Sprawozdanie Obliczenia Naukowe

Piotr Zapała

November 2023

Spis treści

1 Zadanie 1 **2**
1.1 Opis problemu . . . 2
1.2 Rozwiązanie . . . 2

2 Zadanie 2 **3**
2.1 Opis problemu . . . 3
2.2 Rozwiązanie . . . 3

3 Zadanie 3 **4**
3.1 Opis problemu . . . 4
3.2 Rozwiązanie . . . 5

4 Zadanie 4 **5**
4.1 Opis problemu . . . 5
4.2 Rozwiązanie . . . 5
4.3 Wyniki . . . 5
4.4 Wnioski . . . 6

5 Zadanie 5 **6**
5.1 Opis problemu . . . 6
5.2 Rozwiązanie . . . 6
5.3 Wyniki . . . 7
5.4 Wnioski . . . 7

6 Zadanie 6 **7**
6.1 Opis problemu . . . 7
6.2 Rozwiązanie . . . 7
6.3 Wyniki . . . 8
6.4 Wnioski . . . 8

1 Zadanie 1

1.1 Opis problemu

W zadaniu pierwszym należy napisać funkcję rozwiązującą równanie $f(x) = 0$ metodą bisekcji.

`function mbisekcji(f::Function, a::Float64, b::Float64, delta::Float64, epsilon::Float64)`

Dla następujących danych wejściowych:
`f` - funkcja $f(x)$ zadana jako anonimowa funkcja,
`a, b` - końce przedziału początkowego,
`delta, epsilon` - dokładność obliczeń,

Nasza funkcja powinna zwracać następujące parametry:
`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 $[a, b]$),

Metoda bisekcji, czyli inaczej metoda połowienia przedziału, jest konsekwencją własności Darboux dla funkcji ciągłych. Jeśli f jest funkcją ciągłą w przedziale $[a, b]$ i jeśli $f(a)f(b) < 0$, a więc f zmienia znak w $[a, b]$, to ta funkcja musi mieć zero w (a, b). Zatem jeśli $f(a)f(b) < 0$, to obliczamy $c = \frac{1}{2}(a + b)$ i sprawdzamy, czy $f(a)f(c) < 0$. Jeśli tak, to f ma zero w $[a, c]$; wtedy pod b podstawiamy c. W przeciwnym razie jest $f(c)f(b) < 0$; wtedy pod a podstawiamy c. W obu przypadkach nowy przedział $[a, b]$, dwa razy krótszy od poprzedniego, zawiera zero funkcji f, zatem postępowanie można powtórzyć. Jeśli $f(a)f(c) = 0$, to $f(c) = 0$ i zero zostało znalezione. Jednakże z powodu błędów zaokrągleń, sytuacja gdzie $f(c) = 0$, jest mało prawdopodobna i nie powinna stanowić kryterium zakończenia obliczeń. Należy więc dopuścić pewną tolerancję wyników, przykładowo $|f(c)| < 10^{-5}$, gdy precyzja arytmetyki wynosi $\epsilon = 2^{-24}$.

1.2 Rozwiązanie

Aby rozwiązać zadanie pierwsze należało zaimplementować w języku `Julia` następujący algorytm.

Algorithm 1 Algorytm Bisekcji

```
function MBISEKCJI(f, a, b, δ, ϵ)
    it ← 0
    u ← f(a)
    v ← f(b)
    e ← (b − a)
    if sgn(u) == sgn(v) then return (0, 0, it, 1)
    end if
    while true do
        it ← it + 1
        e ← e/2
        c ← a + e
        w ← f(c)
        if |e| < δ or |w| < ϵ then return (c, w, it, 0)
        end if
        if sgn(u) ≠ sgn(v) then
            b ← c
            v ← w
        else
            a ← c
            u ← w
        end if
    end while
end function
```

2 Zadanie 2

2.1 Opis problemu

W zadaniu drugim należy napisać funkcję rozwiązującą równanie $f(x) = 0$ metodą stycznych.

```
function mstycznych(f::Function, pf::Function, x0::Float64, delta::Float64,
                epsilon::Float64, maxit::Int)
```

Dla następujących danych wejściowych:
`f, pf` - funkcją $f(x)$ oraz pochodną $f'(x)$ zadane jako anonimowe funkcje,
`x0` - przybliżenie początkowe,
`delta, epsilon` - dokładność obliczeń,
`maxit` - maksymalna dopuszczalna liczba iteracji,

