

Lista 3

Obliczenia naukowe

Patryk Majewski
250134

Uwaga:

Implementacje zadań 1-3, zgodnie z zaleceniem z listy, umieszczone zostały w jednym pliku `miejsca_zerowe.jl` i opakowane w moduł `MiejscaZerowe`. Testy tych implementacji znajdują się w pliku `testy.jl`.

Zadanie 1 – Metoda bisekcji

Celem zadania była implementacja funkcji rozwiązującej równanie $f(x) = 0$ metodą bisekcji. Metoda pozwala na dokonanie tego dla ustalonej funkcji f ciągłej na przedziale $[a, b]$ spełniającym założenie, że na krańcach przedziału wartości funkcji mają przeciwne znaki. Na mocy twierdzenia Darboux o przyjmowaniu wartości pośrednich istnieje wówczas punkt $r \in [a, b]$, że $f(r) = 0$.

Metoda bisekcji jest iteracyjna, a idea kolejnych iteracji jest następująca: wybieramy punkt c leżący pośrodku przedziału. Jeżeli wartość funkcji w tym punkcie wynosi zero, znaleźliśmy rozwiązanie – $r = c$. W przeciwnym wypadku funkcja musi zmieniać znak na jednym z przedziałów: $[a, c]$ lub $[c, b]$. Wybieramy zatem przedział, na którym zachodzi ten warunek i na nim powtarzamy całą procedurę.

Powyższy opis stanowi oczywiście uproszczenie. W prawdziwej implementacji musimy wziąć pod uwagę precyzję, w której decydujemy się operować – szansa, że znajdziemy c , w którym funkcja przyjmuje dokładnie 0 jest bowiem bardzo mała. Warunkiem zakończenia pracy algorytmu nie będzie zatem $f(c) = 0$, a $|f(c)| < \varepsilon$. Dodany zostanie też warunek na dopuszczalną odległość od prawidłowego rozwiązania $|e| < \delta$, gdzie $e = \frac{b-a}{2}$.

Uproszczony pseudokod metody został przedstawiony po prawej. Oprócz przebiegu opisanego w poprzednich akapitach, z numerycznego punktu widzenia warto jeszcze zwrócić uwagę na dwie kwestie. Zmiany znaku sprawdzamy za pomocą funkcji `signum`, a nie jak w przypadku "ręcznego" liczenia sprowadzamy do warunku $f(a) \cdot f(b) < 0$. Pomnożenie dwóch bardzo małych liczb różnych znaków mogłoby dać wynik, który dla danej precyzji znajduje się w przedziale zera maszynowego, przez co nasz warunek nie byłby spełniony (choć powinien). Środek przedziału c wyznaczamy przez dodanie do jego początku połowy jego długości, dzięki czemu mamy pewność, że $a < c$. Ograniczona precyzja mogłaby bowiem spowodować, że $\frac{a+b}{2} \notin [a, b]$. Działanie metody zostało dodatkowo zilustrowane na rysunku 1.

```
def mbisekcji(f, a, b, δ, ε) :  
    if  $\text{sgn}(f(a)) = \text{sgn}(f(b))$  then  
        return błąd: funkcja nie zmienia znaku na  $[a, b]$   
    end if  
     $e \leftarrow b - a$   
    loop  
         $e \leftarrow e/2$   
         $c \leftarrow a + e$   
        if  $|f(c)| < \varepsilon$  or  $|e| < \delta$  then  
            return c, f(c)  
        else if  $\text{sgn}(f(a)) \neq \text{sgn}(f(c))$  then  
             $b \leftarrow c$   
        else  
             $a \leftarrow c$   
        end if  
    end loop
```

Zadanie 2 – Metoda Newtona

Tym razem za zadanie mieliśmy implementację funkcji rozwiązującej równanie $f(x) = 0$ metodą stycznych, zwaną też metodą Newtona. Funkcja f musi być dwukrotnie różniczkowalna, z twierdzenia Taylora mamy wówczas

$$0 = f(r) = f(x_n + h) \approx f(x_n) + h \cdot f'(x_n)$$

gdzie r jest poszukiwanym pierwiastkiem, x_n jego obecną aproksymacją, a $h = r - x_n$. Po prostym przekształceniu otrzymujemy

