



AKADEMIA GÓRNICZO-HUTNICZA W KRAKOWIE

WYDZIAŁ FIZYKI I INFORMATYKI STOSOWANEJ

Metody Numeryczne

Laboratorium 06:

Andrzej Świętek

26.03.2024

Contents

1	Wstęp teoretyczny	1
1.1	Metoda Falsi	1
1.1.1	Opis	1
1.1.2	Algorytm:	1
1.2	Metoda Siecznych	3
1.2.1	Opis	3
1.2.2	Algorytm	3
1.2.3	Porównanie z metodą Falsi:	3
1.3	Modyfikacja Metody Siecznych	4
2	Problem	4
2.1	Zadania do wykonania:	4
3	Implementacja	5
3.1	Manualne zlokalizowanie Wszystkich Przedziałów Izolacji	5
3.2	Znalezienie pierwszego miejsca zerowego	6
3.3	Metoda zmodyfikowana	8
4	Wyniki	8
5	Wizualizacja Wyników	13
6	Wnioski	17

1. Wstęp teoretyczny

1.1. Metoda Falsi

1.1.1. Opis

Jest to metoda iteracyjna służąca do rozwiązywania równań nieliniowych. Pozwala na znajdowanie pierwiastków funkcji nieliniowych.

Cała metoda opiera się na fałszywym założeniu istnienia lokalnej liniowości funkcji (z tą nazwą).

1.1.2. Algorytm:

- Znalezienie przedziału izolacji gdzie wiemy, że znajduje się pierwiastek.**
Jeżeli $f(a) \cdot f(b) < 0$ oznacza to, że funkcja przechodzi przez oś OX w tym przedziale.
- Znalezienie środka przedziału - x a także $f(x)$**
W sposób analogiczny do metody bisekcji znajdujemy punkt będący środkiem przedziału

3. Prowadzenie prostej

Prowadzimy prostą przez punkty a i b zgodnie z równaniem ze szkoły:

$$y - f(x) = \frac{f(b) - f(a)}{b - a}(x - a), \quad y = 0$$

Wiemy że $y = 0$ bo jest to miejsce zerowe i następnie przekształcamy równanie na zmienną szukaną x .

$$x_1 = a - \frac{f(a)}{f(b) - f(a)}(b - a)$$

4. Sprawdzenie warunku końcowego

Jeżeli $f(x) = 0$ lub błąd jest akceptowalny to przerywamy działanie programu.

Kluczowym elementem tego algorytmu jest proces zwężania przedziału izolacji.

- Jeżeli $f(a) \cdot f(x) < 0 \implies$ **Zwężamy w lewo**
- Jeżeli $f(b) \cdot f(x) < 0 \implies$ **Zwężamy w prawo**
- W przeciwnym wypadku nie dążymy do miejsca zerowego i metoda w tym przedziale nie zadziała

Ustaliwszy kierunek zwężania przedziału, jeżeli zwężamy w lewo to aktualizujemy wartość $b \leftarrow x$ natomiast jeśli w prawo $a \leftarrow x$

$$x_0 = a$$

$$x_{k+1} = x_k - \frac{f(x_k)}{f(b) - f(x_k)}(b - x_k)$$

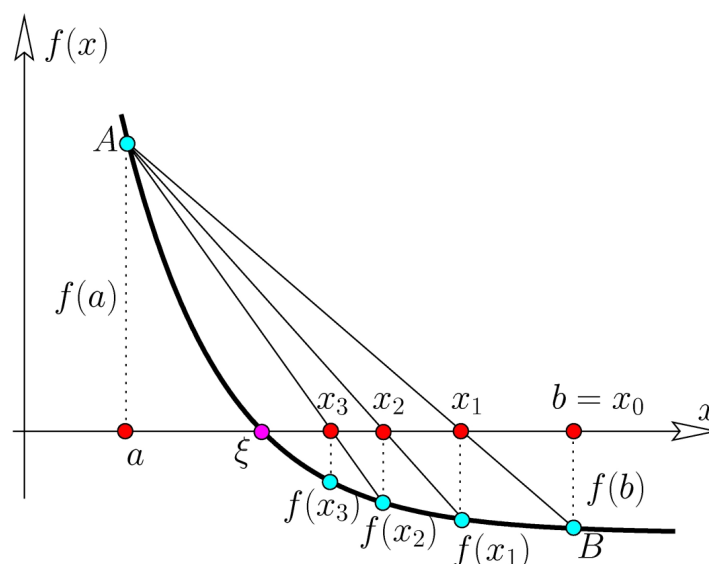


Figure 1: Kolejne iteracje metody Falsi

1.2. Metoda Siecznych

1.2.1. Opis

Metoda siecznych w dużym stopniu jest metodą analogiczną do metody Falsi.

