

## Zadanie 1

### Problem

W zadaniu należało zaimplementować w języku Julia algorytm wyznaczający miejsca zerowe funkcji za pomocą metody bisekcji.

### Pseudokod algorytmu

```
Dane:  $a, b, M, \delta, \epsilon$   
Wyniki:  $k, \tilde{r}, f(\tilde{r})$   
   $u \leftarrow f(a); v \leftarrow f(b);$   
   $e \leftarrow b - a;$   
  if  $\text{sgn}(u) = \text{sgn}(v)$  then  
    return error;  
  end if  
  for  $k \leftarrow 1$  to  $M$  do  
     $e \leftarrow e/2;$   
     $c \leftarrow a + e;$   
     $w \leftarrow f(c);$   
    if  $|e| < \delta$  or  $|w| < \epsilon$  then  
      return  $k, c, w$   
    end if  
    if  $\text{sgn}(w) \neq \text{sgn}(u)$  then  
       $b \leftarrow c; v \leftarrow w;$   
    else  
       $a \leftarrow c; u \leftarrow w;$   
    end if  
  end for
```

### Opis parametrów

$a, b$  – końce przedziału początkowego

$\delta, \epsilon$  – dokładność obliczeń

$M$  – maksymalna liczba iteracji

### Dane wynikowe

$k$  – liczba wykonanych iteracji (w programie it)

$c$  – przybliżenie pierwiastka równania  $f(x) = 0$  (w programie r)

$w$  – wartość  $f(x)$  (w programie v)

### Opis algorytmu

Przed przystąpieniem do algorytmu trzeba przyjąć założenia, że funkcja  $f$  jest ciągła na przedziale  $[a, b]$  oraz  $f(a)f(b) < 0$  (czyli funkcja zmienia w tym przedziale znak).

Funkcja rozpoczyna działanie pętli, której warunkiem zakończenia jest znalezienie przedziału mniejszego niż epsilon (lub wykonanie maksymalnej liczby iteracji). W każdym przebiegu pętli

wyznaczany jest środek bieżącego przedziału oraz liczona wartość funkcji  $f$  dla wyznaczonego środka. Jeżeli wartość funkcji jest wystarczająco bliska 0 to zwracamy ostatni znaleziony środek, w przeciwnym przypadku kontynuujemy działanie algorytmu dla tej połowy przedziału, w której funkcja zmienia znak i tą połowę traktujemy jako nowy przedział.

## Zadanie 2

### Problem

W zadaniu należało zaimplementować w języku Julia algorytm wyznaczający miejsca zerowe funkcji za pomocą metody stycznych (czyli metody Newtona).

### Pseudokod algorytmu

**Dane:**  $x_0, M, \delta, \epsilon$

**Wyniki:**  $k, \tilde{r}, f(\tilde{r})$

$v \leftarrow f(x_0);$

**if**  $|v| < \epsilon$  **then**

**return** 0,  $x_0, v$ ;

**end if**

**for**  $k \leftarrow 1$  to  $M$  **do**

$x_1 \leftarrow x_0 - v/f'(x_0);$

$v \leftarrow f(x_1);$

**if**  $|x_1 - x_0| < \delta$  **or**  $|v| < \epsilon$  **then**

**return**  $k, x_1, v$ ;

**end if**

$x_0 \leftarrow x_1;$

**end for**

### Opis parametrów

$x_0$  – przybliżenie początkowe

delta, epsilon – dokładność obliczeń

$M$  – maksymalna liczba iteracji (w programie maxit)

### Dane wynikowe

$k$  – liczba wykonanych iteracji (w programie it)

$x_0/x_1$  – przybliżenie pierwiastka równania  $f(x) = 0$  (w programie r)

$v$  – wartość  $f(x)$

### Opis algorytmu

W metodzie Newtona przyjmuje się następujące założenia dla funkcji  $f$ : w przedziale  $[a,b]$  znajduje się dokładnie jeden pierwiastek, funkcja ma różne znaki na krańcach przedziału, czyli  $f(a)f(b) < 0$  i pierwsza i druga pochodna funkcji mają stały znak w tym przedziale.

Metoda stycznych wykorzystuje metodę linearyzacji, czyli zastąpienia funkcji  $f$  funkcją liniową, będącą sumą dwóch początkowych składników wzoru Taylora dla funkcji  $f$ .