$$h \approx \hat{h} = -\frac{f(x_n)}{f'(x_n)}$$

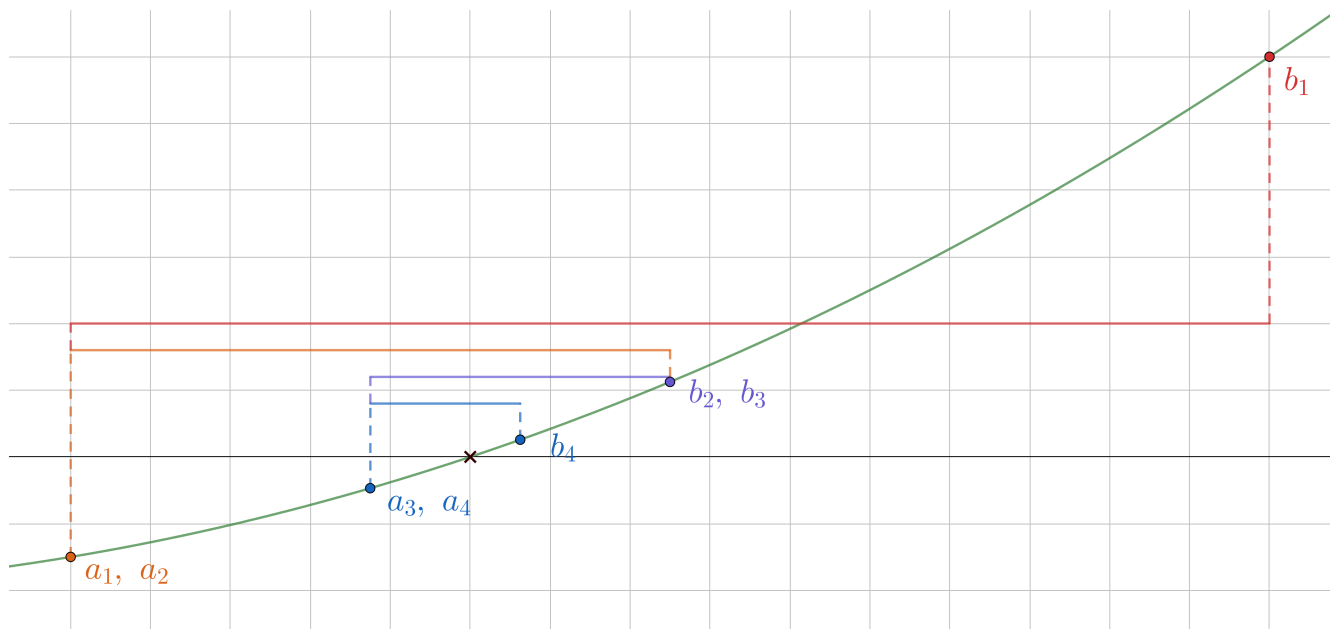
Tworzymy zatem nową aproksymację, x_{n+1} , w następujący sposób:

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

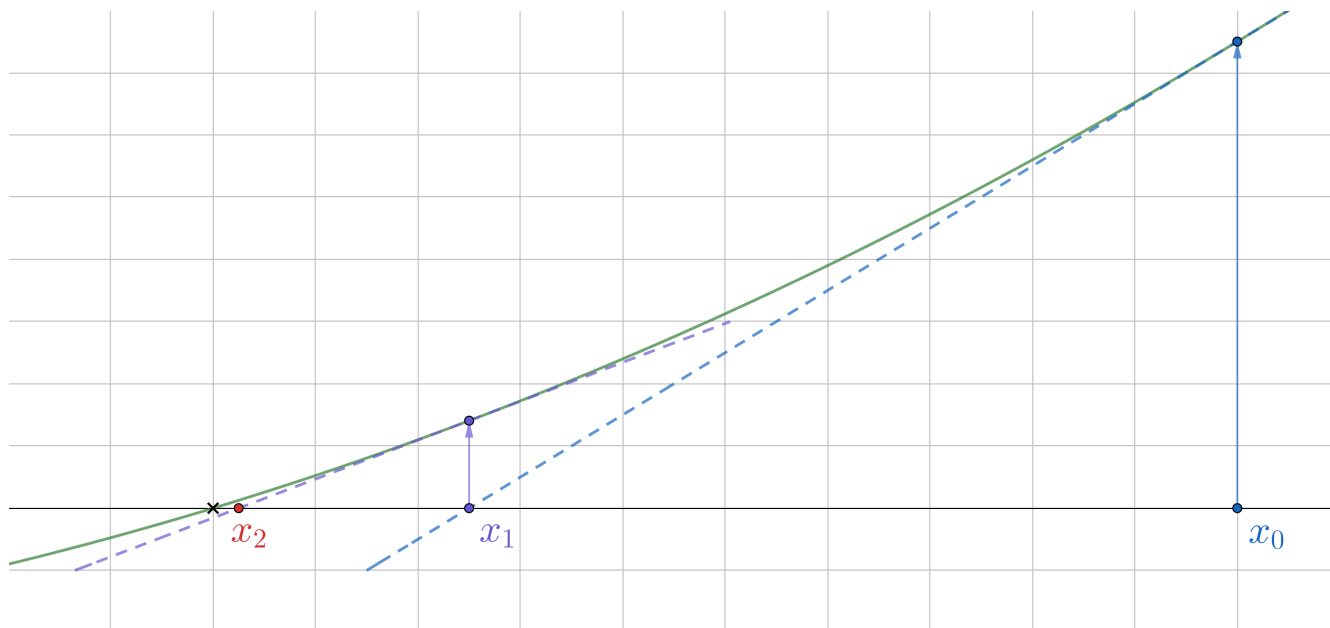
Kolejne przybliżenia tworzymy, podobnie jak poprzednio w metodzie bisekcji, do momentu, aż dla obecnego n zachodzi $|f(x_n)| < \varepsilon$ lub $|x_n - x_{n-1}| < \delta$. Metoda wymaga od nas wybrania x_0 z odpowiednio małego otoczenia r , co oznacza, że dla zbyt odległych punktów początkowych kolejne aproksymacje mogą rozbiegać się coraz dalej od prawdziwego pierwiastka. Warto zatem dodać również wentyl bezpieczeństwa w postaci maksymalnej liczby iteracji. Problem pojawia się również w momencie, gdy dla którejś z wyznaczonych aproksymacji $f'(x_n)$ ma wartość bliską zeru – oznacza to, że kolejna aproksymacja wystrzeliwuje bardzo daleko od obecnej, zasadniczo opuszczając otoczenie pierwiastka, być może uniemożliwiając dotarcie do niego. Będzie to ostatni z naszych warunków końcowych.