Metoda siecznych jest jedną z metod iteracyjnych służących do rozwiązywania równań nieliniowych. Działa na zasadzie przybliżania miejsca zerowego funkcji poprzez stworzenie przybliżonej stycznej do wykresu funkcji i znalezienie jej przecięcia z osią OX.

Metoda siecznych jest iteracyjną metodą numeryczną służącą do znajdowania miejsc zerowych funkcji nieliniowych. Metoda ta nie wymaga znajomości pochodnych funkcji, co stanowi jej zaletę w porównaniu do metody Newtona-Raphsona. Zamiast tego, używa ona dwóch początkowych punktów (lub przybliżeń) do wyznaczenia kolejnych przybliżeń pierwiastka funkcji poprzez przybliżanie stycznej do wykresu funkcji.

1.2.2. Algorytm

1. Wybór początkowych przybliżeń:

Wybierz dwa punkty początkowe x_0 i x_1 , które są bliskie miejscu zerowemu funkcji.

2. Obliczenie kolejnych przybliżeń:

Wykorzystując punkty początkowe, oblicz kolejne przybliżenia pierwiastka funkcji za pomocą wzoru:

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

3. Sprawdzenie warunku stopu:

Kontynuuj obliczenia, dopóki różnica między kolejnymi przybliżeniami jest większa niż założona dokładność lub dopóki osiągnięto maksymalną liczbę iteracji.

1.2.3. Porównanie z metodą Falsi:

- Metoda Falsi (lub reguła fałszywej pozycji) również korzysta z dwóch punktów początkowych, ale w każdej iteracji wybiera punkt, który znajduje się na przecięciu osi OX z linią łączącą dwa punkty początkowe.
- Metoda Falsi wymaga, aby początkowe punkty były tak dobrane, aby miały różne znaki wartości funkcji, co zapewnia zbieżność. Natomiast metoda siecznych może być stosowana w przypadku, gdy znak wartości funkcji w początkowych punktach nie jest różny.
- Metoda Falsi może być bardziej podatna na oscylacje w przypadku funkcji o dużej zmienności, podczas gdy metoda siecznych jest bardziej stabilna.

1.3. Modyfikacja Metody Siecznych

Celem uzyskania lepszej zbieżności a zatem wyniku po mniejszej ilości iteracji wykonujemy modyfikację oryginalnego algorytmu polegającą na zastąpieniu funkcji $f(x)$ przez:

$$u(x) = \frac{f(x)}{f'(x)}$$

Samą wartość pochodnej w punkcie możemy przybliżyć ilorazem różnicowym - w tej metodzie 1 stopnia

(Metoda Newtona wymagałaby 2 co zaczyna wprowadzać poważne źródło błędu, lecz dla ilorazu różnicowego 1 stopnia błąd jest pomijalny)

$$\frac{\partial f(x)}{\partial x} = \frac{f(x + \Delta x) - f(x - \Delta x)}{2 \cdot \Delta x}$$

W wyniku powyższych modyfikacji uzyskany wzór iteracyjny wygląda następująco:

$$x_{k+1} = x_k - u(x) \cdot \frac{x_k - x_{k-1}}{x(x_k) - u(x_{k-1})}$$

2. Problem

Głównym problemem, który został postawiony było wyznaczenie wszystkich pierwiastków równania nieliniowego metodą siecznych

