

Sprawozdanie z listy nr 3 Obliczenia Naukowe

Marek Świergoń (261750)

28 listopada 2022, PWR, WIT INA

1 Zadanie 1

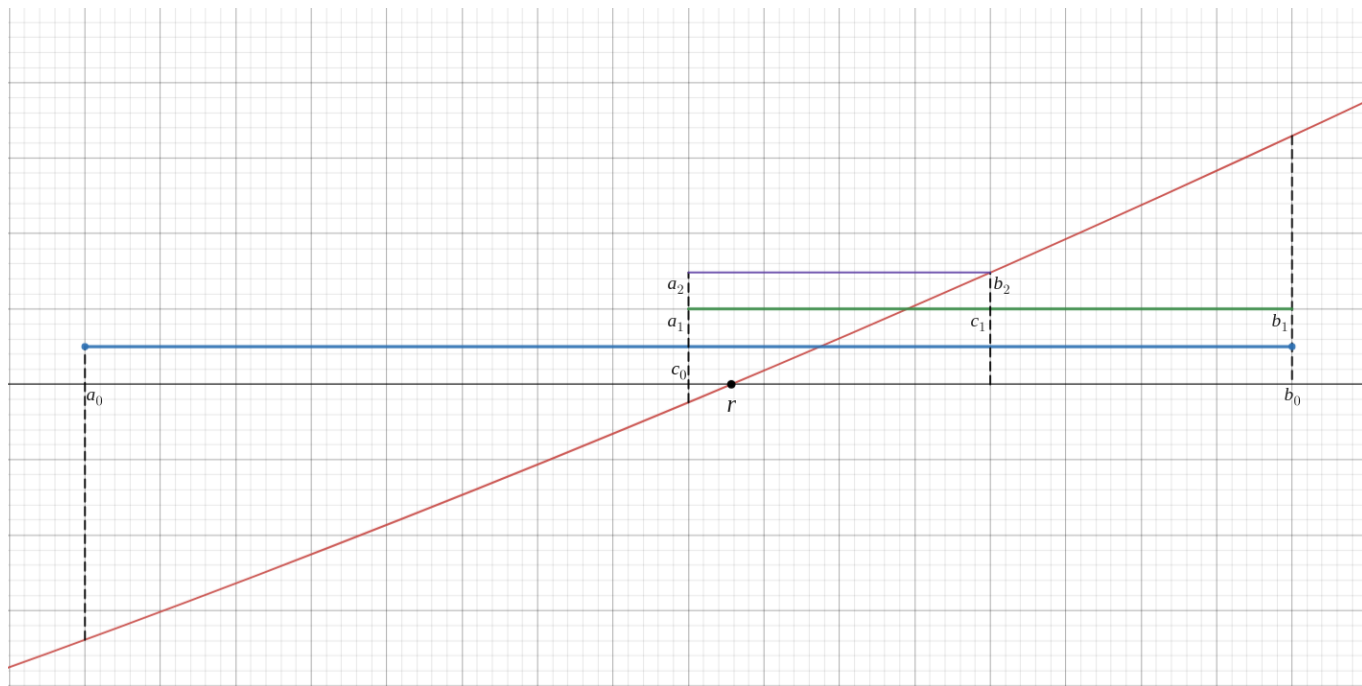
W tym zadaniu należało zaimplementować funkcję znajdującą miejsce zerowe zadanej funkcji f (znaleźć rozwiązanie równania $f(x) = 0$) metodą bisekcji. Aby zrozumieć implementację, w swoich słowach zobrazuję ideę stojącą za działaniem metody bisekcji. Implementacja funkcji została umieszczona w pliku *metody_iteracyjne.jl*, w module *MetodyIteracyjne*. Testy do tej funkcji znajdują się w pliku *tests.jl*.

1.1 Idea metody bisekcji

Metoda bisekcji jest jedną z iteracyjnych metod wyznaczania miejsca zerowego zadanej funkcji. Metoda ta pobiera dane a, b, δ, ϵ oraz funkcję f , dla której szukamy miejsca zerowego. Wartości a, b są krańcami przedziału, na którym wiemy, że funkcja f zmienia znak - jest to warunek konieczny do poprawnego działania tej metody. Dzięki niemu możemy skorzystać z twierdzenia Darboux, wedle którego dla takich a i b istnieje $r \in [a, b]$ taki, że $f(r) = 0$. Oczywiście wartość r będziemy chcieli wyznaczyć z zadaną dokładnością.

Aby to r coraz skuteczniej przybliżać, przedział $[a, b]$ będziemy zmniejszać w taki sposób, żeby w dalszym ciągu dla tych krańców przedziałów dalej można było stosować twierdzenie Darboux. W związku z tym, w każdej iteracji będziemy wyznaczać wartość ze środka przedziału c (dzielimy przedział na dwa, stąd metoda bisekcji; takie podejście jest najbardziej optymalne w przypadku, gdy nie wiemy bliżej którego krańca przedziału znajduje się szukane miejsce zerowe). Następnie sprawdzamy, czy $f(c)$ różni się znakiem z $f(a)$ czy z $f(b)$. Nowy przedział to będzie $[a, c]$ (gdy $f(c) * f(a) < 0$) lub $[c, b]$ (gdy $f(c) * f(b) < 0$). Połowienie przedziału kontynuujemy aż dojdziemy do zadanego maksymalnego odchylenia wartości funkcji od zera, czyli $|f(c)| < \epsilon$ lub aż dojdziemy do zadanego maksymalnego błędu bezwzględnego aproksymacji pierwiastka względem jego rzeczywistej wartości; nie wiemy oczywiście jakie jest dokładnie miejsce zerowe (chcemy to wyliczyć!), jednak ten błąd możemy łatwo ograniczyć z góry przez długość przedziału $[a, b] < \delta$.

Na rysunku 1. przedstawiono kilka pierwszych iteracji metody bisekcji.



Rysunek 1: Schemat działania metody bisekcji. Kolorem czerwonym narysowany jest wykres funkcji, której miejsca zerowe szukamy.

Należy zwrócić na dwie ważne kwestie z numerycznego punktu widzenia:

- Zamiast wyliczać $f(a) * f(c)$ (lub $f(b) * f(c)$) będziemy korzystać z faktu, że w arytmetyce zmiennopozycyjnej mamy jeden bit przeznaczony na reprezentację znaku. Wyliczanie powyższych iloczynów może być obciążone dużym błędem wynikającym z natury zastosowanej arytmetyki, zaś przyrównywanie bitów znaku takiego błędu nie ma. W języku Julia, aby dowiedzieć się o znaku korzystamy z metody `sign()`.
- Ze względu na fakt, że odejmowanie liczb bliskich sobie w arytmetyce zmiennopozycyjnej jest obciążone dużym błędem (zadanie źle uwarunkowane), to zamiast wyliczać za każdym razem środek przedziału ze wzoru $\frac{b-a}{2}$ będziemy korzystać z poprzednio wyliczonej długości i dzielić ją na dwa ($e_{new} = \frac{e}{2}$).