Wzór Taylora:  $f(x) = f(c) + f'(c)(x-c) + \frac{1}{2!}f''(c)(x-c)^2 + \dots$

Funkcja liniowa:  $l(x) = f(c) + f'(c)(x-c)$

Do obliczania kolejnych przybliżeń pierwiastka zadanej funkcji wykorzystywany jest następujący

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

Na początku algorytm sprawdza czy pochodna funkcji  $f$  nie jest bliska zero (czyli mniejsza od epsilon).

Program oblicza przybliżenie korzystając z podanego wzoru na przybliżenie liniowe. Wyznaczana jest wartość  $x_{n+1}$  oraz wartość funkcji  $f$  dla tego argumentu. Jeżeli otrzymane przybliżenie jest już wystarczająco dokładne, to zwracany jest wynik, w przeciwnym przypadku (jeżeli nie została osiągnięta maksymalna liczba iteracji) algorytm wykonuje następną iterację.

### Zadanie 3

Problem

W zadaniu należało zaimplementować w języku Julia algorytm wyznaczający miejsca zerowe funkcji za pomocą metody siecznych.

Pseudokod algorytmu

**Dane:**  $a, b, M, \delta, \epsilon$

**Wyniki:**  $k, \tilde{r}, f(\tilde{r})$

```
fa ← f(a); fb ← f(b);
for k ← 1 to M do
  if |fa| > |fb| then
    a ↔ b; fa ↔ fb;
  end if
  s ← (b - a)/(fb - fa);
  b ← a; fb ← fa;
  a ← a - fa * s;
  fa ← f(a);
  if |b - a| < δ or |fa| < ε then
    return k, a, fa;
  end if
end for
```

Opis parametrów

$a, b$  – końce przedziału początkowego (w programie  $x_0, x_1$ )

$\delta, \epsilon$  – dokładność obliczeń

$M$  – maksymalna liczba iteracji (w programie  $\text{maxit}$ )

Dane wynikowe

$k$  – liczba wykonanych iteracji (w programie  $\text{it}$ )

$c$  – przybliżenie pierwiastka równania  $f(x) = 0$  (w programie  $r$ )

$w$  – wartość  $f(x)$  (w programie  $v$ )

Opis algorytmu

Metoda siecznych jest modyfikacją metody stycznych. Pochodna funkcji  $f$  została zastąpiona ilorazem różnicowym:  $f'(x_n) \approx \frac{f(x_n) - f(x_{n-1})}{x_n - x_{n-1}}$

Stąd kolejne przybliżenia pierwiastka opisane są wzorem:  $x_{n+1} = x_n - f(x_n) \frac{x_n - x_{n-1}}{f(x_n) - f(x_{n-1})}$

W tym przypadku styczna z metody stycznych zostaje zastąpiona sieczną wykresu funkcji  $f$ , więc na wejściu podajemy dwa początkowe punkty przecięcia wykresu  $x_0$  i  $x_1$ .

Algorytm rozpoczyna działanie od wyliczenia wartości funkcji  $f$  dla zadanych  $x_0$  i  $x_1$ .

Następnie rozpoczyna pętlę, w której sprawdza warunek  $|f_{x_0}| > |f_{x_1}|$ . Jeżeli nierówność jest spełniona to zmienne zostają zamienione, żeby moduły funkcji w kolejnych punktach nie rosły. Dalej obliczamy iloraz różnicowy i kolejny  $x_{n+1}$ . Jeżeli przybliżenie funkcji jest wystarczająco dokładne, czyli odległość punktów przecięcia funkcji przez sieczną jest mniejsza niż delta albo wartość funkcji dla bieżącego argumentu jest bliska zeru, czyli mniejsza niż epsilon, to algorytm zwraca wynik. W przeciwnym wypadku algorytm kontynuuje działanie w pętli.

## Zadanie 4

Problem

Zadanie polegało na wyznaczeniu pierwiastków równania  $\sin x - \left(\frac{1}{2}X\right)^2 = 0$  za pomocą metod zaimplementowanych w poprzednich zadaniach.

Wyniki

metoda	przybliżenie pierwiastka - $x_0$	$f(x_0)$	liczba iteracji
bisekcji	1.9337539672851562	-2.7027680138402843e-7	16
stycznych	1.933753779789742	-2.2423316314856834e-8	4
siecznych	1.933753644474301	1.564525129449379e-7	4

Obserwacje i wnioski

Metoda bisekcji potrzebowała wykonać więcej iteracji do obliczenia pierwiastka niż dwie pozostałe i zwróciła wynik z największą dokładnością.