$$f(x) = x^4 - 7.899x^3 + 23.281114x^2 - 30.33468152x + 14.73866033$$

2.1. Zadania do wykonania:

1. Sporządzić wykres funkcji $f(x)$ w zakresie $x \in [1.5, 2.4]$
2. Na podstawie wykresu oraz postaci równania proszę określić krotność pierwiastków oraz oszacować ich przedziały izolacji.
3. Następnie proszę napisać program do wyznaczania pierwiastków równania nieliniowego:
 - (a) niemodyfikowaną metodą siecznych (pierwiastki o nieparzystej krotności)
 - (b) modyfikowaną metodą siecznych (pierwiastki o krotności parzystej)
- zastępujemy funkcję $f(x)$ przez $u(x) = \frac{f(x)}{f'(x)}$
4. Wyznaczyć wszystkie pierwiastki równania (ile ich jest?) przy pomocy swojego programu. Jako warunek zakończenia procesu iteracyjnego proszę przyjąć:

$$\epsilon_{i+1} = |x_{i+1} - x_i| < 10^{-6}$$

Dla każdego pierwiastka proszę stworzyć tabelkę, w której znajdą się informacje dotyczące położenia kolejnych przybliżeń, wartości ϵ_i oraz wartości funkcji i jej pierwszej pochodnej.

5. Przy modyfikacji metody siecznych konieczne jest wyznaczanie pochodnej funkcji $f(x)$. Proszę powtórzyć obliczenia dla pierwiastka wielokrotnego, zastępując pochodną funkcji ilorazem różnicowym:

$$\frac{\partial f(x)}{\partial x} = \frac{f(x + \Delta x) - f(x - \Delta x)}{2 \cdot \Delta x}$$

Obliczenia wykonać dla: $\Delta x = 0.1$ oraz $\Delta x = 0.001$. Porównać zbieżność metody dla trzech rozważanych przypadków (jeden z analityczną pochodną oraz dwa z ilorazem różnicowym).

3. Implementacja

3.1. Manualne zlokalizowanie Wszystkich Przedziałów Izolacji

Wyliczamy wartości funkcji $f(x)$ w przedziale $[1.5, 2.4]$ a następnie używając pakietu matplotlib wykonujemy wykres funkcji w tym przedziale - ważnym czynnikiem dla wyników w pozostałej części zadania jest krok z jakim wyliczamy kolejne wartości. Ja przyjąłem $\Delta x = 0.01$.

```
1 // f(x) --> Pod wykres
2 FILE *fp_function_values = fopen("data/fx.txt", "w");
3 fprintf(fp_function_values, "x\tf_x\n");
4 for (double x = 1.5; x < 2.4; x+= 0.01)
5     fprintf(fp_function_values, "%lf\t%lf\n", x, f(x));
6
7 fclose(fp_function_values);
```

Listing 1: Wyliczenie i zapisanie wartości naszej funkcji $f(x)$

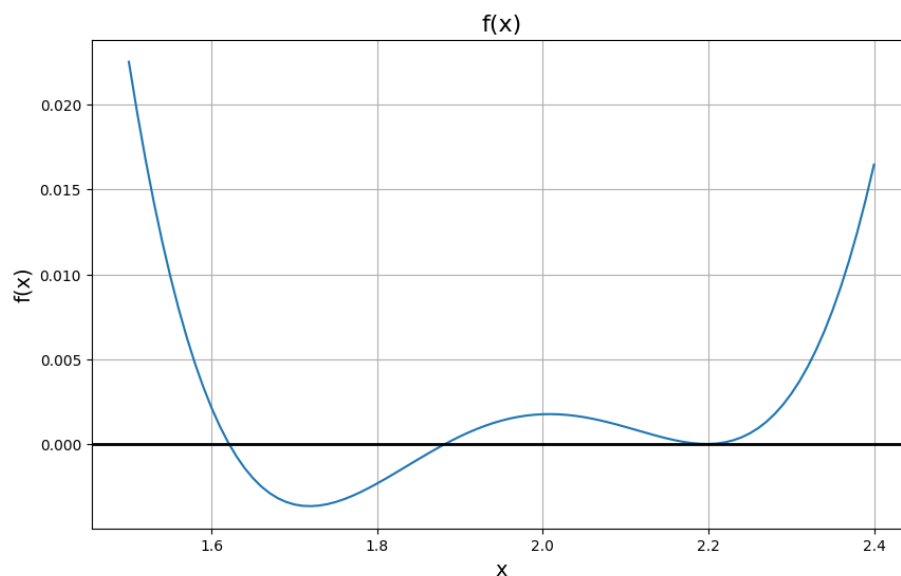


Figure 2: Wykres $f(x)$, który pozwala nam zaobserwować jakie przedziały będą nas interesowały w dalszej części rozwiązania, a także krotności miejsc zerowych (parzysta/nieparzysta krotność)