Algorithm 1 mbisekcji($f, a, b, \delta, \epsilon$)

```
fa ← f(a)
fb ← f(b)
e ← b − a
if sign(fa)=sign(fb) then
    return (Nothing, Nothing, Nothing, err=1) //błąd - funkcja nie zmienia znaku na przedziale [a,b]
end if
it ← 1
r ← a+e
v ← f(r)
while true do
    e ← e/2
    r ← a+e
    v ← f(r)
    if abs(e) < δ or abs(v) < ε then
        return (r, v, it, err=0)
    end if
    if sign(v) ≠ sign(fa) then
        b ← r
        fb ← v
    else
        a ← r
        fa ← v
    end if
    it ← it + 1
end while
```

2 Zadanie 2

W tym zadaniu należało zaimplementować funkcję znajdującą miejsce zerowe zadanej funkcji f (znaleźć rozwiązanie równania $f(x) = 0$) metodą Newtona. Aby zrozumieć implementację, w swoich słowach zobrazuję ideę stojącą za działaniem metody Newtona. Implementacja funkcji została umieszczona w pliku *metody_iteracyjne.jl*, w module *MetodyIteracyjne*. Testy do tej funkcji znajdują się w pliku *tests.jl*.

2.1 Idea metody Newtona (stycznych)

Aby zastosować metodę Newtona, funkcja f musi być dwukrotnie różniczkowalna. Dzięki temu będziemy mogli skorzystać z przybliżenia miejsca zerowego wynikającego z twierdzenia Taylora, tj.

$$f(r) = f(x_n + h) \approx f(x_n) + h * f'(x_n)$$

Oczywiście r jest poszukiwanym miejscem zerowym (czyli $f(r) = 0$), x_n jest przybliżeniem pierwiastka oraz $h = r - x_n$. Na podstawie tej wiedzy możemy stworzyć wzór wyznaczający kolejne przybliżenie pierwiastka x_{n+1} :

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

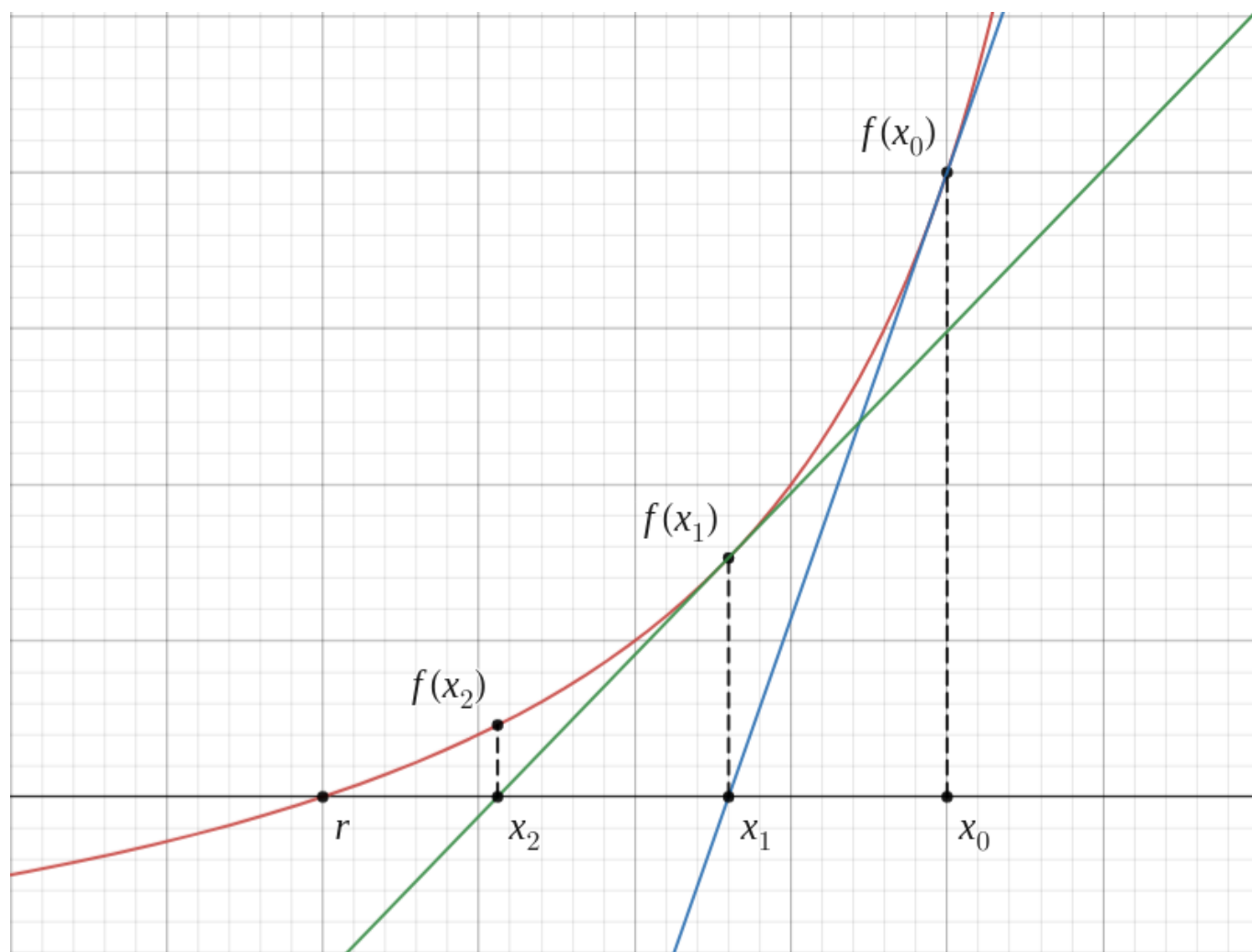
Mając taki wzór, możemy iteracyjnie wyznaczać kolejne przybliżenia, znając wartość funkcji w aktualnym przybliżeniu oraz wartość pochodnej funkcji f w tym samym przybliżeniu.

Metodę tą można zinterpretować również graficznie. Aby wyznaczyć x_{n+1} , tworzymy prostą styczną do wykresu funkcji w punkcie $(x_n, f(x_n))$ i patrzymy na punkt przecięcia tej prostej z osią OX - temu będzie równe x_{n+1} . Właśnie dlatego ta metoda nosi również nazwę metody stycznych.