Nasza funkcja powinna zwracać następujące parametry:
`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),

Metoda stycznych, czyli inaczej metoda Newtona, na ogół szybsza od metody bisekcji i siecznych, ponieważ jej zbieżność jest kwadratowa. Gdy tylko przybliżenia tworzone metodą Newtona są dostatecznie bliskie pierwiastka, staje się szybko zbieżna i zaledwie pare kolejnych iteracji daje już maksymalną dokładność. Niech r będzie takim zerem, a x jego przybliżeniem, Jeśli f'' istnieje, to na mocy twierdzenia **Taylora**.

$$0 = f(r) = f(x+h) = f(x) + hf'(x) + O(h^2)$$

gdzie $h = r - x$. Jeśli h jest małe, to możemy pominąć składnik $O(h^2)$ i rozwiązać otrzymane równanie względem h. Otrzymujemy $h = -f(x)/f'(x)$. Zatem jeśli x jest przybliżeniem r, to $x - f(x)/f'(x)$ powinno być lepszym przybliżeniem tego zera. Skąd metoda Newtona polega na rekurencyjnym stosowaniu wzoru.

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

2.2 Rozwiązanie

Aby rozwiązać zadanie drugie należało zaimplementować w języku `Julia` następujący algorytm.

Algorithm 2 Algorytm Newtona

function MSTYCZNYCH($f, pf, x0, \delta, \epsilon, maxit$)
    $v \leftarrow f(x0)$
    if $|v| < \epsilon$ **then return** $(x0, v, 0, 0)$
    end if
    if $|pf(x0)| < \epsilon$ **then return** $(0, 0, 0, 2)$
    end if
    for $i \leftarrow 1, maxit$ **do**
        $x1 \leftarrow x0 - v/pf(x0)$
        $v \leftarrow f(x1)$
        if $|x1 - x0| < \delta$ or $|v| < \epsilon$ **then return** $(x1, v, i, 0)$
        end if
        $x0 \leftarrow x1$
    end for
end function

3 Zadanie 3

3.1 Opis problemu

W zadaniu trzecim należy napisać funkcję rozwiązującą równanie $f(x) = 0$ metodą siecznych.

```
function msiecznych(f::Function, x0::Float64, x1::Float64, delta::Float64,
                    epsilon::Float64, maxit::Int)
```

Dla następujących danych wejściowych:
`f` - funkcja $f(x)$ zadana jako anonimowa funkcja,
`x0, x1` - przybliżenia początkowe,
`delta, epsilon` - dokładność obliczeń,
`maxit` - maksymalna dopuszczalna liczba iteracji,

Nasza funkcja powinna zwracać następujące parametry:
`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),

Jedną z wad metody Newtona jest konieczność obliczania wartości pochodnej funkcji f, której zera szukamy. W celu umienięcia tego problemu **Johan Frederik Steffensen** zaproponował następujący wzór

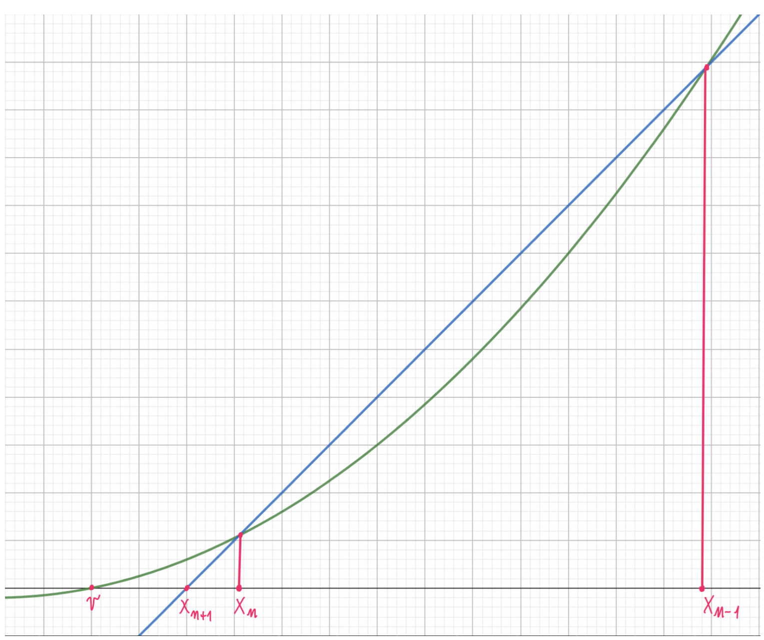
$$x_{n+1} := x_n - \frac{[f(x_n)]^2}{f(x_n + f(x_n)) - f(x_n)}$$

Aby osiągnąć ten sam efekt również można się posłużyć wzorem na iloraz różnicowy.

