-4}$, $\varepsilon = 10^{-4}$.

6.2. Rozwiązanie

Po naszkicowaniu wykresu funkcji $f(x) = 3x - e^x$ zobaczyłem, że miejsca zerowe znajdują się w okolicach $x_1 \approx 0.5$, $x_2 \approx 1.5$, a pomiędzy nimi funkcja przyjmuje wartości dodatnie. Na tej podstawie wyznaczyłem początkowe przedziały dla metody bisekcji jako $[0.0, 1.0]$ i $[1.0, 2.0]$.

6.3. Wyniki



Rysunek 5: Metoda bisekcji zastosowana dla $f(x) = 3x - e^x$

Znaleziony pierwiastek	Wartość funkcji w pierwiastku	Liczba iteracji
0.619140625	$9.066320343276146 \cdot 10^{-5}$	9
1.5120849609375	$7.618578602741621 \cdot 10^{-5}$	13

Tabela 3: Dokładne rozwiązania znalezione przez metodę bisekcji

6.4. Wnioski

Przy pomocy metody bisekcji z łatwością można znaleźć rozwiązanie tego typu równania, przy założeniu, że wiemy gdzie funkcja przyjmuje wartości dodatnie, a gdzie wartości ujemne.

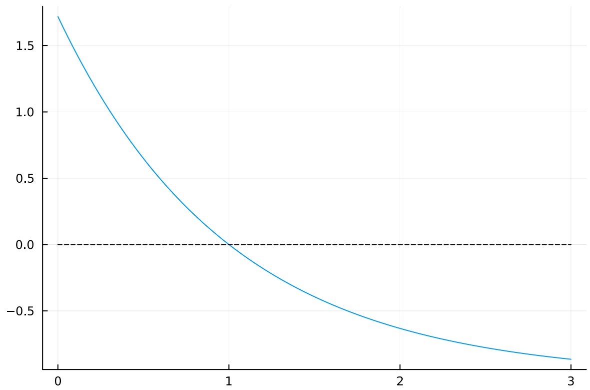
7. Zadanie 6

7.1. Opis

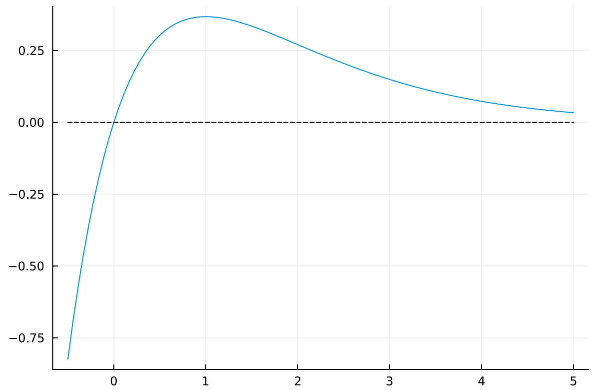
To zadanie polega na znalezieniu miejsc zerowych funkcji $f_1(x) = e^{1-x} - 1$ oraz $f_2(x) = xe^{-x}$ z zadanymi dokładnościami $\delta = 10^{-5}$, $\varepsilon = 10^{-5}$.

7.2. Rozwiązanie

Zacząłem od narysowania obu funkcji w celu wybrania odpowiednich parametrów początkowych.



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



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

Dla metody bisekcji nie ma żadnego problemu - dla f_1 wystarczy wybrać punkt po lewej i po prawej stronie 1. Tak samo dla f_2 - po lewej i po prawej stronie 0.

Dla metody Newtona i siecznych dla f_1 mogłoby się na początku wydawać, że wybór punktu początkowego nie ma znaczenia. Natomiast dla f_2 widać, że może występować problem gdy $x_0 \geq 1$.

7.3. Wyniki

7.3.1. $f_1(x) = e^{1-x} - 1$

Przedział początkowy	Znaleziony pierwiastek	Wartość funkcji	Liczba iteracji
$[0.0, 3.0]$	1.0000076293945312	$-7.629 \cdot 10^{-6}$	17
$[-10.0, 20.0]$	1.0000038146972656	$-3.815 \cdot 10^{-6}$	19
$[-100.0, 200.0]$	0.999993085861206	$6.914 \cdot 10^{-6}$	24
$[-1.0 \cdot 10^{11}, 2.0 \cdot 10^{11}]$	0.9999945316252479	$5.468 \cdot 10^{-6}$	54