Podobnie jak w metodzie bisekcji, możemy narzucić dokładność aproksymacji pierwiastka na dwa sposoby: względem wartości funkcji, tj. $|f(x_n)| < \epsilon$, oraz względem odległości dwóch ostatnich aproksymacji $|x_n - x_{n-1}| < \delta$. Niestety metoda Newtona jest zbieżna tylko lokalnie, co oznacza, że dla pewnych danych metoda Newtona może nie być zbieżna. W związku z tym, konieczne jest narzucenie kolejnego kryterium stopu: maksymalną liczbę iteracji. Po przekroczeniu tej liczby, metoda zwraca wyniki z ostrzeżeniem, że nie spełniają one narzuconej dokładności aproksymacji ($\text{err} = 1$).

Dodatkowo, ze względu na ograniczoną precyzję zastosowanej arytmetyki, metoda Newtona nie zadziała, gdy pochodna $f'(x_n)$ jest bliska zero. Wtedy uzyskamy błąd dzielenia przez zero, tj. wynikiem będzie NaN. Trzeba zatem uodpornić naszą implementację na ten problem - gdy napotkamy wartość pochodnej mniejszą od epsilon maszynowego, to zwracamy kod błędu informujący o tej sytuacji ($\text{err} = 2$).

Na Rysunku 2. widzimy kilka pierwszych iteracji metody Newtona. Wadą metody Newtona, jest konieczność wyznaczania wartości pochodnej $f'(x)$ - nie wszystkie funkcje dają się łatwo zróżniczkować, a samo wyliczenie wartości pochodnej może być czasochłonne.



Rysunek 2: Schemat działania metody Newtona (stycznych). Kolorem czerwonym narysowany jest wykres funkcji, której miejsca zerowego szukamy.

Algorithm 2 mstycznych(f , pf , x_0 , δ , ϵ , maxit)

```
v ← f(x0)
if abs(v) < ε then
    return (x0, v, 0, err=0)
end if
for it = 1, ..., maxit do
    pfx0 ← pf(x0)
    if abs(pfx0) < macheps() then
        return (x0, v, it, err=2) //błąd - pochodna bliska zeru
    end if
    x1 ← x0 - (v / pfx0)
    v ← f(x1)
    if abs(x1-x0) < δ or abs(v) < ε then
        return (r, v, it, err=0)
    end if
    x0 ← x1
end for
return (x0, v, maxit, err=1) //ostrzeżenie o przekroczeniu liczby iteracji
```

3 Zadanie 3

W tym zadaniu należało zaimplementować funkcję znajdującą miejsce zerowe zadanej funkcji f (znaleźć rozwiązanie równania $f(x) = 0$) metodą siecznych. Aby zrozumieć implementację, w swoich słowach zobrazuję ideę stojącą za działaniem metody siecznych. Implementacja funkcji została umieszczona w pliku *metody_iteracyjne.jl*, w module *MetodyIteracyjne*. Testy do tej funkcji znajdują się w pliku *tests.jl*.

3.1 Idea metody siecznych

Metoda siecznych poniekąd bazuje na metodzie Newtona, jednak nie jest obarczona jedną trudnością - nie musimy znać wartości pochodnej $f'(x)$. Wynika to z zastosowania definicji pochodnej funkcji f w punkcie x_n :

$$f'(x_n) = \lim_{h \rightarrow 0} \frac{f(x+h) - f(x)}{h} \approx \frac{f(x_n) - f(x_{n-1})}{x_n - x_{n-1}}$$

Teraz podstawiając taką aproksymację pochodnej do metody Newtona, otrzymujemy

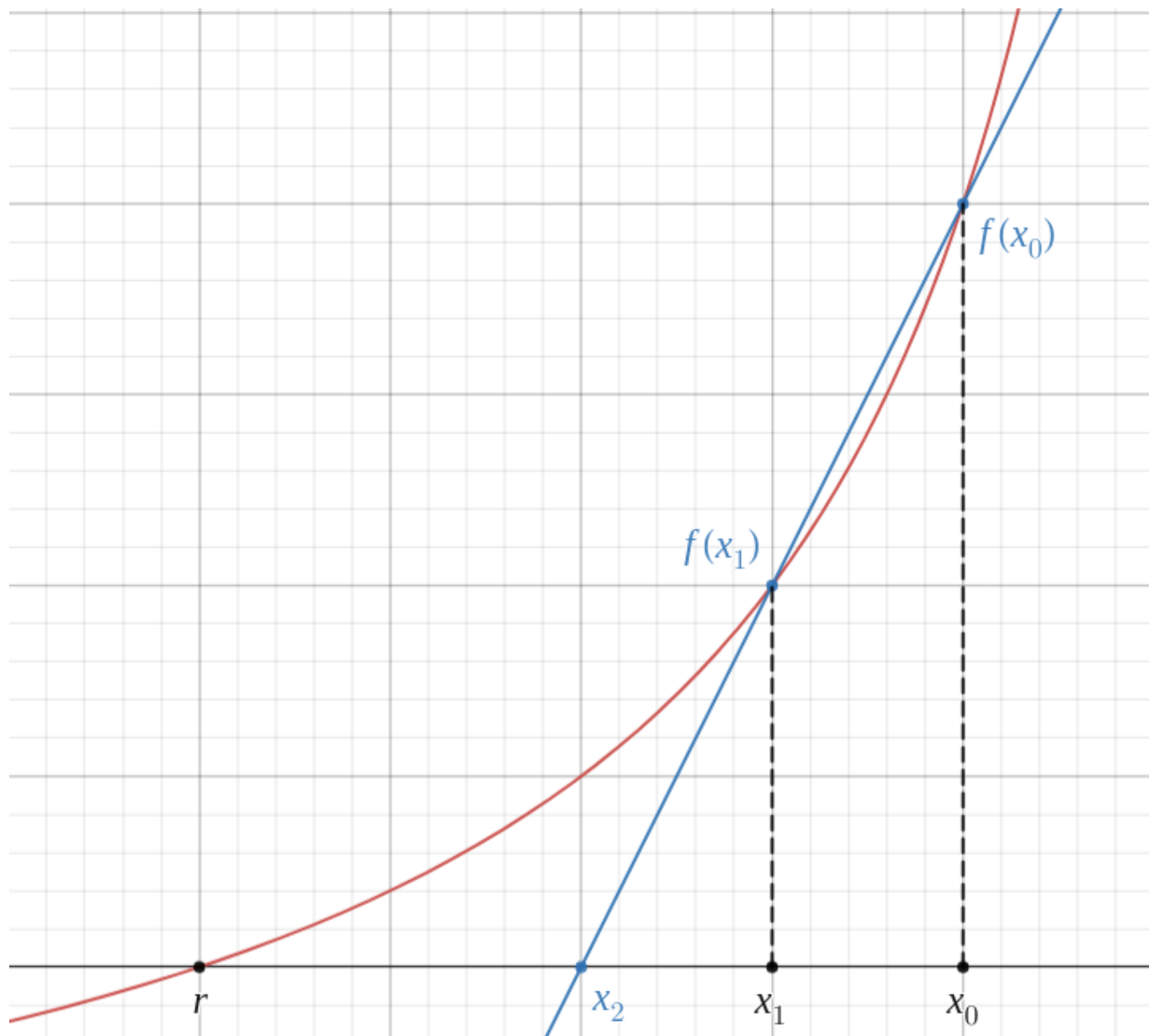
$$x_{n+1} = x_n - f(x_n) * \frac{x_n - x_{n-1}}{f(x_n) - f(x_{n-1})}.$$