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

Powyższa przybliżona równość, bezpośrednio daje nam metodę siecznych opisaną poniższym wzorem.

$$x_{n+1} := x_n - f(x_n)\frac{x_n - x_{n+1}}{f(x_n) - f(x_{n-1})} \quad (n \geqslant 1)$$



3.2 Rozwiązanie

Aby rozwiązać zadanie trzecie należało zaimplementować w języku `Julia` następujący algorytm.

Algorithm 3 Algorytm siecznych

```
function MSIECZNYCH(f, x0, x1, δ, ε, maxit)
    fa ← f(x0)
    fb ← f(x1)
    for i ← 1, maxit do
        if |fa| > |fb| then
            x0 ↔ x1
            fa ↔ fb
        end if
        s ← (x1 − x0)/(fb − fa)
        x1 ← x0
        fb ← fa
        x0 ← x0 − (fa ∗ s)
        fa ← f(x0)
        if |fa| < ε or |x1 − x0| < δ then return (x0, fa, i, 0)
        end if
    end for
return (0, 0, 0, 1)
end function
```

4 Zadanie 4

4.1 Opis problemu

W zadaniu czwartym, należało wyznaczyć pierwiastki równania $\sin(x) - (\frac{1}{2}x)^2 = 0$, za pomocą metod zaimplementowanych w poprzednich zadaniach.

1. metoda bisekcji z przedziałem początkowym $[1.5, 2]$ oraz dokładnością obliczeń $\delta = \frac{1}{2}10^{-5}$, $\epsilon = \frac{1}{2}10^{-5}$
2. metoda Newtona z przybliżeniem początkowym $x_0 = 1.5$ oraz dokładnością obliczeń $\delta = \frac{1}{2}10^{-5}$, $\epsilon = \frac{1}{2}10^{-5}$
3. metoda siecznych z przybliżeniami początkowymi $x_0 = 1$, $x_1 = 2$ oraz dokładnością $\delta = \frac{1}{2}10^{-5}$, $\epsilon = \frac{1}{2}10^{-5}$

Powyższe równanie posiada dwa miejsca zero, $x = 0$ oraz $x \approx 1.93375$

4.2 Rozwiązanie

W celu rozwiązania powyższych przykładów, należało zaimplementować program testujący, który wykorzystuje funkcje z modułu napisanego w zadaniach 1-3.

4.3 Wyniki

algorytm	x	f(x)	iteracje
bisekcji	1.9337539672851562	-2.7027680138402843e-7	16
Newtona	1.933753779789742	-2.2423316314856834e-8	4
siecznych	1.933753644474301	1.564525129449379e-7	4

Tabela 1: Zadanie 4

4.4 Wnioski

Jak widzimy z powyższej tabeli, nasze metody z powodzeniem zwracają poprawne wyniki równania $\sin(x) - (\frac{1}{2}x)^2 = 0$. Przy okazji analizowania tego typu doświadczenia, należy zwrócić uwagę na kwestie liczby iteracji. Wyniki wskazują same za siebie, metoda Newtona oraz metoda siecznych są szybciej zbieżne od metody bisekcji. Przy okazji opisu zadania drugiego nadmnieniłem, że metoda Newtona jest szybsza od metody bisekcji, gdyż w przypadku tej pierwszej zbieżność jest kwadratowa, a nie liniowa bądź nadliniowa. Nasze wyniki faktycznie potwierdzają fakt dotyczący zbieżności użytych metod.

5 Zadanie 5

5.1 Opis problemu

W zadaniu piątym jesteśmy proszeni o to, aby użyć metody bisekcji do wyznaczenia punktu przecięcia wykresów następujących funkcji $y = 3x$ oraz $y = e^x$. Dodatkowo należało zachować następujące dokładności obliczeń $\delta = 10^{-4}$ oraz $\epsilon = 10^{-4}$.