Nieodłącznym elementem opisu idei metody Newtona jest jej interpretacja graficzna, której zawdzięcza ona swoją drugą nazwę. Zasadniczo bowiem w celu wybrania x_{n+1} prowadzimy styczną do wykresu funkcji w punkcie $(x_n, f(x_n))$, a za x_{n+1} przyjmujemy punkt jej przecięcia z osią OX. Takie spojrzenie pozwala w łatwy sposób zauważyć mankamenty algorytmu – na przykład pochodna bliska zeru w x_n oznacza niemal poziomą styczną do wykresu, zatem jej przecięcie z osią OX znajdzie się bardzo daleko od x_n . Kilka pierwszych iteracji metody Newtona przedstawionych zostało na rysunku 2. Pseudokod metody został przedstawiony po prawej.

```
def mstycznych(f, f', δ, ε, maxit) :  
    if |f(x0)| < ε then  
        return x0, f(x0)  
    end if  
    for it from 1 to maxit do  
        if |f'(x0)| < ε then  
            return x0, f(x0) z błędem: pochodna bliska zeru  
        end if  
        x1 ← x0 -  $\frac{f(x_0)}{f'(x_0)}$   
        if |x1 - x0| < δ or |f(x1)| < ε then  
            return x1, f(x1)  
        end if  
        x0 ← x1  
    end for  
    return x0, f(x0) z ostrzeżeniem: nie osiągnięto wymaganej  
    precyzji w maxit iteracjach
```



Rysunek 1: Graficzna ilustracja kilku pierwszych iteracji metody bisekcji. Indeksy przy zaznaczonych punktach oznaczają numer iteracji, w której dany punkt (a dokładniej jego rzut na oś OX) przyjęto za początek lub koniec przedziału poszukiwań, krzyżykiem zaś zaznaczono poszukiwany punkt r .



Rysunek 2: Graficzna interpretacja metody Newtona. Rozpoczynamy w punkcie x_0 , gdzie tworzymy styczną do wykresu. Punkt x_1 wyznaczamy jako przecięcie tej stycznej z osią OX. Analogicznie wyznaczony został punkt x_2 . Poszukiwany punkt r oznaczono krzyżykiem.

Zadanie 3 – Metoda siecznych

Ostatnim zaimplementowanym sposobem rozwiązania równania $f(x) = 0$ jest metoda siecznych. Jedną z trudności w wykorzystaniu metody Newtona jest konieczność znania wzoru pochodnej badanej funkcji. Metoda siecznych jest jednym ze sposobów jej obejścia. Z definicji pochodnej funkcji f w punkcie x_n mamy bowiem

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

Zastosujemy zatem przybliżenie

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

Podstawiając do wzoru na kolejną aproksymację z metody Newtona otrzymujemy

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

Zauważmy, że do wyznaczenia kolejnej aproksymacji, oprócz bieżącej, potrzebujemy również poprzedniej. Na starcie metoda potrzebuje więc dwóch punktów początkowych należących do odpowiednio małego otoczenia pierwiastka r . Ponieważ metoda siecznych oparta jest na metodzie stycznych, również dla niej może się okazać, że początkowe przybliżenia nieleżące dostatecznie blisko właściwej wartości r spowodują rozbieganie się kolejnych aproksymacji.