Widzimy zatem, że zamiast wyznaczać wartość pochodnej $f'(x)$ musimy mieć zapamiętaną wartość nie tylko bieżącej aproksymacji, ale i poprzedniej, tj. x_{n-1} . Zapamiętanie tej wartości nie stanowi większego problemu, jedyne co może być problemem, to podanie dobrych aproksymacji wstępnych na początku działania metody, czyli x_0 i x_1 . Niestety metoda siecznych ma zwykle wykładnik zbieżności mniejszy od metody Newtona, bo równy $\gamma \approx 1.618$ (dla metody Newtona zazwyczaj wynosi on $\gamma = 2$).

Ponieważ metoda siecznych wynika częściowo z metody Newtona, to nie jest ona również zbieżna globalnie; możemy mieć tylko pewność co do jej lokalnej zbieżności dla pewnego ϵ -otoczenia r . W związku z tym istnieją przypadki, dla których metoda ta będzie rozbiegać, stąd konieczność narzucenia dodatkowego kryterium stopu, jakim jest maksymalna liczba iteracji. Po przekroczeniu tej liczby, zwracamy wyniki z ostrzeżeniem, że nie spełniają one narzuconej dokładności aproksymacji ($\text{err} = 1$).

Nazwa metody siecznej wynika z jej geometrycznego zobrazowania. Przedstawia je Rysunek 3., stworzymy sieczną przecinającą wykres funkcji f w dwóch punktach, $(x_{n-1}, f(x_{n-1}))$, $(x_n, f(x_n))$. Za nową aproksymację x_{n+1} przyjmujemy miejsce przecięcia tej siecznej z osią OX.

Dodatkowo, w implementacji ważne jest założenie, że $|f(x_n)| < |f(x_{n-1})|$, dzięki niemu zabezpieczamy się, że schodzimy aproksymacją rzeczywiście w stronę szukanego miejsca zerowego.



Rysunek 3: Schemat działania metody siecznych. Kolorem czerwonym narysowany jest wykres funkcji, której miejsca zerowego szukamy.

Algorithm 3 msiecznych($f, x_0, x_1, \delta, \epsilon, \text{maxit}$)

```
fx0  $\leftarrow f(x_0)$ 
fx1  $\leftarrow f(x_1)$ 
for it = 1, ..., maxit do
  if abs(fx1) > abs(fx0) then
    swap(x0, x1)
    swap(fx0, fx1)
  end if
  s  $\leftarrow (x_0 - x_1) / (fx_0 - fx_1)$ 
  x0  $\leftarrow x_1$ 
  fx0  $\leftarrow fx_1$ 
  x1  $\leftarrow x_1 - fx_1 * s$ 
  fx1  $\leftarrow f(x_1)$ 
  if abs(x0-x1) <  $\delta$  or abs(fx1) <  $\epsilon$  then
    return (x1, fx1, it, err=0)
  end if
end for
return (x1, fx1, maxit, err=1) //ostrzeżenie o przekroczeniu liczby iteracji
```

4 Zadanie 4

4.1 Cel zadania

Wyznaczyć pierwiastek następującego równania:

$$\sin(x) - \left(\frac{1}{2}x\right)^2 = 0,$$

z użyciem zaimplementowanych w zadaniach 1-3 metod:

- bisekcji z przedziałem początkowym $[1.5, 2]$ i $\delta = \epsilon = 0.5 * 10^{-5}$,
- Newtona z przybliżeniem początkowym $x_0 = 1.5$ i $\delta = \epsilon = 0.5 * 10^{-5}$,
- siecznych z przybliżeniami początkowymi $x_0 = 1, x_1 = 2$ i $\delta = \epsilon = 0.5 * 10^{-5}$.

4.2 Rozwiązanie

Wykorzystując zaimplementowane w module `MetodyIteracyjne` metody, odpowiednio

`mbisekcji(f, 1.5, 2.0, delta, epsilon),`

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

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

Oczywiście w metodzie Newtona i siecznych liczba maksymalnych iteracji została wyznaczona eksperymentalnie, po upewnieniu się, że metody dla tego równania zachowują się przyzwoicie. Ponadto

$$f(x) = \sin(x) - \left(\frac{1}{2}x\right)^2 = 0,$$

$$pf(x) = \cos(x) - \frac{x}{2}.$$

4.3 Wyniki

Rodzaj metody	\tilde{r}	$f(\tilde{r})$	liczba iteracji	kod err
bisekcji	1.9337539672851562	$-2.7027680138402843 * 10^{-7}$	16	0
Newtona	1.933753779789742	$-2.2423316314856834 * 10^{-8}$	4	0
siecznych	1.933753644474301	$1.564525129449379 * 10^{-7}$	4	0

Tabela 1: Wartości aproksymacji pierwiastka zadanego w zadaniu równania, dla zadanych w zadaniu metod iteracyjnych i parametrów początkowych.

4.4 Interpretacja wyników i wnioski

Zauważmy, że każda z metod zachowała się poprawnie dla zadanej funkcji f i zadanych parametrów początkowych - każda metoda zwróciła kod **err** = 0, brak błędów czy ostrzeżeń. Ponadto, metoda bisekcji była najwolniejsza, potrzebowała aż 16 iteracji na ustalenie wyniku. Metoda Newtona i siecznych potrzebowały tylko 4 iteracje (czterokrotnie mniej!). Wynika to z wykładnika zbieżności tych funkcji: metoda bisekcji jest zbieżna **liniowo**, metoda Newtona zazwyczaj **kwadratowo**, zaś metoda siecznych **superliniowo**.