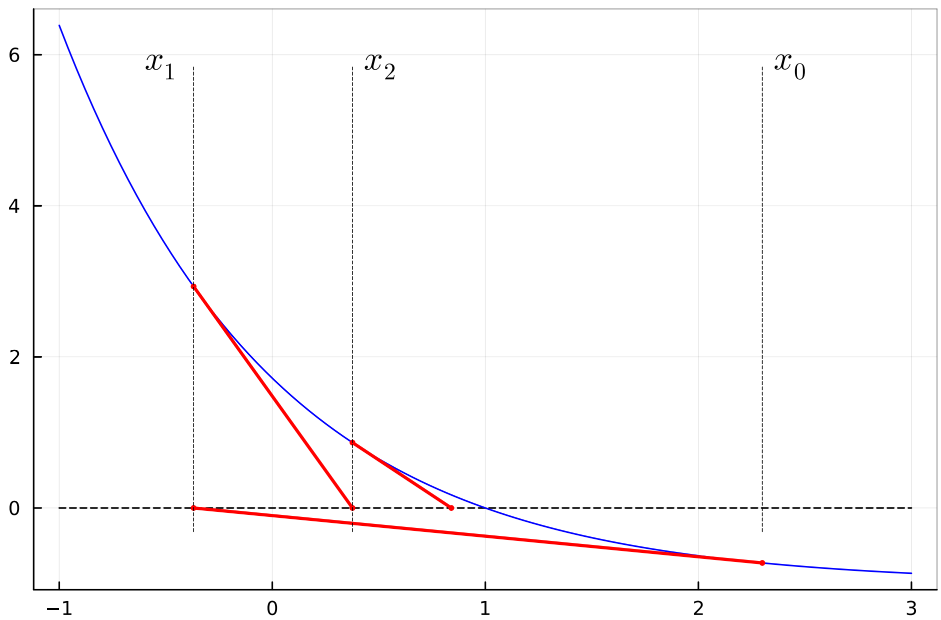
Tabela 4: Rozwiązania znalezione przez metodę bisekcji

Przybliżenie początkowe	Znaleziony pierwiastek	Wartość funkcji	Liczba iteracji
-2.0	0.9999999999251376	$7.48626 \cdot 10^{-11}$	7
0.0	0.9999984358892101	$1.56411 \cdot 10^{-6}$	4
2.0	0.9999999810061002	$1.89939 \cdot 10^{-8}$	5
4.0	0.9999999995278234	$4.72177 \cdot 10^{-10}$	21
7.0	0.9999999484165362	$5.15835 \cdot 10^{-8}$	401
8.0	`NaN`	`NaN`	`MAX`

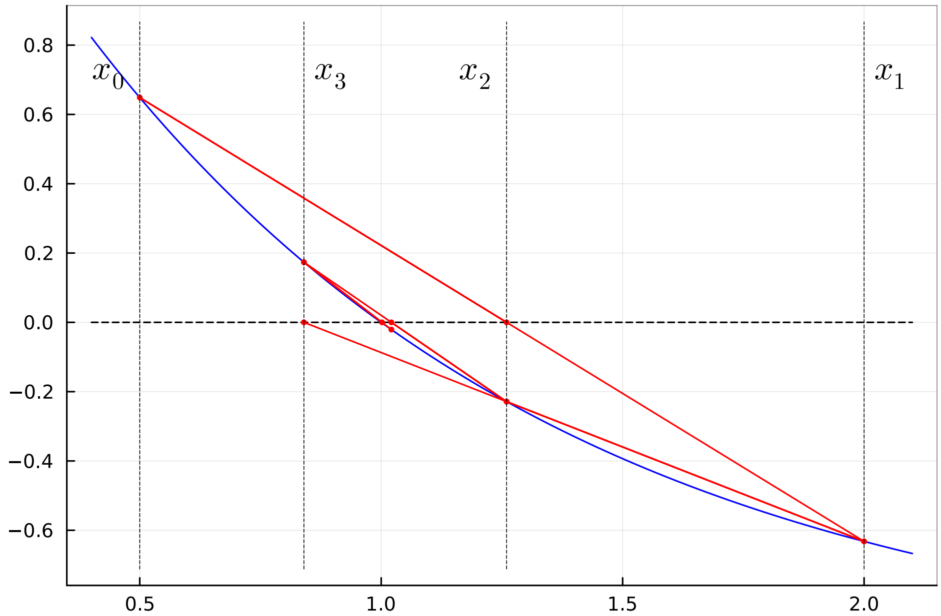
Tabela 5: Rozwiązania znalezione przez metodę Newtona