Wydziwimy 3 miejsca zerowe w tym jedno jest podwójnym (bo wykres się "odbija"). Na podstawie tego wykresy odczytujemy przedziały izolacji:

- [1.6, 1.75]
- [1.8, 2.0]
- [2.1, 2.4]

Przedziały te chcemy ustawić na możliwie najmniejsze i nie mogą zawierać 2 miejsc zerowych - natomiast same rozmiary przedziałów można delikatnie modyfikować aby uzyskać wynik w mniejszą ilość iteracji.

3.2. Znalezienie pierwszego miejsca zerowego

```

1 FILE *fp_solutions_1 = fopen("data/pierwiastek_1.csv", "w");
2 fprintf(fp_solutions_1, "iteracja\tx\tepsilon\n");
3
4 double x1 = 1.6;      double x2 = 2.1;      double x3 = 2.0;
5 Find_Solution(x1, x2, x3, fp_solutions_1);
6
7 fclose(fp_solutions_1);

```

Listing 2: Znalezienie pierwszego miejsca zerowego $f(x)$

```

1 void Find_Solution(double x1, double x2, double x3, FILE* file)
2 {
3
4     int iter_count = 0;
5     double epsilon = abs(x3 - x1);
6     while( abs(x3 - x1) > pow(10, -6)) {
7         double tmp1 = x2;
8         double tmp2 = x3;
9
10        epsilon = abs(x3 - x1);
11        x3 = x1 - f(x1)*(x1-x2)/(f(x1) - f(x2));
12
13        x1 = tmp1;
14        x2 = tmp2;
15        fprintf(file, "%d\t%lf\t%lf\n", iter_count, x3, epsilon);
16        iter_count++;
17    }
18    std::cout<<x3<<std::endl;
19 }

```

Listing 3: Funkcja znajdująca pojedynczy pierwiastek w zadanym przedziale izolacji

Główna pętla while wykonuje się do momentu aż błąd liczony jako $abs(x_3 - x_1)$ nie będzie w granicy akceptowalnego.

W ciele pętli przeliczamy nową wartość błędu (epsilon) a następnie wyliczmy w sposób zgodny wprost z wzorem punkt x_3 reprezentujący nasze przybliżenie rozwiązania

Aktualizujemy punkty graniczne:

- Lewy punkt $x_1 \leftarrow$ Stary prawy punkt brzegowy
- Prawy punkt $x_2 \leftarrow$ Obecna wartość przybliżenia rozwiązania czyli: x_3

Następne wywołania dla kolejnych pierwiastków wyglądają analogicznie. Różnią się tylko i wyłącznie punktami okrślającymi przedziały izolacji.

```

1 /// Drugi Pierwiastek
2 x1 = 1.75;
3 x2 = 2;
4 x3 = 1.55;
5 Find_Solution(x1, x2, x3, fp_solutions_2);
6
7 fclose(fp_solutions_2);

```

Listing 4: Znalezienie drugiego miejsca zerowego $f(x)$

```

1 /// Trzeci Pierwiastek
2 x1 = 1.4;
3 x2 = 1.7;
4 x3 = 1.55;
5 Find_Solution(x1, x2, x3, fp_solutions_3);
6
7 fclose(fp_solutions_3);

```

Listing 5: Znalezienie trzeciego miejsca zerowego $f(x)$

3.3. Metoda zmodyfikowana

Celem wprowadzenia modyfikacji musimy zdefiniować 2 funkcje pomocnicze: Pochodną w punkcie x a także funkcję $u(x)$ którą zastąpimy $f(x)$ w nowej wersji kodu.

```
1  double dfdx( double x ) {
2      const double Delta_x = 0.001;
3
4      return ( f(x + Delta_x) - f(x - Delta_x) )
5              / (2*Delta_x);
6  }
7
8  double u( double x ) {
9      return f(x) / dfdx(x);
10 }
```