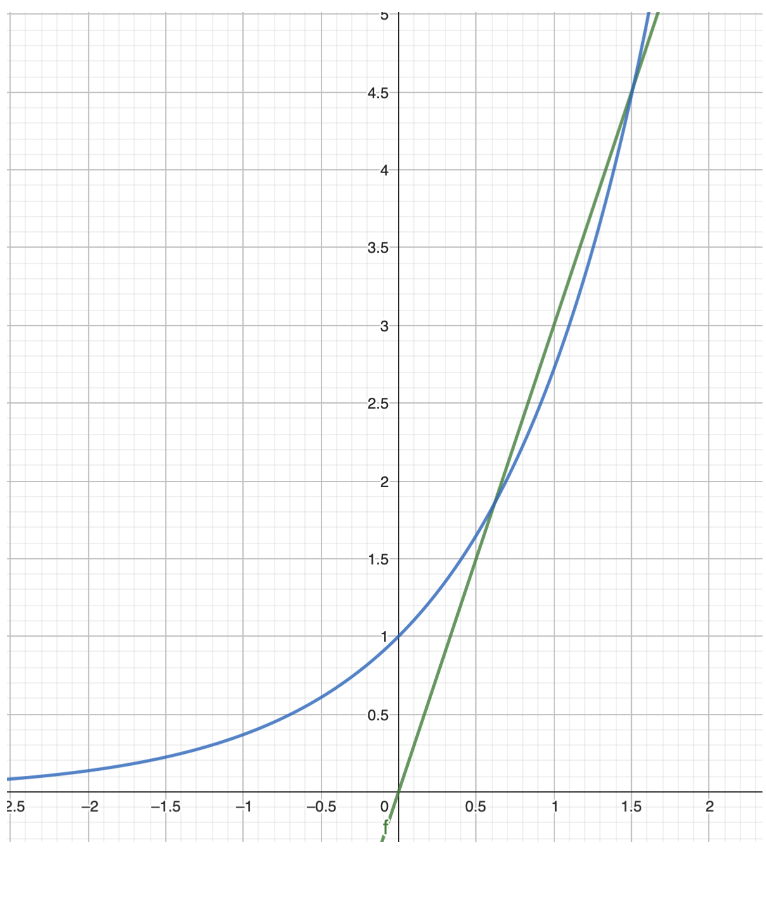
5.2 Rozwiązanie

Aby rozwiązać powyższe zadanie, należało napisać program testujący wykorzystujący funkcję `function mbisekcji(f::Function, a::Float64, b::Float64, delta::Float64, epsilon::Float64)`.

Nastepnie należało wykonać, przekształcenie powyższych równań do następującej postaci:

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

Aby wyznaczyć przedział początkowy posłużyłem się programem `GeoGebra`, z poniższego wykresu wynika, że nasze funkcje przecinają się w dwóch miejscach. Pierwsze rozwiązanie zawiera się w przedziale $[0.0, 1.0]$, natomiast drugie w $[1.0, 2.0]$, przybliżone wartości miejsc zerowych to $x \approx 0.619061$ oraz $x \approx 1.51213$.



5.3 Wyniki

Poniższa tabela zawiera wyniki przeprowadzonych doświadczeń.

przedział	x	f(x)	iteracje
$[0.0, 1.0]$	0.619140625	-9.066320343276146e-5	9
$[1.0, 2.0]$	1.5120849609375	-7.618578602741621e-5	13

Tabela 2: Zadanie 5

5.4 Wnioski

Porównując wartości otrzymane z metody bisekcji oraz wartości obliczone przy pomocy `Wolfram Alpha`, możemy stwierdzić, że metoda bisekcji daje wiarygone i jak najbardziej zadawalające wyniki dla naszego przypadku. Niemniej jednak, należy zwrócić uwagę na dość istotny szczegół, mianowicie jakość wyników końcowych oraz liczba iteracji jest mocno uzależniona od przedziału początkowego.