Podobnie jak poprzednio, metoda siecznych zawdzięcza nazwę geometrycznej interpretacji swojego przebiegu. Aby wybrać x_{n+1} , najpierw konstruujemy sieczną przebiegającą przez punkty $(x_{n-1}, f(x_{n-1}))$ oraz $(x_n, f(x_n))$. Za kolejną aproksymację pierwiastka przyjmujemy wtedy punkt przecięcia narysowanej siecznej z osią OX. Działanie metody zostało dodatkowo zilustrowane na rysunku 3. Uproszczony pseudokod metody został podany po prawej. Dzięki warunkowi $|f(x_0)| \geq |f(x_1)|$ mamy zapewnione, że wartość funkcji nie będzie rosła w kolejnych iteracjach.

```
def msiecznych(f, x0, x1, delta, epsilon, maxit):
```

```
    for it from 1 to maxit do
```

```
        if |f(x0)| < |f(x1)| then
```

```
            swap: x0 ↔ x1
```

```
        end if
```

```
        df ← (x1 - x0) / (f(x1) - f(x0))
```

```
        x0 ← x1
```

```
        x1 ← x1 - f(x1) · df
```

```
        if |x1 - x0| < delta or |f(x1)| < epsilon then
```

```
            return x1, f(x1)
```

```
        end if
```

```
    end for
```

```
    return x1, f(x1) z ostrzeżeniem: nie osiągnięto wymaganej precyzji w maxit iteracjach
```

Zadanie 4

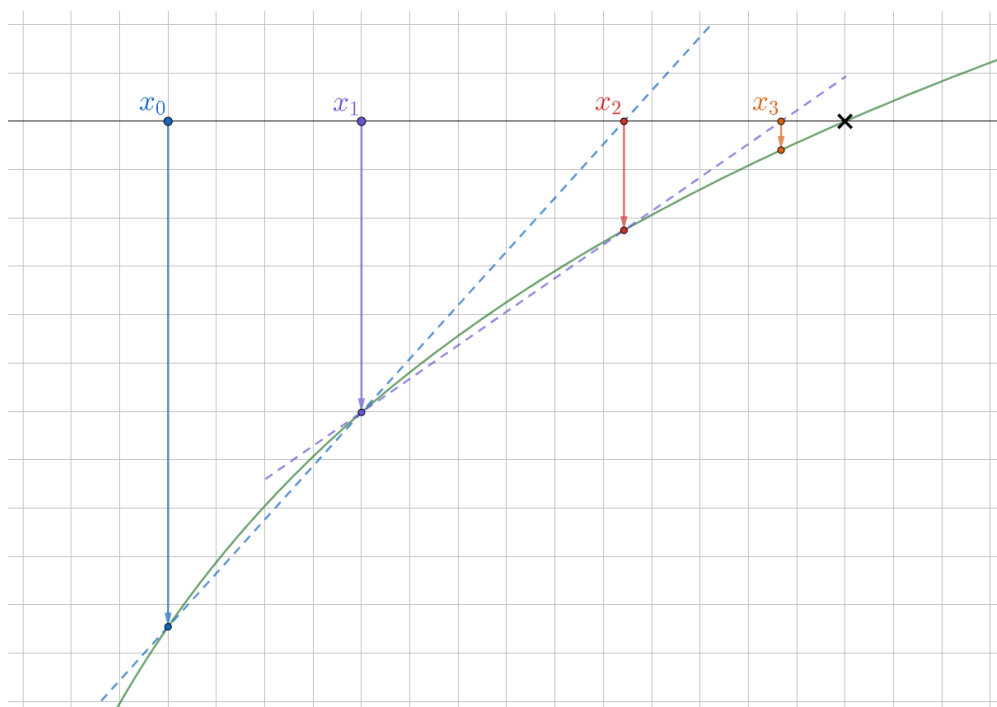
Celem zadania było wykorzystanie zaimplementowanych metod do wyznaczenia rozwiązania równania

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

z ustaloną dokładnością $\varepsilon = \delta = \frac{1}{2} \cdot 10^{-5}$. Wyniki przedstawione zostały w tabeli 1.

metoda	dane początkowe	pierwiastek r	$f(r)$	liczba iteracji	kod błędu
bisekcji	$a = 1.5, b = 2$	1.9337539672851562	-2.7027680138402843e-7	16	0
stycznych	$x_0 = 1.5$	1.933753779789742	-2.2423316314856834e-8	4	0
siecznych	$x_0 = 1, x_1 = 2$	1.933753644474301	1.564525129449379e-7	4	0

Tabela 1: Wyniki działania zaimplementowanych metod dla funkcji $f(x) = \sin x - (\frac{1}{2}x)^2$. Pogrubione cyfry w wyznaczonych pierwiastkach są zgodne z dokładną wartością miejsca zerowego $\approx 1.9337537628270212$ (za WolframAlpha).



Rysunek 3: Graficzna interpretacja metody siecznych. Rozpoczynamy w punktach x_0 i x_1 , między którymi kreślimy sieczną wykresu. Punkt x_2 wyznaczamy jako jej przecięcie z osią OX. Analogicznie wyznaczony został punkt x_3 . Poszukiwany punkt r oznaczono krzyżykiem.