Możemy zatem wysunąć wniosek, że sprawniejsze dla tego zadania są metody Newtona i siecznych. Funkcja pochodnej w tym przypadku jest łatwa do wyznaczenia, a jej wartości są łatwe do wyliczenia, ponadto metoda Newtona dała w tym przypadku najdokładniejszy wynik. W związku z tym, dla tego przykładu najlepsza okazała się metoda Newtona, nie jest tak jednak zawsze. Metody siecznych i siecznych mogą zawieść (brak zbieżności globalnej), metoda bisekcji jest najwolniejszą, ale i najbezpieczniejszą metodą.

5 Zadanie 5

5.1 Cel zadania

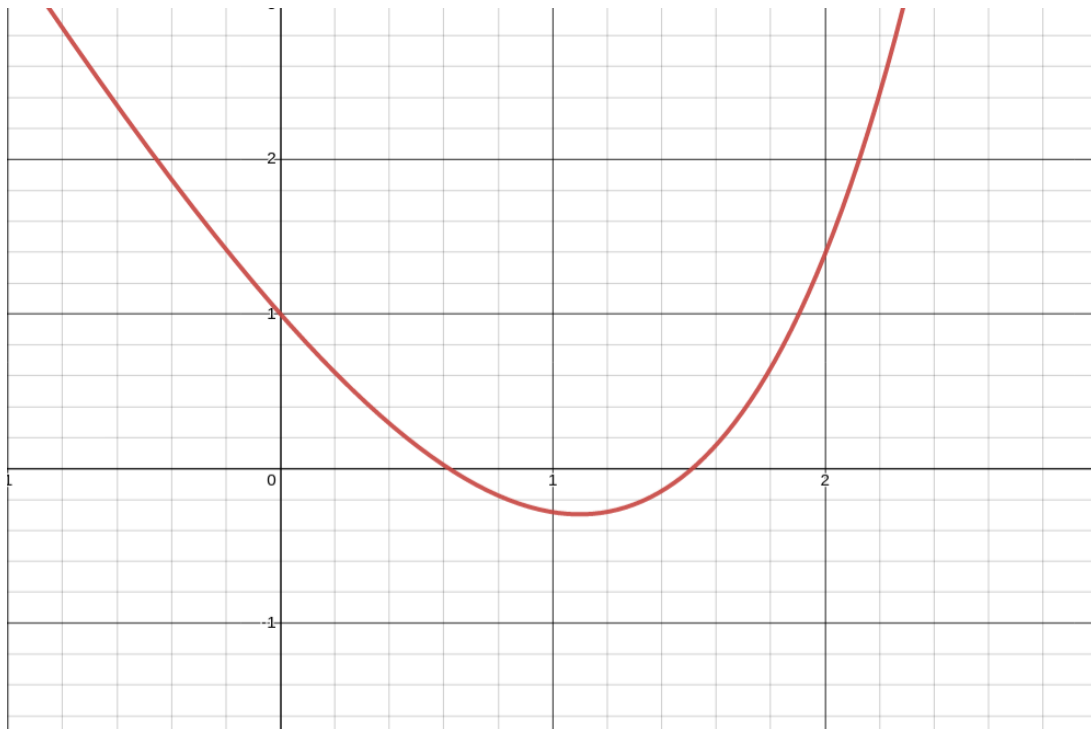
Metodą bisekcji znaleźć wartości zmiennej x , dla której przecinają się wykresy funkcji $y = 3x$ i $y = e^x$. Wymagana dokładność obliczeń to $\delta = \epsilon = 10^{-4}$.

5.2 Rozwiązanie

Aby móc skorzystać z metody bisekcji, muszę przekształcić ten problem do problemu szukania miejsc zerowych zadanych funkcji. Zatem

$$\begin{aligned} 3x &= e^x, \\ f(x) &= e^x - 3x = 0. \end{aligned}$$

Wykres funkcji f został przedstawiony na Rysunku 4.



Rysunek 4: Wykres funkcji $f(x) = e^x - 3x$, dla której szukamy miejsc zerowych.

Widzimy zatem, że poszukujemy dwóch miejsc zerowych.

Dla pierwszego z lewej pierwiastka, zauważam, że dla $a = 0.5$ $f(a) \approx 0.15$, dla $b = 1$ $f(b) \approx -0.28$, czyli funkcja zmienia znak na przedziale $[a, b]$. Wywołuję zatem `mbisekcji(f, 0.5, 1.0, delta, epsilon)`.

Dla drugiego z lewej pierwiastka, zauważam, że dla $a = 0.1$ $f(a) \approx -0.28$, dla $b = 2$ $f(b) \approx 1.39$, czyli funkcja zmienia znak na przedziale $[a, b]$. Wywołuję zatem `mbisekcji(f, 1.0, 2.0, delta, epsilon)`.

5.3 Wyniki

$[a, b]$	\tilde{r}	$f(\tilde{r})$	liczba iteracji	kod <code>err</code>
$[0.5, 1]$	0.619140625	$-9.066320343276146 * 10^{-5}$	8	0
$[1, 2]$	1.5120849609375	$-7.618578602741621 * 10^{-5}$	13	0

Tabela 2: Wartości aproksymacji pierwiastka r , wartości funkcji f w tej aproksymacji, liczba potrzebnych iteracji i kod błędu dla zadanych parametrów.

5.4 Interpretacja wyników i wnioski

Żeby móc wykorzystać metodę bisekcji, musieliśmy przekształcić nasz problem do takiego, który tą metodą da się rozwiązać. Następnie musieliśmy dobrać odpowiednie parametry początkowe. Wybór złych parametrów może skutkować błędem lub dużą liczbą wykonywanych iteracji. Zatem najpierw musieliśmy zapoznać się wstępnie z wyglądem i zachowaniem funkcji f - bez tego znalezienie pierwiastków mogłoby być karkołomne.

