

Obliczenia naukowe. Sprawozdanie z listy 3.

Maciej Bazela 261743

26 listopad 2022

# Spis treści

|          |   |          |
|----------|---|----------|
| <b>1</b> | <b>Zadania 1. 2. 3.</b>                   | <b>2</b> |
| 1.1      | Opis problemu . . . . .                   | 2        |
| 1.2      | Metoda bisekcji (połowienia) . . . . .    | 2        |
| 1.2.1    | Przebieg algorytmu . . . . .              | 2        |
| 1.3      | Metoda Newtona (stycznych) . . . . .      | 2        |
| 1.3.1    | Przebieg algorytmu . . . . .              | 2        |
| 1.4      | Metoda siecznych . . . . .                | 3        |
| 1.4.1    | Przebieg algorytmu . . . . .              | 3        |
| 1.5      | Testy poprawności implementacji . . . . . | 3        |
| <b>2</b> | <b>Zadanie 4</b>                          | <b>4</b> |
| 2.1      | Opis problemu . . . . .                   | 4        |
| 2.2      | Rozwiązanie . . . . .                     | 4        |
| 2.3      | Wyniki . . . . .                          | 4        |
| 2.4      | Wnioski . . . . .                         | 4        |
| <b>3</b> | <b>Zadanie 5</b>                          | <b>4</b> |
| 3.1      | Opis problemu . . . . .                   | 4        |
| 3.2      | Rozwiązanie . . . . .                     | 5        |
| 3.3      | Wyniki . . . . .                          | 5        |
| 3.4      | Wnioski . . . . .                         | 5        |
| <b>4</b> | <b>Zadanie 6</b>                          | <b>5</b> |
| 4.1      | Opis problemu . . . . .                   | 5        |
| 4.2      | Rozwiązanie . . . . .                     | 6        |
| 4.3      | Wyniki . . . . .                          | 6        |
| 4.4      | Wnioski . . . . .                         | 7        |

## 1 Zadania 1. 2. 3.

### 1.1 Opis problemu

Zadania 1. 2. 3. polegały na zaimplementowaniu algorytmów przybliżania pierwiastka funkcji  $f(x) = 0$ :

1. Przybliżanie  $f(x)$  metodą bisekcji,
2. Przybliżanie  $f(x)$  metodą Newtona (stycznych),
3. Przybliżanie  $f(x)$  metodą siecznych.

### 1.2 Metoda bisekcji (połowienia)

Funkcja  $f$  musi spełniać poniższe założenia:

- $f$  jest ciągła na przedziale  $[a, b]$ ,
- $f$  przyjmuje różne znaki w punktach  $a$  i  $b$  ( $f(a) \cdot f(b) < 0$ ).

#### 1.2.1 Przebieg algorytmu

Liczymy środek dla przedziału  $[a, b]$ . Oznaczmy go jako  $middle = \frac{a+b}{2}$ .

Jeśli  $f(a) \cdot f(middle) < 0$  (funkcja przyjmuje różne znaki na przedziale  $[a, middle]$ ) to znaczy, że miejsce zerowe musi znajdować się gdzieś w przedziale  $[a, middle]$ .

W przeciwnym przypadku musi być spełnione  $f(b) \cdot f(middle) < 0$ . W takim razie pierwiastek zawiera się w przedziale  $[middle, b]$ .

W kolejnym kroku iteracji, w zależności od powyższych warunków, podstawiamy  $middle$  za  $a$  lub  $b$  i kontynuujemy obliczenia.

Warunkiem stopu jest osiągnięcie zadanej dokładności przybliżenia  $|f(middle)| \leq \delta$  lub  $|b - a| \leq \epsilon$ . Zarówno  $\delta$  jak i  $\epsilon$  są parametrami implementowanej metody.

Implementacja tej metody znajduje się w pliku **iterative\_root\_approx.jl**, w funkcji **mbisekcji**.

### 1.3 Metoda Newtona (stycznych)

Funkcja  $f$  musi spełniać poniższe założenia:

- $f \in C^2[a, b]$  ( $f, f', f''$  są ciągłe na przedziale  $[a, b]$ ;  $f$  jest dwukrotnie różniczkowalna),
- $f'(r) \neq 0$  ( $r$  jest pierwiastkiem jednokrotnym).