Wnioski

Startując z podobnych przybliżeń, każdej z metod udało się sprawnie (lepiej niż minimum narzucone przez δ i ε) przybliżyć pierwiastek równania. Liczba iteracji dobrze obrazuje współczynniki zbieżności poszczególnych algorytmów – metody Newtona (zbiegająca kwadratowo) i siecznych (ze współczynnikiem około 1.618) osiągały wynik czterokrotnie szybciej niż liniowo zbieżna metoda bisekcji. Jak się okazuje, metoda stycznych zwróciła również wynik najbardziej dokładny, zarówno pod względem bliskości pierwiastka, jak i wartości funkcji w nim. Można zatem wysnuć wniosek, że w przypadku, gdy mamy do czynienia z porządną funkcją (to znaczy znamy wzór jej pochodnej, łatwo ją sobie wyobrazić i wybrać x_0 , nie ma złośliwych ekstremów lokalnych), warto skorzystać z metody Newtona. Metoda siecznych, radząc sobie z niektórymi z tych trudności, jest w stanie zaoferować nam dobre rozwiązanie w podobnej liczbie iteracji. Jeśli jednak przedkładamy niezawodność nad szybkość działania, powinniśmy zastosować metodę bisekcji.

Zadanie 5

Naszym zadaniem jest znalezienie takich punktów x , że $e^x = 3x$. Oznaczmy $g(x) = e^x - 3x$, wówczas problem sprowadza się do znalezienia miejsc zerowych g . Spróbujmy najpierw zawęzić obszar przeszukiwań. Zauważmy, że dla $x \leq 0$ mamy $3x \leq 0 < e^x$, czyli $g > 0$, zatem w tym przedziale nie znajduje się żaden z poszukiwanych punktów. Ponadto dla $x = 1$ mamy $e \approx 2.718 < 3 = 3 \cdot 1$, natomiast dla $x = 4$ mamy $e^3 > 2^4 = 16 > 12 = 3 \cdot 4$. Z twierdzenia Darboux mamy zatem przynajmniej po jednym pierwiastku dla g na $[0, 1]$ oraz $[1, 4]$. Co więcej, $g'(x) = e^x - 3$ ma tylko jedno miejsce zerowe, z twierdzenia Rolle'a wiemy zatem, że g ma dokładnie dwa pierwiastki. Uruchamiając dla g metodę bisekcji na przedziałach $[0, 1]$ i $[1, 4]$ znajdziemy zatem wszystkie x , że $e^x = 3x$.

Wyniki działania algorytmu z $\delta = \varepsilon = 10^{-4}$ przedstawiono w tabeli 2.

przedział	pierwiastek r	$e^r - 3r$	liczba iteracji	kod błędu
$[0, 1]$	0.619 140625	-9.066320343276146e-5	9	0
$[1, 4]$	1.512 14599609375	1.7583570236290313e-5	14	0

Tabela 2: Wyniki działania metody bisekcji dla funkcji $g(x) = e^x - 3x$. Pogrubione cyfry w wyznaczonych pierwiastkach są zgodne z dokładną wartością miejsc zerowych (za WolframAlpha).

Wnioski

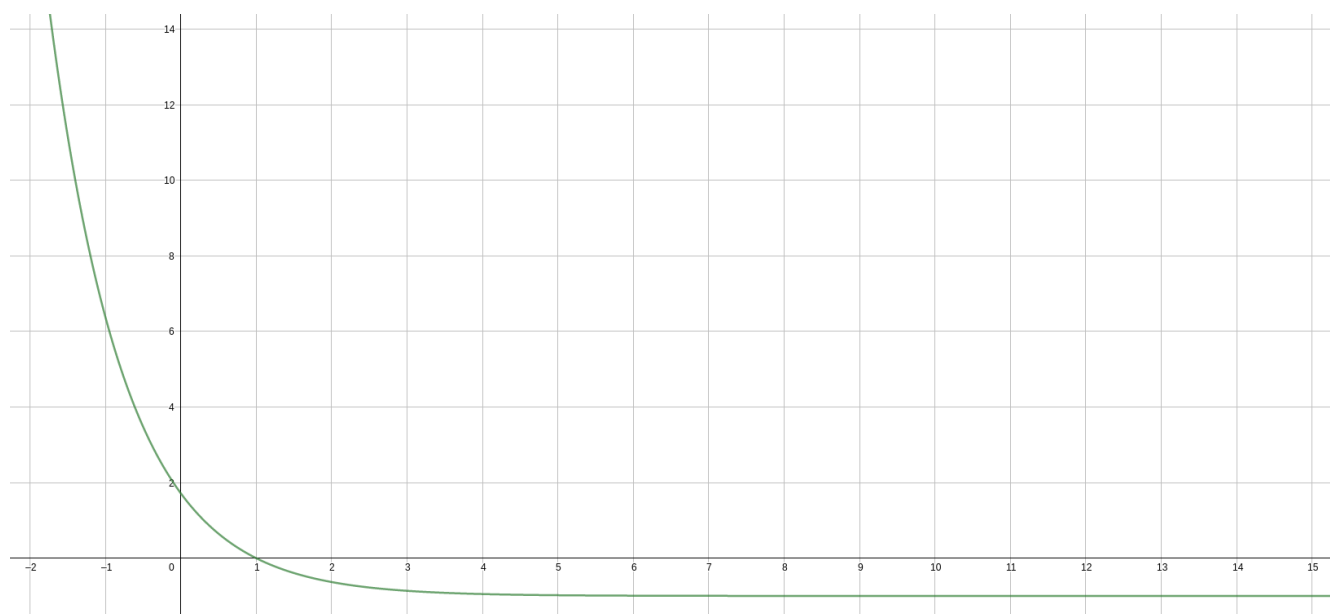
Podstawowa analiza problemu jest kluczowa, aby poprawnie dobrać początkowe parametry dla wybranej metody szukania pierwiastków. Metoda bisekcji, pod warunkiem spełnienia wymagań początkowych, zawsze znajdzie nam jakiś pierwiastek. Chcąc jednak mieć pewność, że uda się znaleźć wszystkie rozwiązania, musimy posłkować się pewnymi faktami z analizy matematycznej. Wstępne rozważenie problemu pozwoli nam też na zawężenie przedziałów poszukiwań, zasadniczo zmniejszając liczbę iteracji algorytmu, która może być duża w porównaniu do metod stycznych i siecznych.