Uzyskane wyniki potwierdzają zbieżność każdej z metod determinowaną przez współczynnik zbieżności. Dla metody bisekcji otrzymano zbieżność kwadratową, dla metody stycznych zbieżność liniową i dla metody siecznych zbieżność nadliniową. Metoda bisekcji, mimo że zbiega najwolniej spośród wszystkich trzech metod, to jest to metoda najstabilniejsza, gdyż uzyskaliśmy w jej przypadku wyniki najbardziej zbliżone do zadanej na wejściu dokładności. Metody stycznych oraz siecznych są najszybciej zbieżne.

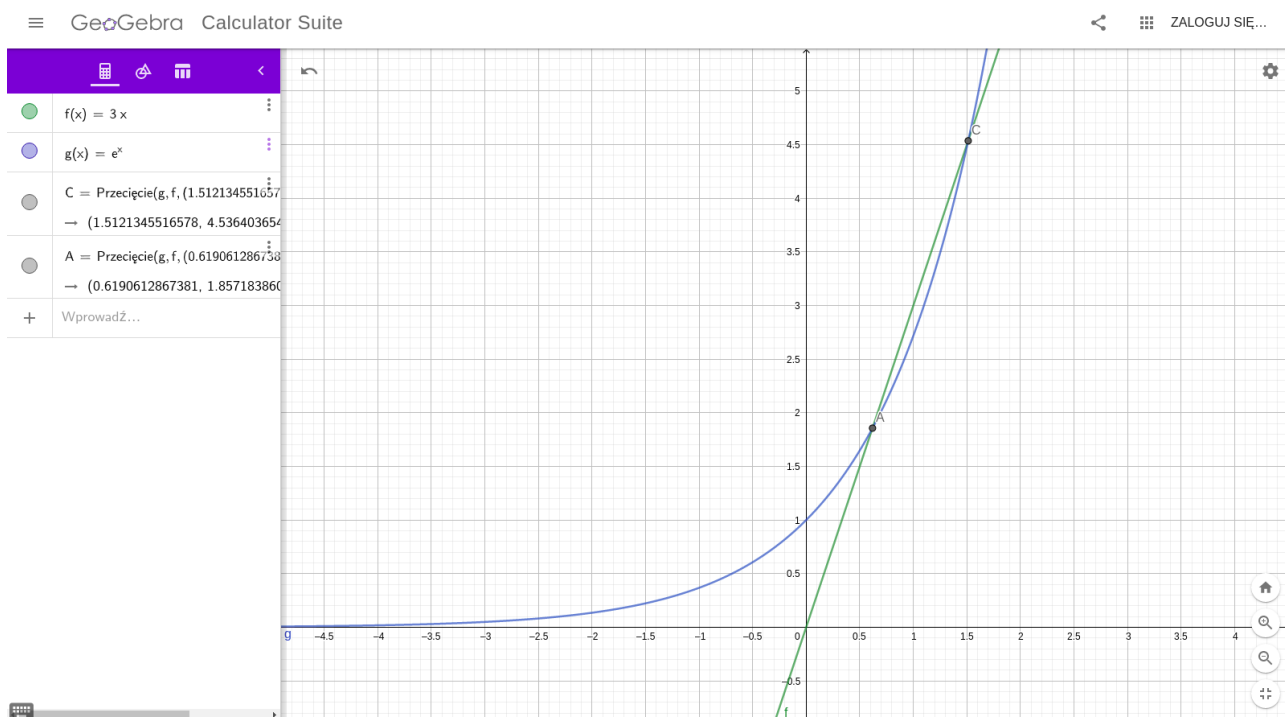
## Zadanie 5

Problem

Zadanie polegało na znalezieniu wartości zmiennej  $x$ , dla której przecinają się wykresy funkcji  $y = 3x$  i  $y = e^x$  wykorzystując metodę bisekcji z zadanymi dokładnościami  $10^{-4}$ .

Rozwiązanie

Na podstawie wykresu wygenerowanego w GeoGebraze stwierdzamy, że  $x$  należy szukać w przedziale  $[0, 2]$ .



Problem sprowadzamy do szukania miejsc zerowych funkcji pomocniczej będącej różnicą wejściowych funkcji:  $f(x) = e^x - 3x$ . W celu znalezienia miejsc zerowych wywołujemy funkcję bisekcji dwa razy – na przedziałach:  $[0, 1]$  i  $[1, 2]$ , ponieważ na całym przedziale  $[0, 2]$  funkcja nie zmienia znaku.