Listing 6: Funkcja Pochodnej $\frac{\partial f}{\partial x}$ w postaci ilorazu różnicowego i funkcja zastępcza $u(x)$

```
1  void Find_Solution_Modified(double x1, double x2, double x3, FILE* file)
2  {
3
4      int iter_count = 0;
5      double epsilon = abs(x3 - x1);
6      while( abs(x3 - x1) > pow(10, -6)) {
7          double tmp1 = x2;
8          double tmp2 = x3;
9
10         epsilon = abs(x3 - x1);
11         x3 = x1 - u(x1)*(x1-x2)/(u(x1) - u(x2));
12
13         x1 = tmp1;
14         x2 = tmp2;
15         fprintf(file, "%d\t%lf\t%lf\n", iter_count, x3, epsilon);
16         iter_count++;
17     }
18     std::cout<<x3<<std::endl;
19 }
```

Listing 7: Zmodyfikowana metoda stycznych znajdowania pierwiastków równań nieliniowych

4. Wyniki

Wyniki i wielokrotne próby dopasowania parametrów x_1, x_2, x_3 pokazują że jest to kluczowy element dla poprawnego funkcjonowania tej metody. Niejednokrotnie pozornie poprawny przedział znajduje rozwiązanie obok.

Błędnie dobrane parametry mogą lub ustawienie przełłów zbyt szerokich może skutkować potężną ilością wykonanych iteracji rzędu kilku tysięcy (gdzie raczej należałoby się spodziewać dla funkcji tego typu wielkości rzędu kilku lub kilkunastu iteracji)

Table 1: Parametry początkowe dla metody siecznych

Wersja	Pierwiastek	x_1	x_2	x_3
Podstawowa	1	1.5	1.75	1.65
	2	1.85	2.0	1.9
	3	2.0	2.3	2.4
Zmodyfikowana	1	1.6	1.8	1.7
	2	1.85	2.0	1.9
	3	2.0	2.3	2.4

Jak widać aby uzyskać poprawne wyniki należało zmodyfikować parametry odpowiednio.

Inną ważną rzeczą wartą obserwacji jest fakt że podstawowa metoda znalazła pierwiastek podwójnych - choć zgodnie z założeniami wcale nie musiało się tak stać. Najpewniej zastosowanie drobnej siatki i małego kroku pozwoliło na "nie przskoczenie" za daleko a przez co zgubienie pierwiastka.

Table 2: Wyniki iteracji dla pierwiastka pierwszego

Iteracja	x	Epsilon
0	1.717059	0.15
1	1.505388	0.032941
2	1.567604	0.144612
3	1.685463	0.149455
4	1.596801	0.180076
5	1.646236	0.029198
6	1.634785	0.039227
7	1.62544	0.037983
8	1.617836	0.020796
9	1.620508	0.016949
10	1.621103	0.004932
11	1.620989	0.003267
12	1.621	0.000481
13	1.621	0.000103
14	1.621	0.000011

Table 3: Wyniki iteracji dla drugiego pierwiastka

Iteracja	x	Epsilon
0	1.900409	0.05
1	1.865885	0.099591
2	1.880898	0.034115
3	1.882754	0.019511
4	1.881961	0.016869
5	1.882002	0.001063
6	1.882	0.000753
7	1.882	0.000038
8	1.882	0.000002

Table 4: Wyniki iteracji dla pierwiastka trzeciego

Iteracja	x	Epsilon
0	1.561533	0.4
1	2.278147	0.738467
2	0.821382	0.121853
3	2.476294	0.740151
4	2.279664	0.198147
5	2.517847	1.458282
...
635	2.197777	0.002221
636	2.199968	0.000058
637	2.588963	0.001746
638	2.197719	0.391186
639	2.199966	0.00225
640	2.197719	0.388997

Zgodnie z oczekiwaniami ilość iteracji jest znacząco większa ale i to dzięki dobrze dobranym parametrom udało się zmniejszyć z ok. 2000 do 600 w wariancie podstawowym.