6 Zadanie 6

6.1 Cel zadania

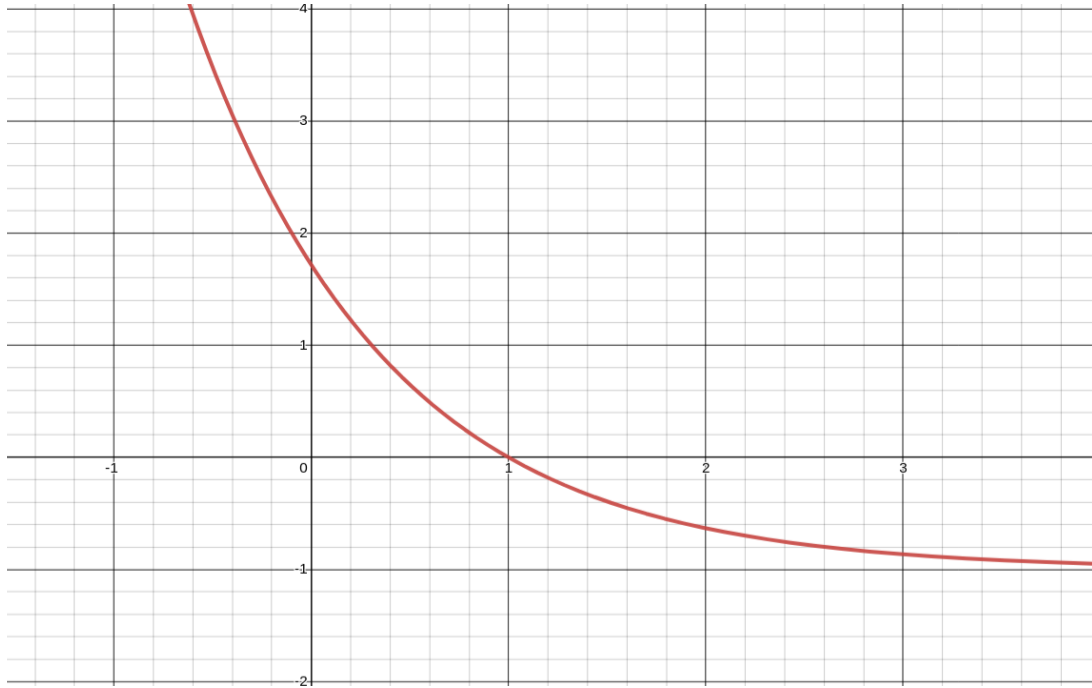
Znaleźć miejsce zerowe dwóch funkcji: $f_1(x) = e^{1-x} - 1$ i $f_2(x) = xe^{-x}$. Wykorzystać w tym celu metodę bisekcji, Newtona i siecznych, dla dokładności $\delta = \epsilon = 10^{-5}$. Następnie przeanalizować co się stanie, gdy w metodzie Newtona dla f_1 wybierzemy $x_0 \in (1, \infty]$ a dla f_2 wybierzemy $x_0 \in [1, \infty]$.

6.2 Rozwiązanie, wyniki i wnioski

Dla każdej z funkcji zastosowane zostały metody zaimplementowane w module `MetodyIteracyjne`. Do każdej z metod dobrano odpowiednie parametry początkowe.

6.2.1 Funkcja f_1

Na początku przyjrzyjmy się wykresowi funkcji f_1 .



Rysunek 5: Wykres funkcji $f_1(x) = e^{1-x} - 1$, dla której szukamy miejsca zerowego.

Z wykresu widzimy, że funkcja ma tylko jedno miejsce zerowe - jest to 1, bo $f_1(1) = 0$. W związku z tym dla metody bisekcji warto wziąć przedział symetryczny względem 1, np. $[0, 2]$ - wtedy powinniśmy otrzymać wynik już w pierwszej iteracji. Dla porównania wywołałem też metodę bisekcji dla przedziałów $[0, 3]$ i dla dużego przedziału: $[-10^6, 10^6]$. Wyniki znajdują się w tabeli poniżej.

$[a, b]$	\tilde{r}	$f_1(\tilde{r})$	liczba iteracji	kod <code>err</code>
$[0, 2]$	1.0	0.0	1	0
$[0, 3]$	1.0000076293945312	$-7.6293654275305656 * 10^{-6}$	17	0
$[-10^6, 10^6]$	1.00000761449337	$-7.614464379912533 * 10^{-6}$	33	0

Tabela 3: Metoda bisekcji dla f_1 : wartości aproksymacji pierwiastka r , wartości funkcji f_1 w tej aproksymacji, liczba potrzebnych iteracji do uzyskania wyniku i kod błędu.

Widzimy, że dla przedziału takiego, że jego środek jest dokładnie szukanym pierwiastkiem, to natychmiast otrzymujemy wynik. Z drugiej strony zwykle nie wiemy, jaka jest dokładna wartość pierwiastka. Wtedy dla metody bisekcji warto spróbować w najmniejszy przedział, na którym funkcja zmienia znak. Widzimy, że dla ogromnego przedziału metoda bisekcji potrzebowała więcej iteracji by odnaleźć właściwą aproksymację miejsca zerowego.

Z wykresu funkcji zauważyć możemy, że z lewej strony funkcja szybko maleje, zaś wraz ze wzrostem x zaczyna maleć coraz słabiej, "wypłaszcza się". Oznacza to, że pochodna $f'_1(x)$ jest bliska zeru dla odpowiednio dużego x . Będzie to skutkować słabym działaniem metody Newtona i metody siecznych dla $x_0 > 1$. W poniższej tabeli przedstawione zostały wyniki działania metody Newtona dla maksymalnej liczby iteracji równej 200 i różnych przybliżeniach początkowych x_0 .

x_0	\tilde{r}	$f_1(\tilde{r})$	liczba iteracji	kod err
-1000.0	NaN	NaN	200	1
-100.0	0.999999998780821	$1.2191803122618694 * 10^{-10}$	105	0
-1.0	0.9999922654776594	$7.734552252003368 * 10^{-6}$	5	0
0.0	0.9999984358892101	$1.5641120130194253 * 10^{-6}$	4	0
0.9	0.99999999931772	$6.822808984452422 * 10^{-11}$	3	0
1.0	1.0	0.0	0	0
1.1	0.9999999991094	$8.906009263398573 * 10^{-11}$	3	0
2.0	0.999999810061002	$1.8993900008368314 * 10^{-8}$	5	0
4.0	0.999999995278234	$4.721767421500545 * 10^{-10}$	21	0
8.0	NaN	NaN	200	1
10000.0	10000.0	-1.0	1	2

Tabela 4: Metoda Newtona dla f_1 : wartości aproksymacji pierwiastka r , wartości funkcji f_1 w tej aproksymacji, liczba potrzebnych iteracji do uzyskania wyniku i kod błędu.

Z powyższej tabeli możemy wyciągnąć następujące wnioski:

- Gdy $x_0 < 1$ i jego wartość nie jest duża co do modułu, to metoda Newtona zachowuje się przyzwoicie i bardzo szybko wyznacza pożądaną aproksymację pierwiastka.
- Gdy $x_0 \ll 1$ to wartość pochodnej $f'_1(x)$ jest bardzo duża i powoduje problemy z numerycznego punktu widzenia. Otrzymujemy wynik NaN.
- Gdy $x_0 = 1$ to nie potrzebujemy wykonywać żadnych iteracji - jesteśmy w pierwiastku.
- Gdy $x_0 > 1$ to pochodna $f'_1(x)$ jest bardzo mała. Skutkuje to słabą zbieżnością dla x_0 bliskich 1 i brakiem zbieżności dla x_0 bardziej oddległych na prawo ($\text{err} = 1$). Oczywiście dla $x_0 \gg 1$ pochodna jest tak mała, że staje się niereprezentowalna - dostajemy błąd $\text{err} = 2$.