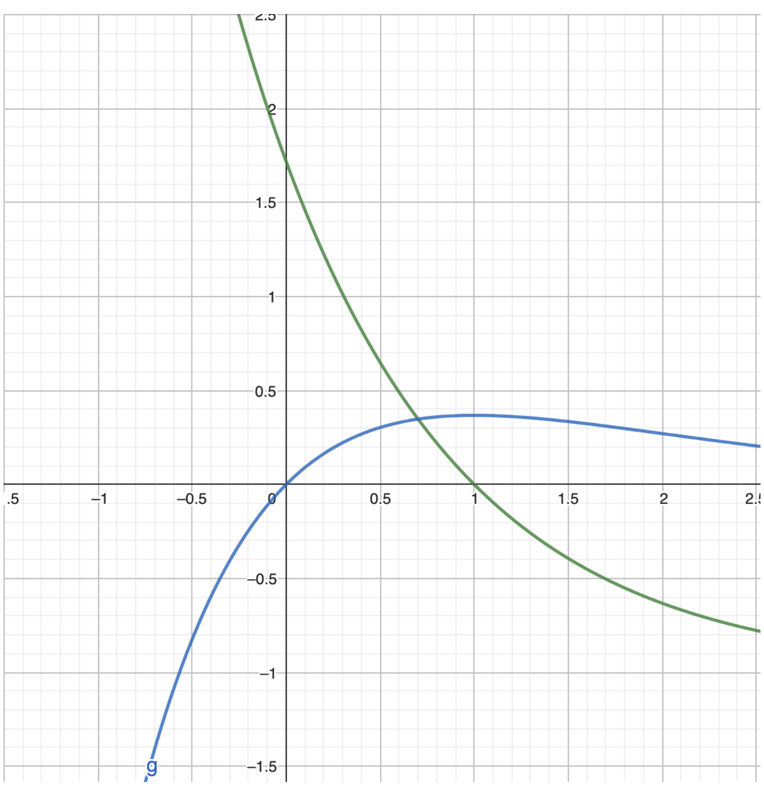
6 Zadanie 6

6.1 Opis problemu

W zadaniu szóstym, należy znaleźć miejsca zerowe funkcji $f_1(x) = e^{1-x} - 1$ oraz $f_2(x) = xe^{-x}$, za pomocą metod bisekcji, Newtona i siecznych. Kolejnym wymaganiem naszego zadania jest, zastosowanie następujących dokładności obliczeń $\delta = 10^{-5}$, $\epsilon = 10^{-5}$ oraz dobranie odpowiednich przedziałów i punktów początkowych. Następnie jesteśmy proszeni o sprawdzenie, co się stanie w metodzie Newtona dla $f_1(x) = e^{1-x} - 1$, gdy wybierzemy $x_0 \in (1, \infty]$, a dla $f_2(x) = xe^{-x}$ wybierzemy $x_0 > 1$.

6.2 Rozwiązanie

Rozwiązanie polega na zaimplementowaniu programu testującego, który dla f_1, f_2, danych przedziałów i punktów początkowych, oraz podanej dokładności obliczeń wyznaczy miejsca zerowe funkcji. Aby dobrać odpowiednie punkty oraz przedziały początkowe, posłużyłem się programem `GeoGebra`.



Na podstawie wykresu z pewnością możemy stwierdzić, że miejsce zerowe funkcji f_1 (wykres zielony) zawiera się w przedziale $[0.0, 2.0]$, natomiast miejsce zerowe f_2 (wykres niebieski) zawiera się w przedziale $[-1.0, 1.0]$. Możemy z dużym prawdopodobieństwem założyć, że nasze pierwiastki znajdują się dokładnie w połowie powyższych przedziałów. Aby lepiej zaobserwować zachowanie naszym algorytmów należy wybrać nieco inny przedział, gdyż przykładowo dla przedziału $[0.0, 2.0]$ w metodzie bisekcji najprawdopodobniej już pierwsza iteracja zwróci nam poprawny wynik.