Zadanie 6

Celem zadania było znalezienie miejsc zerowych funkcji $f_1(x) = e^{1-x} - 1$ oraz $f_2(x) = xe^{-x}$. Analiza tych funkcji jest znacznie prostsza niż w zadaniu poprzednim. Łatwo zauważyć, że $f_1(1) = 0$. Co więcej $f_1'(x) = -e^{1-x} < 0$, stąd f_1 malejąca, czyli to jedyny pierwiastek. Podobnie dla f_2 mamy $f_2 = 0 \iff x = 0$, bo $e^{-x} > 0$ dla dowolnego x .

Przyjmujemy $\delta = \varepsilon = 10^{-5}$. Wypróbujemy różne parametry startowe dla zaimplementowanych metod i sprawdzimy, jak dobrze sobie poradzą.

Funkcja f_1



Rysunek 4: Wykres funkcji f_1 . Wraz ze wzrostem x wykres robi się bardzo płaski, co zbliża pochodną do 0 i powoduje problemy w metodach stycznych i siecznych.

przedział	pierwiastek r	$f_1(r)$	liczba iteracji	kod błędu
[0.0, 2.0]	1.0	0.0	1	0
[-0.1, 2.0]	1.0000038146972656	-3.814689989667386e-6	17	0
[-21.37, 37.21]	1.0000023031234724	-2.3031208201684805e-6	22	0
[0.1, 1024.0]	0.9999975502490998	2.449753900846119e-6	24	0
[0.9, 1.0e6]	1.0000006714486518	-6.714484264147558e-7	33	0

Tabela 3: Wyniki działania metody bisekcji dla funkcji f_1 . Widać, że jeżeli przedział jest symetryczny względem pierwiastka, natychmiast otrzymujemy wynik. Większe przedziały wydłużają czas pracy algorytmu, ale metoda dalej zbiega do prawidłowego rozwiązania.

x_0	pierwiastek r	$f_1(r)$	liczba iteracji	kod błędu
0.0	0.9999984358892101	1.5641120130194253e-6	4	0
-4.0	0.999999998856633	1.1433654023562667e-10	9	0
10.0	NaN	NaN	100	1
1.0e6	1.0e6	-1.0	1	2

Tabela 4: Wyniki działania metody stycznych dla funkcji f_1 . Dla zerowego i ujemnego x_0 metoda dość szybko zbiega do dobrego rozwiązania. Stosunkowo niewielkie $x_0 = 10$ powoduje już jednak całkowitą rozbieżność algorytmu z powodu bardzo małej wartości pochodnej, a dla dużego x_0 , zgodnie z intuicją z wykresu, pochodna jest już mniejsza niż ε .

x_0, x_1	pierwiastek r	$f_1(r)$	liczba iteracji	kod błędu
0.0, 2.0	1.0000017597132702	-1.7597117218937086e-6	6	0
-0.1, 2.0	1.0000032272298756	-3.227224668056472e-6	6	0
-21.37, 37.21	37.209999988712916	-0.9999999999999998	1	0
0.1, 1024.0	NaN	NaN	100	1
0.9, 1.0e6	1.0000000260374222	-2.603742188522773e-8	10	0

Tabela 5: Wyniki działania metody siecznych dla funkcji f_1 . Zdaje się ona być bardziej odporna na niekorzystne położenia startowe od metody Newtona – zbiega nawet dla ogromnego przedziału w ostatnim wierszu tabeli. W przedziale z trzeciego wiersza różnica między kolejnymi krokami jest mniejsza niż δ , więc algorytm kończy pracę, natomiast w przedziale z czwartego wiersza jest rozbieżny.