Pozostało jeszcze wykorzystać metodę siecznych do znajdowania miejsca zerowego funkcji f_1 . Poniższa tabela obrazuje wyniki działania tej metody, dla różnych wybranych parametrów początkowych.

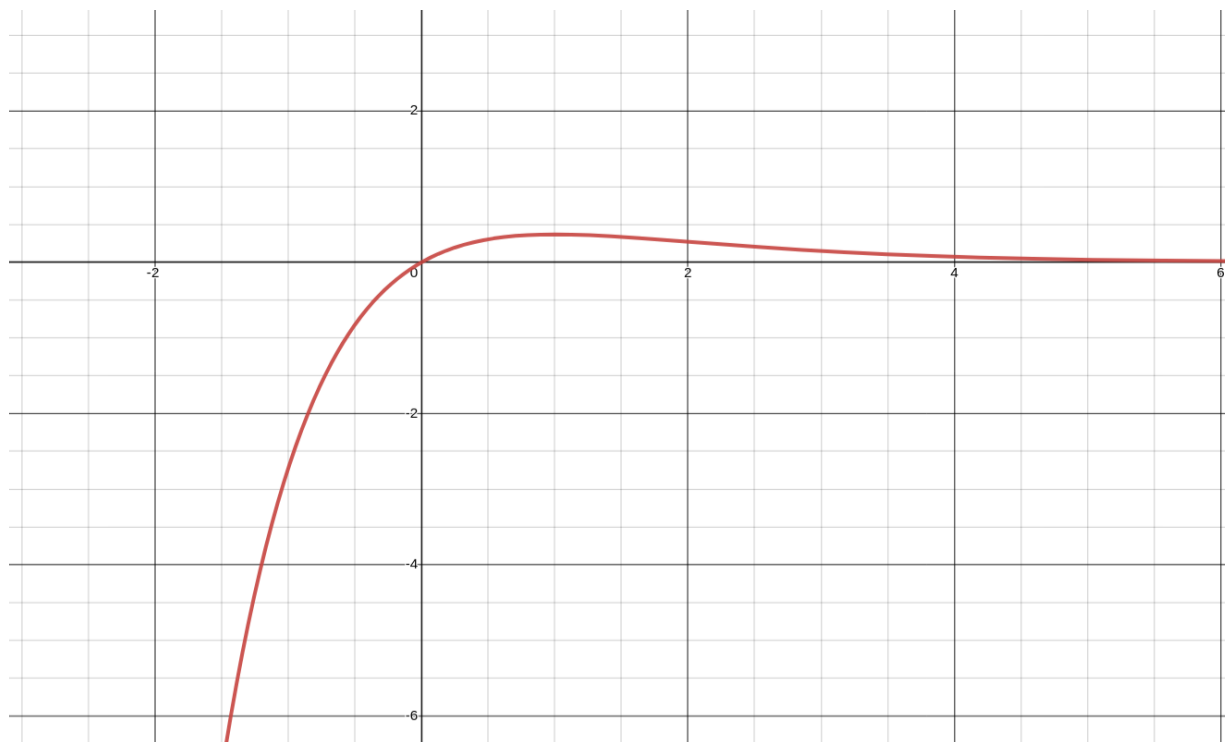
x_0, x_1	\tilde{r}	$f_1(\tilde{r})$	liczba iteracji	kod err
0.0, 0.5	0.9999998133327657	$1.8666725165594755 * 10^{-7}$	5	0
0.0, 2.0	1.0000017597132702	$-1.7597117218937086 * 10^{-6}$	6	0
0.9, 20000	1.000000519980702	$-5.199805668265611 * 10^{-7}$	8	0
10000.0, 10000.5	NaN	NaN	200	1
-100.0, -99.5	0.9999999198705259	$8.012947727564779 * 10^{-8}$	150	0

Tabela 5: Metoda siecznych dla f_1 : wartości aproksymacji pierwiastka r , wartości funkcji f_1 w tej aproksymacji, liczba potrzebnych iteracji do uzyskania wyniku i kod błędu.

Zauważyć możemy, że i metoda siecznych potrzebuje dobrania odpowiednich parametrów początkowych (x_0 i x_1). Wydaje się jednak, że w tym przypadku metoda siecznych jest bardziej odporna na słabe parametry początkowe niż metoda Newtona - metoda zbiega nawet dla wybranych bardzo oddległych od siebie wartości x_0 i x_1 .

6.2.2 Funkcja f_2

Tak jak poprzednio, na początku spójrzmy na wykres funkcji f_2 .



Rysunek 6: Wykres funkcji $f_2(x) = xe^{-x}$, dla której szukamy miejsca zerowego.

Widzimy, że miejsce zerowe jest równe 0, tj. $f_2(0) = 0$. Dla ujemnych x funkcja zaczyna coraz szybciej maleć, zaś prawa strona wykresu zdaje się wypłaszczać, dla $x \gg 0$ $f(x) \approx 0$. Oznacza to również, że dla dużych dodatnich x pochodna funkcji f_2 będzie bliska zeru - informacja ważna dla metody Newtona.

Poniżej znajdują się wyniki dla metody bisekcji z wykorzystaniem różnych przedziałów początkowych.

$[a, b]$	\tilde{r}	$f_1(\tilde{r})$	liczba iteracji	kod err
$[-1, 1]$	0.0	0.0	1	0
$[-1, 2]$	$7.62939453125 * 10^{-6}$	$-7.62933632381113 * 10^{-6}$	17	0
$[-0.1, 30]$	14.950000000000001	$4.807714704475964 * 10^{-6}$	1	0
$[-10^6, 0.01]$	$2.834147453540843 * 10^{-6}$	$2.8341394211604368 * 10^{-6}$	36	0
$[-10^6, 10^8]$	$4.95 * 10^7$	0.0	1	0

Tabela 6: Metoda bisekcji dla f_2 : wartości aproksymacji pierwiastka r , wartości funkcji f_2 w tej aproksymacji, liczba potrzebnych iteracji do uzyskania wyniku i kod błędu.

Widzimy, że metoda bisekcji dla przedziałów o prawym krańcu odległym od pierwiastka daje \tilde{r} bardzo dalekie od rzeczywistego pierwiastka. Działa ona jednak poprawnie - podane przez nią wartości $f(\tilde{r})$ mieszczą się w zadanej dokładności ϵ . Warto zatem zadbać o to, żeby przedział był dostatecznie mały. Dodatkowo zredukuję to liczbę iteracji potrzebną do uzyskania rozwiązania.