#### 1.3.1 Przebieg algorytmu

Metoda Newtona polega na linearyzacji dwóch pierwszych składników szeregu Taylora:

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

Gdzie  $x_n$  jest  $n$ -tym przybliżeniem pierwiastka funkcji  $f$ , zadany wzorem:

$$n \geq 0$$
$$x_{n+1} = x_n - \frac{f(x_n)}{f'(x_n)}$$

$x_0$  jest przybliżeniem początkowym podawanym jako parametr funkcji.

Algorytm zaczynamy od sprawdzenia, czy  $|f'(x_0)| \neq 0$  (jeśli tak, zwracamy błąd) oraz czy  $|f(x_0)| \leq \epsilon$  dla zadanej dokładności  $\epsilon$ . Jeśli  $|f(x_0)| \leq \epsilon$  to szukanym pierwiastkiem jest  $x_0$ , czyli nie wykonujemy dalszych obliczeń.

W przeciwnym wypadku będziemy obliczać kolejne wartości  $x_n$ . W każdym kroku pętli sprawdzamy, czy  $|f'(x_n)| \leq \epsilon$  (jeśli tak, zwracamy błąd) oraz czy  $|f(x_n)| \leq \epsilon$  lub  $|x_n - x_{n-1}| \leq \delta$  dla zadanych dokładności  $\epsilon$  i  $\delta$ .

Jako parametr funkcji podajemy też maksymalną liczbę iteracji (*maxit*). Jeśli przekroczymy tę wartość, zwracamy odpowiedni błąd.

*Implementacja tej metody znajduje się w pliku **iterative\_root\_approx.jl**, w funkcji **mstycznych**.*

## 1.4 Metoda siecznych

Funkcja  $f$  musi spełniać poniższe założenia:

- $f \in C^2[a, b]$  ( $f$ ,  $f'$ ,  $f''$  są ciągłe na przedziale  $[a, b]$ ;  $f$  jest dwukrotnie różniczkowalna),
- $f'(r) \neq 0$  ( $r$  jest pierwiastkiem jednokrotnym).

### 1.4.1 Przebieg algorytmu

Metoda siecznych polega na wyznaczaniu siecznej przechodzącej przez punkty  $(x_n, f(x_n))$  i  $(x_{n-1}, f(x_{n-1}))$ . Za przybliżoną wartość pierwiastka przyjmujemy punkt przecięcia siecznej z osią  $OX$ .

W tej metodzie nie musimy obliczać  $f'(x_n)$ , ponieważ aproksymujemy pochodną poprzez:

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

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

$$n \geq 1$$

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

$x_0$  i  $x_1$  są dane jako parametr funkcji.

Obliczenia wykonujemy, aż do osiągnięcia zadanych dokładności  $\delta$  i  $\epsilon$ :  $|x_{n+1} - x_n| \leq \delta$  lub  $|f(x_{n+1})| \leq \epsilon$ .

Podobnie jak w *metodzie stycznych* wykonujemy maksymalnie *maxit* iteracji. Jeśli przekroczymy ten próg: zwracamy błąd.

*Implementacja tej metody znajduje się w pliku **iterative\_root\_approx.jl**, w funkcji **msiecznych**.*

## 1.5 Testy poprawności implementacji

Aby sprawdzić poprawność zaimplementowanych przeze mnie powyższych metod:

- wybrałem parę prostych funkcji,
- sprawdziłem ich pierwiastki przy pomocy serwisu **WolframAlpha**,
- na podstawie wyliczeń z zaufanego serwisu wybrałem odpowiednie przedziały i wartości początkowe dla wszystkich metod,

- sprawdziłem wyniki zwracane przez moje implementacje i porównałem je z wyliczeniami z serwisu.

Testy dla tych funkcji wpisałem do pliku *test\_iterative\_root\_approx.jl*.

Dla każdej metody sprawdzałem także przypadki, które powodują zwrócenie błędu.

Do sprawdzenia poprawności wykorzystałem moduł *Test*.

## 2 Zadanie 4

### 2.1 Opis problemu

W zadaniu 4. należało wyznaczyć pierwiastek równania  $\sin(x) - (\frac{1}{2}x)^2 = 0$  przy użyciu zaprogramowanych metod.

Zadane parametry:

1. *metoda bisekcji*: przedział początkowy  $[1.5, 2]$ ,  $\delta = \frac{1}{2}10^{-5}$ ,  $\epsilon = \frac{1}{2}10^{-5}$ ,
2. *metoda Newtona*: przybliżenie początkowe  $x_0 = 1.5$ ,  $\delta = \frac{1}{2}10^{-5}$ ,  $\epsilon = \frac{1}{2}10^{-5}$ ,
3. *metoda siecznych*: przybliżenia początkowe  $x_0 = 1$ ,  $x_1 = 2$ ,  $\delta = \frac{1}{2}10^{-5}$ ,  $\epsilon = \frac{1}{2}10^{-5}$ ,

### 2.2 Rozwiązanie

Kod źródłowy rozwiązań do tego zadania znajduje się w pliku *zad4.jl*.