### Wyniki

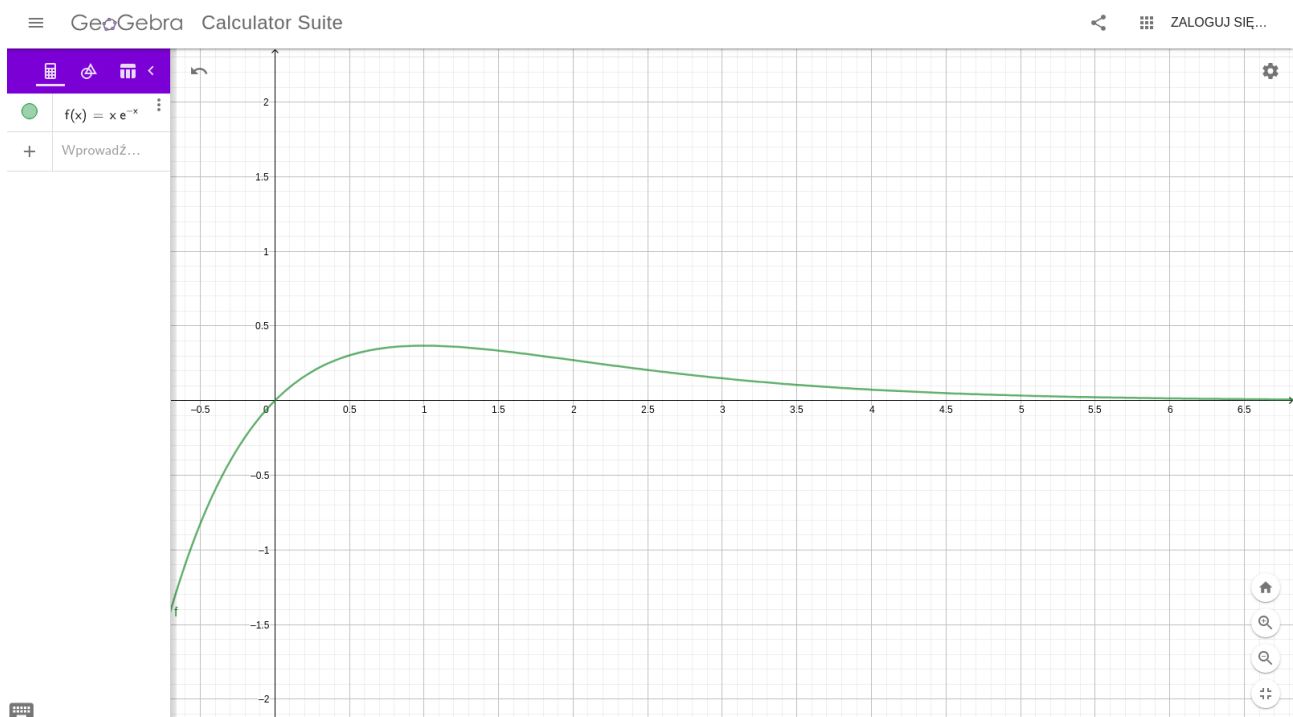
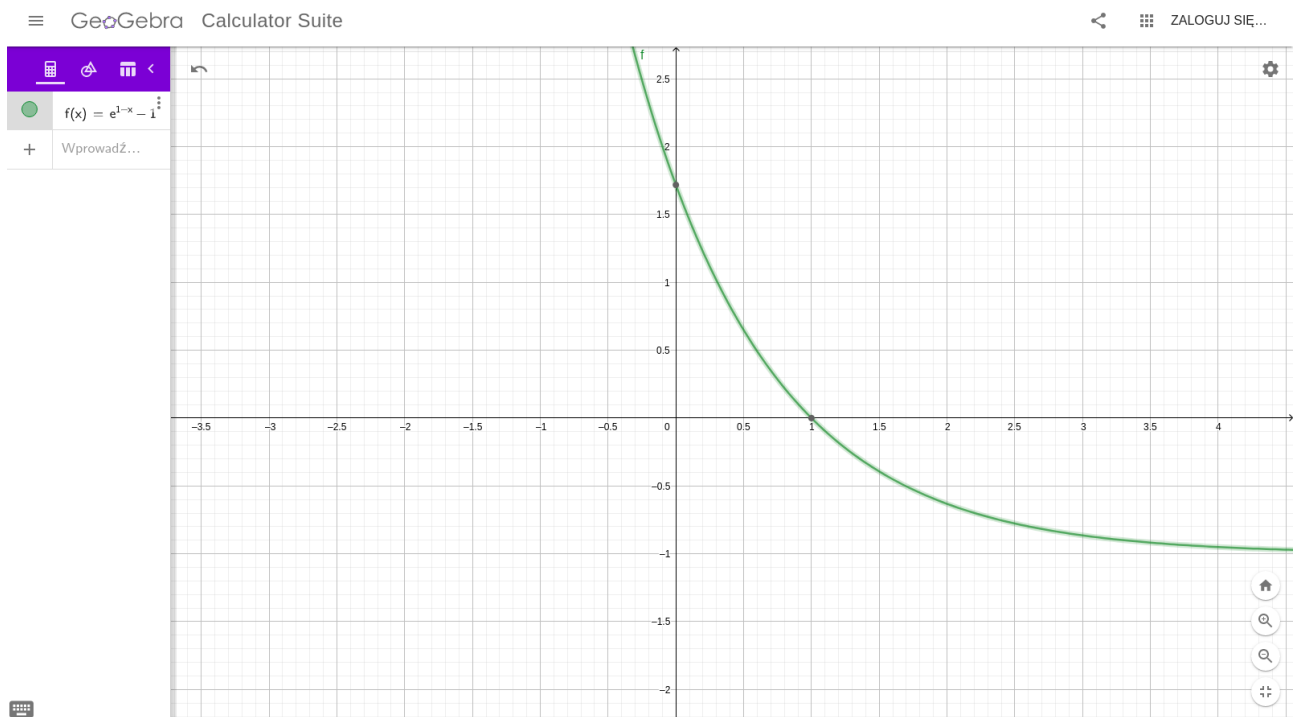
przedział	przybliżenie pierwiastka - $x_0$	$f(x_0)$	liczba iteracji
$[0, 1]$	0.61907958984375	-2.091677592419572e-5	14
$[1, 2]$	1.51214599609375	1.7583570236290313e-5	14