Zobaczmy w poniższej tabeli jakie wyniki zwróci metoda Newtona dla wybranych wartości przybliżenia początkowego.

x_0	\tilde{r}	$f_2(\tilde{r})$	liczba iteracji	kod err
-1000.0	NaN	NaN	200	1
-100.0	$4.356806237879908 * 10^{-6}$	$-4.356825219681853 * 10^{-6}$	108	0
-1.0	$-3.0642493416461764 * 10^{-7}$	$-3.0642502806087233 * 10^{-7}$	5	0
-0.1	$-6.707074105306854 * 10^{-9}$	$-6.7070741502916976 * 10^{-9}$	3	0
0.0	0.0	0.0	0	0
0.1	$-1.4906619716777104 * 10^{-8}$	$-1.490661993898442 * 10^{-8}$	3	0
0.9	$-1.302601427744421 * 10^{-7}$	$-1.30260159742148 * 10^{-7}$	15	0
1.0	1.0	0.36787944117144233	1	2
2.0	14.398662765680003	$8.03641534421721 * 10^{-6}$	10	0
4.0	14.398662765680003	$8.03641534421721 * 10^{-6}$	9	0
8.0	14.636807965014	$6.438155219843286 * 10^{-6}$	6	0
1000.0	1000.0	0.0	0	0

Tabela 7: Metoda Newtona dla f_2 : wartości aproksymacji pierwiastka r , wartości funkcji f_2 w tej aproksymacji, liczba potrzebnych iteracji do uzyskania wyniku i kod błędu.

Z powyższej tabeli możemy wyciągnąć następujące wnioski:

- Gdy $x_0 \ll 0$ to wartość pochodnej $f'_2(x)$ jest bardzo duża i powoduje problemy z numerycznego punktu widzenia. Otrzymujemy wynik NaN i $\text{err} = 1$.
- Gdy $x_0 < 0$ i jego wartość nie jest duża co do modułu, to metoda Newtona zachowuje się przyzwoicie i bardzo szybko wyznacza pożądaną aproksymację pierwiastka.
- Gdy $x_0 = 0$ to nie potrzebujemy wykonywać żadnych iteracji - jesteśmy w pierwiastku.
- Gdy $x_0 \in (0, 1)$ to metoda jest zbieżna do pierwiastka (im bliżej zera starujemy, tym szybciej powinniśmy do tego zera dojść).
- Gdy $x_0 = 1$, to zauważmy, że $f'_2(1) = 0$, stąd otrzymujemy błąd $\text{err} = 2$ już przy pierwszej iteracji.
- Gdy $x_0 > 1$, ale jest bliskie 1, to metoda jest rozbieżna. Odbiega w prawo od właściwego rozwiązania do momentu, gdy napotka pierwsze rozwiązanie, dla którego $|f(\tilde{r})| < \epsilon$.
- Gdy $x_0 \gg 1$, to metoda kończy działanie od razu, zwracając x_0 . Wynika to z tego, że wtedy $|f(x_0)| < \epsilon$.

Ostatnim celem tego zadania było wykorzystanie metody siecznych do znalezienia miejsca zerowego funkcji f_2 . Wyniki dla wybranych przybliżeń początkowych x_0, x_1 obrazuje poniższa tabela.

x_0, x_1	\tilde{r}	$f_2(\tilde{r})$	liczba iteracji	kod err
-1.0, 0.5	$-1.2229958402039555 * 10^{-7}$	$-1.2229959897758473 * 10^{-7}$	6	0
1.0, 0.5	$8.76690178927691 * 10^{-8}$	$8.766901020691273 * 10^{-8}$	9	0
0.0, 2.0	0.0	0.0	1	0
-0.9, 20000.0	20000.0	0.0	1	0
-100.0, -99.5	$-4.2135634554565904 * 10^{-6}$	$-4.213581209610988 * 10^{-6}$	155	0
-200.0, -199.5	-61.978488205895864	$-5.118661414969158e28 * 10^{28}$	200	1
-10000.0, -0.5	-0.5	-0.8243606353500641	1	0

Tabela 8: Metoda siecznych dla f_2 : wartości aproksymacji pierwiastka r , wartości funkcji f_2 w tej aproksymacji, liczba potrzebnych iteracji do uzyskania wyniku i kod błędu.

Dla przybliżeń początkowych będących blisko szukanego pierwiastka, metoda siecznych działa prawidłowo i zwraca oczekiwany rezultat. Gdy jednym z przybliżeń początkowych jest sam szukany pierwiastek, to metoda poprawnie zwraca jako wynik właśnie to przybliżenie początkowe (wiersz 3. Tabeli 8.). Widzimy ponownie problem wynikający z tego, że gdy $x \gg 0$, to $f(x) \approx 0$ - wtedy metoda siecznych zwróci wynik odległy od szukanego miejsca zerowego. Dla przybliżeń początkowych odległych od pierwiastka metoda może nie zdążyć zbiec, jak miało to miejsce w przedostatnim wierszu. Interesujący wynik widzimy w ostatnim wierszu

powyższej tabeli. Sieczna przecinająca wykres funkcji f_2 w $x_0 = -10000.0$ i w $x_1 = -0.5$ jest z numerycznego punktu widzenia (w zastosowanej w zadaniu arytmetyce) prostą całkowicie pionową. W związku z tym otrzymamy przybliżenie $x_2 = x_1 = -0.5$. Dostaniemy wtedy $|x_2 - x_1| < \delta$, kończymy działanie po pierwszej iteracji z **błędnym wynikiem** (choć nie otrzymaliśmy informacji o żadnym błędzie!).

6.3 Główny wniosek z zadania

Przy korzystaniu z metod iteracyjnych do wyznaczania miejsc zerowych funkcji należy zachować czujność. Istotne jest odpowiednie dobranie parametrów, żeby wyniki, które otrzymamy, nie były błędne. **Żadna z trzech zaimplementowanych metod nie jest całkowicie niezawodna.** Najbezpieczniejszą ale i zarazem zazwyczaj najwolniejszą okazała się metoda bisekcji. Przy metodzie Newtona i siecznych, ze względu na brak globalnej zbieżności, trzeba dokładnie przeanalizować zachowanie funkcji i jej pochodnej, najlepiej patrząc na wykres (jeśli jest to możliwe). Często może się okazać, że wyznaczenie miejsca zerowego danej funkcji metodami będącymi wyłącznie lokalnie zbieżne jest bardzo trudne i niewielkie zmiany parametrów początkowych mogą prowadzić do zupełnie odmiennego zachowania metody (mogą decydować o jej zbieżności lub rozbieżności).