Rozwiązanie polega na uruchomieniu metod dla podanych parametrów. Po uruchomieniu pliku źródłowego wyniki drukowane są do konsoli.

### 2.3 Wyniki

| Metoda    | root               | $f(\text{root})$           | Liczba iteracji |
|-----------|--------------------|----------------------------|-----------------|
| Bisekcji  | 1.9337539672851562 | $-2.7027680138402843e - 7$ | 16              |
| Newtona   | 1.933753779789742  | $-2.2423316314856834e - 8$ | 4               |
| Siecznych | 1.933753644474301  | $1.564525129449379e - 7$   | 4               |

Tabela 1: Wartości przybliżenia pierwiastka równania  $\sin(x) - (\frac{1}{2}x)^2 = 0$  dla wszystkich metod.

### 2.4 Wnioski

Wyliczone przybliżenia są podobne dla wszystkich metod. Można zauważyć, że *metoda bisekcji* wykonała czterokrotnie więcej iteracji niż *metoda Newtona* i *metoda siecznych*.

*Metody Newtona i siecznych*, mimo bardziej rygorystycznych warunków, szybciej osiągają zadaną dokładność.

## 3 Zadanie 5

### 3.1 Opis problemu

W zadaniu 5. należało przy pomocy *metody bisekcji* znaleźć wartości  $x$ , dla których przecinają się wykresy funkcji  $y = 3x$  i  $y = e^x$ .

Zadane dokładności obliczeń:  $\delta = 10^{-4}$ ,  $\epsilon = 10^{-4}$ .

## 3.2 Rozwiązanie

Kod źródłowy rozwiązań do tego zadania znajduje się w pliku `zad5.jl`.

Wartości  $x$  przecięcia obu funkcji można wyznaczyć za pomocą poniższego wzoru:

$$\begin{aligned}e^x &= 3x \\ e^x - 3x &= 0\end{aligned}$$

Realne wartości miejsc zerowych funkcji  $e^x - 3x$  sprawdziłem przy pomocy serwisu **WolframAlpha**. Wynoszą one  $x_1 \approx 0.619061\dots$  oraz  $x_2 \approx 1.512134\dots$

Dla powyższych wartości dobrałem odpowiednie przedziały, aby wykonać możliwie najmniej iteracji przy użyciu *metody bisekcji*. Były to kolejno przedziały:  $[0.5, 0.7]$  oraz  $[1.4, 1.6]$ .

## 3.3 Wyniki

| $x_i$ | root               | $f(\text{root})$        | Liczba iteracji |
|-------|--------------------|-------------------------|-----------------|
| $x_1$ | 0.6191406249999999 | $-9.066320343253942e-5$ | 9               |
| $x_2$ | 1.512109375        | $-3.868007140983565e-5$ | 9               |

Tabela 2: Wyliczone pierwiastki równania  $e^x - 3x = 0$  dla *metody bisekcji*.

## 3.4 Wnioski

*Metoda bisekcji* zwróciła poprawne wyniki dla zadanych przedziałów i dokładności.

Warto jednak zauważyć, że dobranie odpowiedniego przedziału wymagało od nas znajomości realnych wartości pierwiastka lub wykresu funkcji  $e^x - 3x$ . Gdybyśmy nie mogli sprawdzić wcześniej jakiego wyniku powinniśmy się spodziewać, byłoby nam trochę trudniej dobrać odpowiedni przedział. Mogłoby to skutkować zwiększeniem liczby wymaganych iteracji do osiągnięcia zadanej dokładności, lub zwracaniem błędów przez nieodpowiednie dobranie wartości granicznych przedziałów.

Dlatego ważny jest przeanalizowanie zadanego problemu oraz odpowiedni dobór danych wejściowych przy korzystaniu z metod przybliżających pierwiastki równań.

# 4 Zadanie 6

## 4.1 Opis problemu

W zadaniu 6. mieliśmy znaleźć miejsca zerowe funkcji:

$$\begin{aligned}f_1(x) &= e^{1-x} - 1 \\ f_2(x) &= xe^{-x}\end{aligned}$$

za pomocą *metody bisekcji*, *Newtona* i *siecznych*.