Wybrane przedziały oraz punkty początkowe przedstawiam w poniższej tabeli.

algorytm	**funkcja**	**parametry**
bisekcji	f_1	$[0.5, 2.0]$
bisekcji	f_2	$[-1.5, 0.5]$
Newtona	f_1	$x_0 = 0$
Newtona	f_2	$x_0 = -1$
siecznych	f_1	$x_0 = 0.5, x_1 = 2.0$
siecznych	f_2	$x_0 = -1.5, x_1 = 0.5$

Tabela 3: przedziały oraz punkty początkowe

6.3 Wyniki

Wyniki z przeprowadzonych doświadczeń przedstawiam w poniższych tabelach.

algorytm	**parametry**	**x**	**f(x)**	**iteracje**
bisekcji	$[0.5, 2.0]$	0.9999923706054688	7.629423635080457e-6	16
Newtona	$x_0 = 0$	0.9999984358892101	1.5641120130194253e-6	4
siecznych	$x_0 = 0.5, x_1 = 2.0$	1.000000014307199	-1.4307198870078253e-8	6

Tabela 4: funkcja $f_1(x) = e^{1-x} - 1$

algorytm	**parametry**	**x**	**f(x)**	**iteracje**
bisekcji	$[-1.5, 0.5]$	0.0	0.0	2
Newtona	$x_0 = -1$	-3.0642493416461764e-7	-3.0642502806087233e-7	5
siecznych	$x_0 = -1.5, x_1 = 0.5$	1.7791419742860986e-8	1.779141942632637e-8	8

Tabela 5: funkcja $f_2(x) = xe^{-x}$

6.4 Wnioski

Z powyższych wyników, możemy potwierdzić wcześniejsze przypuszczenia dotyczące wartości pierwiastków równań f_1 i f_2. W przypadku f_1 jest to $x = 1$, a dla f_2 mamy $x = 0$. Najbardziej uwidaczniającą się obserwacją jest to, że obcięcie przedziału $[0.0, 2.0]$ do $[0.5, 2.0]$ dla metody bisekcji, poskutkowało zwiększeniem liczby iteracji potrzebnych do wyznaczenia zera funkcji f_1. Dla funkcji f_2 w metodzie bisekcji, zastosowaliśmy przedział $[-1.5, 0.5]$, którego środkiem jest 0. Zatem nie powinien dziwić nas fakt, iż metoda bisekcji już po dwóch iteracjach zwraca poprawny wynik. Kolejny raz otrzymane wyniki potwierdzają fakt, iż czas potrzebny na wyznaczenie pierwiastków metodą bisekcji jest uzależniony od wyboru przedziału początkowego. Jeśli chodzi o resztę metod, to można powiedzieć, że wyniki są zgodne z naszymi oczekiwaniami. Pod warunkiem, że środkiem wybranego przez nas przedziału początkowego nie jest pierwiastek danego równania, należy się spodziewać, że metoda Newtona i siecznych dadzą wynik po mniejszej liczbie iteracji niż metoda połowienia przedziału (bisekcji). Zgodnie z poleceniem, należało jeszcze sprawdzić co się stanie w metodzie Newtona, gdy dla f_1 wybierzemy $x_0 \in (1, \infty]$, oraz dla f_2, $x_0 > 1$. W moim przypadku wybrałem $x_0 = 50$, przy próbie wykonania obliczeń dla funkcji f_1 otrzymałem komunikat o błędzie, natomiast dla f_2 wynikiem jest $x = 50$, wtedy wartość funkcji dla tego argumentu jest równa $f_2(50) = 9.643749239819589e - 21$. Generalnie rzecz biorąc wartość otrzymana dla $f_2(50)$ jest bliższa zeru niż wartość f_2, którą wyznaczyliśmy prowadząc obliczenia dla punktu początkowego $x_0 = -1.0$. W tym konkretnym przypadku, metoda Newtona wskazuje liczbę 50 jako pierwiastek równania f_2, jednak nie jest to prawda. Wykres f_2 nie pozostawia złudzeń i jednoznacznie wskazuje na to co odpowiada, za powyższe nieścisłości. Granica $\lim_{x\to\infty} f_2(x) = 0$, zatem dla $x_0 \in (1, \infty]$ funkcja jest zbieżna do zera, a wartości bliskie zera są błędnie interpretowane jako miejsca przecięcia z osią OX. Natomiast dla punktu początkowego $x_0 = 1$, metoda Newtona zwraca komunikat o błędzie. Powodem takiego obrotu spraw jest to, że wartość pochodnej $f_2'(x) = -e^{-x}(x - 1)$ w punkcie $x_0 = 1$ wynosi $f_2'(1) = 0$, co jest jednoznaczne z tym że styczna leży na osi OX.

Literatura

[1] D. Kincaid, W. Cheney, Analiza numeryczna, Wydawnictwa Naukowo-Techniczne, Warszawa 2006. ISBN 83-204-3078-X