Przybliżenia początkowe	Znaleziony pierwiastek	Wartość funkcji	Liczba iteracji
$-1.0, 0.0$	0.9999990043764041	$9.95624 \cdot 10^{-7}$	6
$0.0, 0.5$	0.9999998133327657	$1.86667 \cdot 10^{-7}$	5
$1.5, 2.0$	1.0000034269838276	$-3.42698 \cdot 10^{-6}$	5
$3.0, 5.0$	1.0000002553381668	$-2.55338 \cdot 10^{-7}$	21

Tabela 6: Rozwiązania znalezione przez metodę siecznych



Rysunek 8: Metoda Newtona dla f_1 z $x_0 = 2.3$



Rysunek 9: Metoda Siecznych dla f_1 z $x_0 = 0.5$ i $x_1 = 2.0$

7.3.2. $f_2(x) = xe^{-x}$

Przedział początkowy	Znaleziony pierwiastek	Wartość funkcji	Liczba iteracji
$[-1.0, 2.0]$	$7.63 \cdot 10^{-6}$	$7.62934 \cdot 10^{-6}$	17
$[-10.0, 20.0]$	$-9.54 \cdot 10^{-6}$	$-9.53683 \cdot 10^{-6}$	20
$[-100.0, 200.0]$	50.0	$9.64375 \cdot 10^{-21}$	1

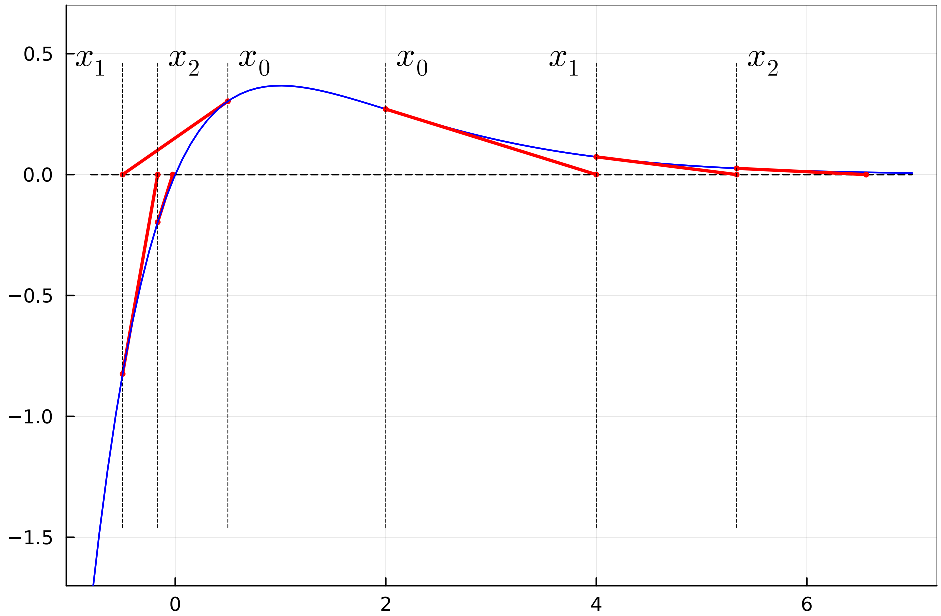
Tabela 7: Rozwiązania znalezione przez metodę bisekcji

Przybliżenie początkowe	Znaleziony pierwiastek	Wartość funkcji	Liczba iteracji
-1.0	$-3.06425 \cdot 10^{-7}$	$-3.06425 \cdot 10^{-7}$	5
0.99	$-3.99833 \cdot 10^{-6}$	$-3.99835 \cdot 10^{-6}$	107
1.0	`NaN`	`NaN`	`MAX`
1.01	102.01	$5.08467 \cdot 10^{-43}$	1
1.5	14.7874	$5.59488 \cdot 10^{-6}$	10