Zadane dokładności obliczeń:  $\delta = 10^{-5}$ ,  $\epsilon = 10^{-5}$ .

Sami mieliśmy dobrać odpowiednie przedziały i przybliżenia początkowe.

Powinniśmy także sprawdzić, co się stanie, gdy w *metodzie Newtona* dla  $f_1$  dobierzemy  $x_0 \in (1, \infty]$  i dla  $f_2$  wybierzemy  $x_0 > 1$ , oraz odpowiedzieć na pytanie, czy można przyjąć  $x_0 = 1$  dla  $f_2$ ?

## 4.2 Rozwiązanie

Kod źródłowy rozwiązań do tego zadania znajduje się w pliku *zad6.jl*.

Rozwiązanie polega na uruchomieniu zaimplementowanych metod dla podanych wartości startowych.

Można zauważyć, że dla  $f_1$  :  $root = 1$ , a dla  $f_2$  :  $root = 0$ . W takim razie wybrałem poniższe przedziały i przybliżenia początkowe:

$f_1$ :

1. *metoda bisekcji*: przedział początkowy  $[0.2, 2]$ ,  $\delta = 10^{-5}$ ,  $\epsilon = 10^{-5}$ ,
2. *metoda Newtona*: przybliżenie początkowe  $x_0 = 0.9$ ,  $\delta = 10^{-5}$ ,  $\epsilon = 10^{-5}$ ,
3. *metoda siecznych*: przybliżenia początkowe  $x_0 = 0.2$ ,  $x_1 = 2$ ,  $\delta = 10^{-5}$ ,  $\epsilon = 10^{-5}$ ,

$f_2$ :

1. *metoda bisekcji*: przedział początkowy  $[-1, 0.5]$ ,  $\delta = 10^{-5}$ ,  $\epsilon = 10^{-5}$ ,
2. *metoda Newtona*: przybliżenie początkowe  $x_0 = 0.9$ ,  $\delta = 10^{-5}$ ,  $\epsilon = 10^{-5}$ ,  $maxit = 1000$ ,
3. *metoda siecznych*: przybliżenia początkowe  $x_0 = -1$ ,  $x_1 = 0.5$ ,  $\delta = 10^{-5}$ ,  $\epsilon = 10^{-5}$ ,  $maxit = 1000$ ,

## 4.3 Wyniki

| Metoda    | root               | $f(root)$                 | Liczba iteracji |
|-----------|--------------------|---------------------------|-----------------|
| Bisekcji  | 0.9999969482421875 | $3.0517624691750456e - 6$ | 16              |
| Newtona   | 0.99999999931772   | $6.822808984452422e - 11$ | 3               |
| Siecznych | 1.0000003998949842 | $-3.998949043015898e - 7$ | 6               |

Tabela 3: Wartości przybliżeń pierwiastku dla funkcji  $f_1$

| Metoda    | root                       | $f(root)$                  | Liczba iteracji |
|-----------|----------------------------|----------------------------|-----------------|
| Bisekcji  | $7.62939453125e - 6$       | $7.62933632381113e - 6$    | 16              |
| Newtona   | $-1.302601427744421e - 7$  | $-1.30260159742148e - 7$   | 15              |
| Siecznych | $-1.1737426154042664e - 6$ | $-1.1737439930768023e - 6$ | 7               |

Tabela 4: Wartości przybliżeń pierwiastku dla funkcji  $f_2$

| $x_0$ | root               | $f(root)$                 | Liczba iteracji | Kod błędu |
|-------|--------------------|---------------------------|-----------------|-----------|
| 2     | 0.9999999810061002 | $1.8993900008368314e - 8$ | 5               | 0         |
| 3     | 0.9999999710783241 | $2.892167638712806e - 8$  | 9               | 0         |
| 4     | 0.999999995278234  | $4.721767421500545e - 10$ | 21              | 0         |
| 5     | 0.9999996427095682 | $3.572904956339329e - 7$  | 54              | 0         |
| 6     | 0.9999999573590406 | $4.264096031825204e - 8$  | 147             | 0         |
| 7     | 0.9999999484165362 | $5.15834650549607e - 8$   | 401             | 0         |
| 8     | 0.0                | 0.0                       | 0               | 1         |
| 9     | 0.0                | 0.0                       | 0               | 1         |
| 10    | 0.0                | 0.0                       | 0               | 1         |
| 11    | 0.0                | 0.0                       | 0               | 1         |
| 12    | 0.0                | 0.0                       | 0               | 1         |