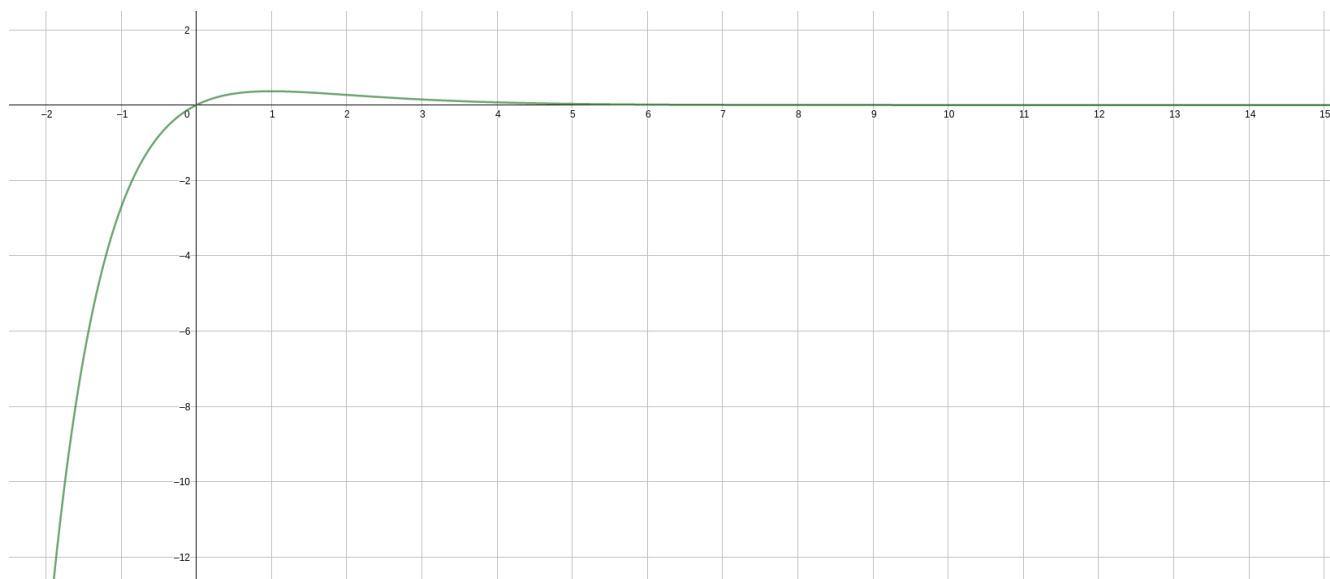
Funkcja f_2

przedział	pierwiastek r	$f_2(r)$	liczba iteracji	kod błędu
[-1.0, 2.0]	7.62939453125e-6	7.62933632381113e-6	17	0
[-0.1, 2.0]	7.629394531248679e-6	7.629336323809809e-6	16	0
[-21.37, 37.21]	-3.418922425933261e-6	-3.4189341149837973e-6	22	0
[-2.1, 1024.0]	510.9499999999993	6.3916483747854906e-220	1	0
[0.9, 1.0e6]	500000.45	0.0	1	0

Tabela 6: Wyniki działania metody bisekcji dla funkcji f_2 . Dla dużych x mamy $|f(x)| < \varepsilon$, więc metoda traktuje je jak miejsca zerowe.

x_0	pierwiastek r	$f_2(r)$	liczba iteracji	kod błędu
-1.0	-3.0642493416461764e-7	-3.0642502806087233e-7	5	0
-4.0	-1.5586599258811135e-6	-1.5586623553037713e-6	9	0
1.0	1.0	0.36787944117144233	1	2
10.0	14.380524159896261	8.173205649825554e-6	4	0
1.0e6	1.0e6	0.0	0	0

Tabela 7: Wyniki działania metody stycznych dla funkcji f_2 . Zgodnie z przewidywaniami po analizie wykresu, w 1 zerowa pochodna przerywa działanie algorytmu, a dla większych x zbiega do nieprawdziwych miejsc zerowych.



Rysunek 5: Wykres funkcji f_2 . W $x = 1$ funkcja ma lokalne maksimum, czyli natrafienie na ten punkt eliminuje metodę stycznych. Co więcej, wartości funkcji są bardzo bliskie zeru dla dużych x , co powodować może niesłuszne traktowanie całych odcinków jako miejsc zerowych funkcji przez każdą z metod.

x_0, x_1	pierwiastek r	$f_2(r)$	liczba iteracji	kod błędu
-0.1, 2.0	2.3230523266763964e-6	2.323046930110552e-6	5	0
-0.7, 1.5	14.850014182115977	5.277735885143915e-6	14	0
-21.37, 37.21	37.21	2.573724154298971e-15	1	0
0.1, 1024.0	1024.0	0.0	1	0
0.9, 1.0e6	1.0e6	0.0	1	0

Tabela 8: Wyniki działania metody siecznych dla funkcji f_2 . Nawet dla tak niewielkich początkowych położań jak -0.7 i 1.5 metoda zbiega do nieprawdziwych miejsc zerowych.