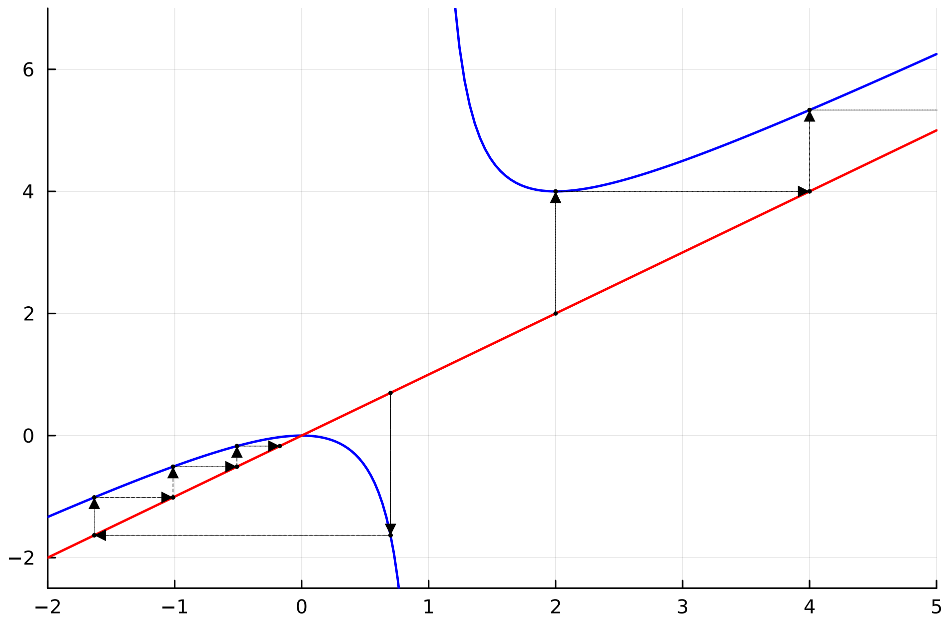
Tabela 8: Rozwiązania znalezione przez metodę Newtona

Przybliżenia początkowe	Znaleziony pierwiastek	Wartość funkcji	Liczba iteracji
$-1.0, -0.5$	$-1.223 \cdot 10^{-7}$	$-1.223 \cdot 10^{-7}$	6
$0.5, 1.0$	$8.7669 \cdot 10^{-8}$	$8.7669 \cdot 10^{-8}$	9
$0.5, 2.0$	14.4562	$7.61762 \cdot 10^{-6}$	1
$1.0, 2.0$	14.7877	$5.59375 \cdot 10^{-6}$	14

Tabela 9: Rozwiązania znalezione przez metodę siecznych



Rysunek 10: Metoda Newtona dla f_2 z $x_0 = 0.5$ i $x_0 = 2.0$



Rysunek 11: Iteracja graficzna metody Newtona dla f_2 z $x_0 = 0.7$ i $x_0 = 2.0$

7.4. Wnioski

Funkcje w tym zadaniu pokazują sytuację, w których poznane algorytmy nie potrafią sobie czasem poradzić.

Funkcja f_1 przyjmuje bardzo szybko, bardzo duże wartości dla $x < 1$. Już dla $x = -750$ następuje nadmiar podczas liczenia wartości funkcji. Natomiast dla $x > 1$ funkcja się wypłaszcza asymptotycznie do prostej $y = -1$.

- Metoda bisekcji jednak sobie z tym radzi - przez to, że sprawdzamy znak poprzez funkcję $\operatorname{sign}(x)$, nie obchodzi nas do końca ten nadmiar.

- Metoda Newtona dla $x_0 > 1$ zaczyna coraz wolniej zbiegać - staje się wolniejsza od metody bisekcji, aż w końcu dla $x_0 > 8$ występuję problem. Pochodna staje się na tyle mała, że kolejnym przybliżeniem rozwiązania jest liczba $x_1 \leq -750$. Dochodzi do nadmiaru, z którego już nie możemy się wydostać.

Funkcja f_2 wydaje się podobna, przyjmuje bardzo duże wartości dla $x < 0$, ale jednocześnie posiada maksimum dla $x = 1$ i dla $x > 1$ wypłaszcza się asymptotycznie do prostej $y = 0$. To powoduje problem dla naszych metod. W każdej z nich występuje warunek końca $|f(x)| < \varepsilon$. Ta funkcja jednak dla $x > 1$ przyjmuje dowolnie bliskie 0 wartości nigdy go nie przecinając.

- Metoda bisekcji polega dla zbyt dużych przedziałów. Algorytm przedwcześnie kończy działanie, gdy środek przedziału jest znacznie oddalony na prawo od 1.
- Metodę Newtona możemy zbadać, patrząc na wykres iteracji graficznej. Dla $x < 1$ metoda jest zbieżna. Problem jest dla $x = 0$, wtedy pochodna funkcji to 0 i od razu otrzymujemy błąd. Natomiast dla $x > 1$ metoda się rozbiega i, podobnie jak metoda bisekcji, zatrzymuje się na bardzo małej wartości $f(x)$, pomimo że nie jest nawet blisko miejsca zerowego funkcji.
- Metoda siecznych cierpi na te same problemy.