Tabela 5: Wartości przybliżeń pierwiastku funkcji  $f_1$  i  $x_0 > 1$  dla *metody Newtona*

| $x_0$ | root               | $f(\text{root})$          | Liczba iteracji | Kod błędu |
|-------|--------------------|---------------------------|-----------------|-----------|
| 2     | 0.0                | 0.0                       | 0               | 2         |
| 3     | 14.366810590034204 | $8.278159774083757e - 6$  | 5               | 0         |
| 4     | 0.0                | 0.0                       | 0               | 2         |
| 5     | 0.0                | 0.0                       | 0               | 2         |
| 6     | 15.356587449552173 | $3.2886187381569198e - 6$ | 3               | 0         |
| 7     | 0.0                | 0.0                       | 0               | 2         |
| 8     | 14.96875332463598  | $4.7243129834432185e - 6$ | 2               | 0         |
| 9     | 16.839847490215476 | $8.182486791183465e - 7$  | 2               | 0         |
| 10    | 0.0                | 0.0                       | 0               | 2         |
| 11    | 15.046673852885865 | $4.39291785343123e - 6$   | 1               | 0         |
| 12    | 16.414553294057306 | $1.220335231860097e - 6$  | 1               | 0         |
| 13    | 0.0                | 0.0                       | 0               | 2         |
| 14    | 0.0                | 0.0                       | 0               | 2         |
| 15    | 0.0                | 0.0                       | 0               | 2         |
| 16    | 0.0                | 0.0                       | 0               | 2         |
| 17    | 0.0                | 0.0                       | 0               | 2         |
| 18    | 0.0                | 0.0                       | 0               | 2         |
| 19    | 0.0                | 0.0                       | 0               | 2         |
| 20    | 0.0                | 0.0                       | 0               | 2         |

Tabela 6: Wartości przybliżeń pierwiastku funkcji  $f_2$  i  $x_0 > 1$  dla *metody Newtona*

#### 4.4 Wnioski

Dla dobrze dobranych przedziałów i początkowych przybliżeń wszystkie metody dają poprawne wyniki. *Metoda bisekcji* znowu wykonywała najwięcej iteracji. *Metoda Newtona* i *metoda siecznych* szybciej znajdują przybliżenia pierwiastka dla danej dokładności.

Jednak przy testach *metody Newtona* dla  $x_0 > 1$  możemy zauważyć dziwne anomalie.

W przypadku  $f_1$  *metoda Newtona* co prawda dla  $1 < x_0 < 8$  daje dobre wyniki, ale wraz ze wzrostem wartości  $x_0$  bardzo szybko rośnie liczba iteracji potrzebnych do osiągnięcia zadanej dokładności. Dla  $x_0 = 8, 9, \dots$  zwracany jest kod błędu 1, który oznacza, że przekroczyliśmy maksymalną liczbę iteracji. Dzieje się tak, ponieważ  $\lim_{x \rightarrow \infty} f'_1(x) = 0$ , przez co  $\frac{f(x)}{f'(x)}$  szybko rośnie.

W przypadku  $f_2$  dla dużych wartości  $x$  funkcja  $f_2$  jest bliska 0 (problematyczny jest czynnik  $e^{-x}$ ). Tutaj także  $\lim_{x \rightarrow \infty} f'_2(x) = 0$ . Dla większości wartości  $x_0$  zwracany jest kod błędu 2 - pochodna bliska zeru.

Dla  $x_0 = 1$   $f'_2(x_0) = -e^{-x_0}(x_0 - 1) = \frac{-1}{e}(1 - 1) = 0$ . Nie możemy użyć tutaj *metody Newtona*, ponieważ  $f'(x_n) \neq 0$ .

Zadanie to pokazuje jak ważne jest odpowiednie dobranie parametrów startowych dla implementowanych metod. Nawet najmniejsza zmiana początkowego przedziału lub przybliżenia może mieć drastyczny wpływ na czas wykonywania obliczeń.

Co więcej, odpowiednie dobranie dokładności  $\delta$  i  $\epsilon$  też ma znaczenie, bo na przykładzie  $f_2$  i  $x_0 > 1$  możemy zauważyć, że otrzymaliśmy wyniki daleko odbiegające od prawdziwego pierwiastka  $f_2$ . Spowodowane jest to tym, że od pewnego momentu wykres funkcji  $f_2$  przybliża się bardzo blisko osi  $OX$ .