Dodatkowe eksperymenty

- **Co się stanie, gdy w metodzie Newtona dla f_1 przyjmiemy $x_0 > 1$?**

Intuicja nabyta po spojrzeniu na wykres funkcji podpowiada nam, że styczna narysowana w takim punkcie będzie prawie pozioma, co powoduje, że x_1 wystrzeli daleko w tył w stosunku do x_0 . Dla ujemnych x wykres funkcji jest z kolei dosyć stromy, więc styczna będzie prawie pionowa, co powodować będzie zbliżanie się do pierwiastka małymi krokami. Może się też okazać, że uzyskany ujemny x_1 będzie miał ogromną wartość funkcji (bo rośnie eksponencjalnie ze spadkiem x) – wówczas metoda będzie rozbieżna. W przypadku wyjątkowo dużego x_0 okazać się może, że $|f'_1(x_0)| < \varepsilon$, co spowoduje zwrócenie przez metodę błędu. Niektóre z wyników z tabeli 4 zdają się potwierdzać tę intuicję. Wyniki dodatkowego eksperymentu, umieszczone w tabeli 10, również są z nią zgodne.

- **A $x_0 \geq 1$ dla f_2 ?**

Jak spostrzeżliśmy wcześniej, $f'_2(1) = 0$, zatem przyjęcie $x_0 = 1$ spowoduje natychmiastowe zakończenie działania metody błędem. Co więcej, na wykresie widać, że niedaleko za 1 następuje przegięcie i f_2 zaczyna być wypukła, przez co narysowane styczne będą oddalać nas od prawidłowego rozwiązania, powodując rozbieżność metody. Ponadto bardzo duże x_0 może mieć $|f_2(x_0)| < \varepsilon$, co spowoduje przyjęcie go za miejsce zerowe, lub też $|f'_2(x_0)| < \varepsilon$ co zakończy działanie metody błędem.

x_0	pierwiastek r	$f_1(r)$	liczba iteracji	kod błędu
1.1	0.9999999991094	8.905987058938081e-11	3	0
2.0	0.9999999810061002	1.8993900008368314e-8	5	0
5.0	0.9999996427095682	3.572904956339329e-7	54	0
10.0	NaN	NaN	100	1
10000.0	10000.0	-1.0	1	2

Tabela 9: Wyniki działania metody stycznych dla funkcji f_1 i $x_0 > 1$. Startując w punkcie 5 potrzeba już dosyć dużo iteracji, w 10 metoda jest rozbieżna, a w 10^4 pochodna zbyt bliska zeru.

x_0	pierwiastek r	$f_2(r)$	liczba iteracji	kod błędu
1.0	1.0	0.36787944117144233	1	2
1.1	14.272123938290509	9.040322779745447e-6	3	0
2.0	14.398662765680003	8.036415344217211e-6	10	0
5.0	14.118053159563352	1.043235195613951e-5	9	2
10.0	14.380524159896261	8.173205649825554e-6	4	0
10000.0	10000.0	0.0	0	0

Tabela 10: Wyniki działania metody stycznych dla funkcji f_2 i $x_0 \geq 1$. Widać, jak dla wartości większych od 1 stopniowo odbiegamy coraz dalej od prawidłowego rozwiązania tak długo, aż spełni się warunek $|f(x)| < \varepsilon$. Dla 5 natrafiamy nawet po drodze na punkt o prawie zerowej pochodnej.

Wnioski

Używając metod znajdujących pierwiastki należy zachować czujność i przeprowadzić wstępną analizę badanych funkcji, aby być w stanie odsiać przekłamane wyniki. Dla pozornie porządných funkcji okazuje się, że dostosowanie parametrów metody jest kluczową i subtelną kwestią. Jeśli trafimy w niebezpieczne obszary, niewielka zmiana warunków początkowych w najlepszym wypadku wydłuży działanie algorytmu, a w najgorszym uniemożliwi dojście do rozwiązania.

Obserwacje z wyników uzyskanych w zadaniu istotnie potwierdzają największą niezawodność metody bisekcji – w teorii poradzi sobie z dowolnymi prawidłowymi a i b , ale nieuwaga użytkownika może wprowadzić go w błąd nawet przy jej użyciu, jak w przypadku dużych przedziałów dla funkcji f_2 (tabela 6). Chcąc szybko otrzymać orientacyjne wartości pierwiastków, nieostrożny użytkownik może wybrać niewielką dokładność i dosyć duży przedział w przekonaniu o niezawodności metody, co dla złośliwej funkcji może poskutkować dużym błędem wyniku.

Warto też zwrócić uwagę, że samo dostosowanie parametrów δ i ε może nie wystarczyć dla metod innych niż bisekcja. Na przykład dla f_2 przy niewłaściwym x_0 styczne nadal będą odsuwać kolejne przybliżenia coraz dalej, a w końcu znajdzie się wystarczająco duży x , dla którego wartość funkcji jest mniejsza od epsilon lub nawet wpada w zero maszynowe.