Znalezione pierwiastki są zgodne z wykresem. Dla odpowiednich przedziałów wejściowych metoda bisekcji działa poprawnie.

### Zadanie 6

#### Problem

W zadaniu należało znaleźć miejsca zerowe zadanych funkcji  $f_1(x) = e^{1-x} - 1$  oraz  $f_2(x) = xe^{-x}$  za pomocą wszystkich trzech uprzednio zaimplementowanych funkcji. Na początku rysujemy wykresy, żeby na ich podstawie wyznaczyć sobie przedziały do poszukiwania miejsc zerowych. Następnie testujemy funkcje na różnych przedziałach.



## Wyniki

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

Metoda bisekcji

[0.0,2.0]: (1.0, 0.0, 1, 0)

[-2.0,2.0]: (1.0, 0.0, 2, 0)

[0.1,1.8]: (0.9999992370605468, 7.62939744269886e-7, 17, 0)

[0.1,1.2]: (0.9999938964843748, 6.103534251789e-6, 14, 0)

[-0.2,1.8]: (0.9999969482421875, 3.0517624691750456e-6, 17, 0)

[-5.0,500.0]: (0.9999921917915344, 7.808238949635893e-6, 24, 0)

Metoda stycznych

$x_0 = -1.0$ : (0.9999922654776594, 7.734552252003368e-6, 5, 0)  
 $x_0 = 0.0$ : (0.9999984358892101, 1.5641120130194253e-6, 4, 0)  
 $x_0 = 1.1$ : (0.99999999991094, 8.905987058938081e-11, 3, 0)  
 $x_0 = 2.0$ : (0.9999999810061002, 1.8993900008368314e-8, 5, 0)  
 $x_0 = 6.0$ : (0.9999999573590406, 4.264096031825204e-8, 147, 0)  
 $x_0 = 8.0$ : (NaN, NaN, NaN, 1)  
 $x_0 = 15.0$ : (NaN, NaN, NaN, 1)

Metoda siecznych:

$x_0 = -2.0, x_1 = 2.0$ : (1.0000000080618678, -8.061867839970205e-9, 8, 0)  
 $x_0 = -0.3, x_1 = 1.8$ : (1.0000003464708873, -3.4647082725047795e-7, 6, 0)  
 $x_0 = 0.1, x_1 = 1.3$ : (0.999999935820667, 6.41793329592133e-9, 5, 0)  
 $x_0 = -2.0, x_1 = 6.0$ : (5.6042543962200835, -0.989990837907685, 3, 0)  
 $x_0 = -10.0, x_1 = 10.0$ : (9.99966600720658, -0.999876548971044, 3, 0)  
 $x_0 = 10.0, x_1 = 100.0$ : (10.0, -0.9998765901959134, 2, 0)

Funkcja  $f_2(x) = xe^{-x}$

Metoda bisekcji

$[-0.5, 0.5]$ : (0.0, 0.0, 1, 0)  
 $[-0.7, 0.4]$ : (4.57763671885093e-6, 4.5776157641409616e-6, 16, 0)  
 $[-10.0, 100.0]$ : (45.0, 1.2881333612472272e-18, 1, 0)

Metoda stycznych

$x_0 = -1.0$ : (-3.0642493416461764e-7, -3.0642502806087233e-7, 5, 0)  
 $x_0 = -0.4$ : (-1.8440313309425922e-8, -1.844031364947108e-8, 4, 0)  
 $x_0 = 0.0$ : (0.0, 0.0, 0, 2)  
 $x_0 = 1.0$ : (NaN, NaN, NaN, 1)  
 $x_0 = 6.0$ : (14.97432014974184, 4.699833827208111e-6, 8, 0)  
 $x_0 = 8.0$ : (14.636807965014, 6.438155219843286e-6, 6, 0)  
 $x_0 = 15.0$ : (15.0, 4.588534807527386e-6, 0, 2)

Metoda siecznych

$x_0 = -2.0, x_1 = 2.0$ : (14.294924723787231, 8.85064549833867e-6, 15, 0)  
 $x_0 = -0.3, x_1 = 1.8$ : (14.661140570698615, 6.293834366782407e-6, 14, 0)  
 $x_0 = 0.1, x_1 = 1.3$ : (2.4829988427991286e-7, 2.48299822627088e-7, 5, 0)  
 $x_0 = -2.0, x_1 = 6.0$ : (14.812321857602736, 5.466552122239313e-6, 12, 0)  
 $x_0 = -10.0, x_1 = 10.0$ : (9.99999958776927, 0.0004539993144685704, 1, 0)  
 $x_0 = 10.0, x_1 = 100.0$ : (100.0, 3.7200759760208363e-42, 1, 0)

Obserwacje i wnioski

Metoda bisekcji dla odpowiednio podanego przedziału (czyli takiego, żeby funkcja zmieniała znak na przedziale) zawsze znajduje poprawnie miejsca zerowe z określoną dokładnością. Jeżeli rozwiązanie leży pośrodku przedziału metoda znajduje je bardzo szybko, w przeciwnym przypadku może wykonywać dużo iteracji.

Metoda stycznych działa zauważalnie szybciej. Jeżeli wybierzemy  $x_0 \in [0, \infty]$  dla pierwszej funkcji ze wzrostem  $x_0$  bardzo szybko wzrasta liczba iteracji potrzebnych do obliczenia pierwiastka. Natomiast dla drugiej funkcji przekazanie  $x_0 = 1$  spowoduje zakończenie programu z errorem 1, ponieważ pochodna w tym miejscu jest bliska zeru. Z kolei zwiększanie wartości powyżej 1 powoduje zmniejszanie się dokładności przybliżenia wyniku. Metoda stycznych działa właściwie tylko przy dobrze dobranym początkowym przybliżeniu.

W metodzie siecznych dobranie odpowiednio bliskiego rozwiązaniu przedziału początkowego również wpływa znacząco na poprawność końcowego. Im bardziej ścisły przedział, tym mniej iteracji wymaga metoda.

Metody siecznych i stycznych są szybsze od metody bisekcji (szybkość zbieżności opisana w zadaniu 4). Natomiast metoda bisekcji wydaje się być najbardziej niezawodna. Wszystkie trzy metody wymagają wcześniejszej analizy danych i dobrania odpowiednich wartości wejściowych (przedziałów i początkowych przybliżeń), bo w przeciwnym przypadku mogą zwracać błędne wyniki. Warto porównać wyniki działania wszystkich trzech algorytmów, w celu wyeliminowania ryzyka błędów.