Table 5: Wyniki iteracji dla metody zmodyfikowanej dla pierwszego pierwiastka

Iteracja	x	Epsilon
0	1.545736	0.1
1	1.778703	0.254264
2	1.567343	0.078703
3	1.377314	0.021607
4	1.465784	0.401389
5	1.665703	0.101559
6	1.712605	0.288389
7	1.575178	0.246821
8	1.662505	0.090526
9	1.579382	0.0501
10	1.606473	0.004205
11	1.607711	0.056032
12	1.624759	0.028329
13	1.6223	0.018286
14	1.620645	0.014588
15	1.621036	0.004115
16	1.620997	0.001264
17	1.621	0.000352
18	1.621	0.000036
19	1.621	0.000003

Table 6: Wyniki iteracji dla metody zmodyfikowanej dla drugiego pierwiastka

Iteracja	x	Epsilon
0	1.854513	0.05
1	1.898031	0.145487
2	1.880908	0.001969
3	1.88106	0.026395
4	1.881947	0.016971
5	1.882002	0.001039
6	1.882	0.000942
7	1.882	0.000053
8	1.882	0.000002

Table 7: Wyniki iteracji dla metody zmodyfikowanej dla trzeciego pierwiastka

Iteracja	x	Epsilon
0	2.313519	0.4
1	2.17338	0.013519
2	2.171021	0.22662
3	2.203811	0.142497
4	2.196138	0.030431
5	2.198404	0.025117
...
507	2.198371	0.00006
508	2.198232	0.000238
509	2.198781	0.000173
510	2.197118	0.000409
511	2.1976	0.001115
512	2.197997	0.00118
513	2.198226	0.000879
514	2.197599	0.000625
515	2.198225	0.000398

Dla pierwiastka drugiego stopnia (parzystego stopnia) widoczne jest znaczące przyspieszenie - o minimum 100 iteracji. W przypadkach pierwiastka 1 i 2 to albo się nic nie zmieniło albo minimalnie wydłużył proces iteracji. Wynikiem tego mogą być różne paramtry wejściowe.

5. Wizualizacja Wyników

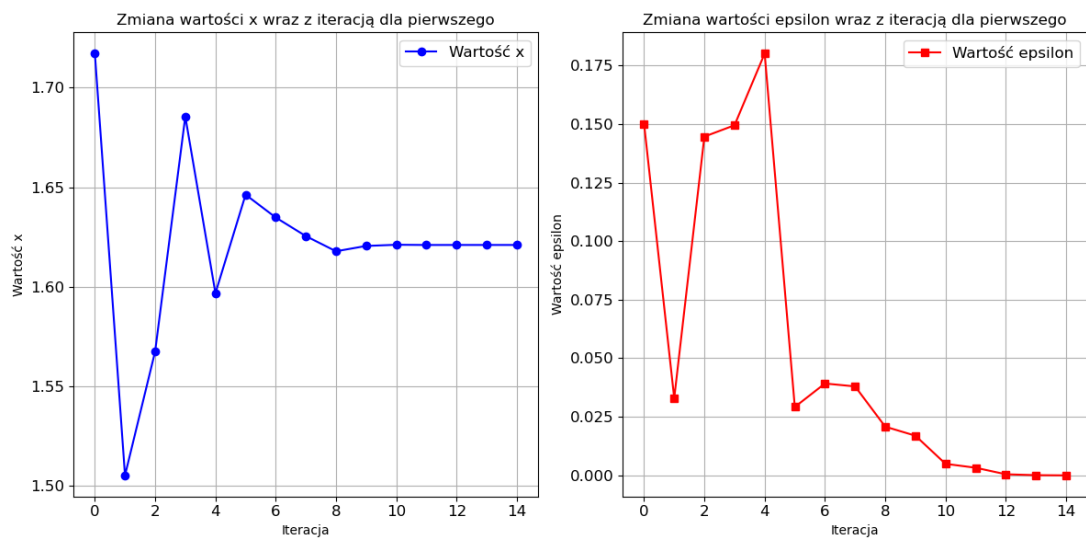


Figure 3: Wykres prezentujący zmianę przybliżenia pierwiastka 1 w zależności od iteracją

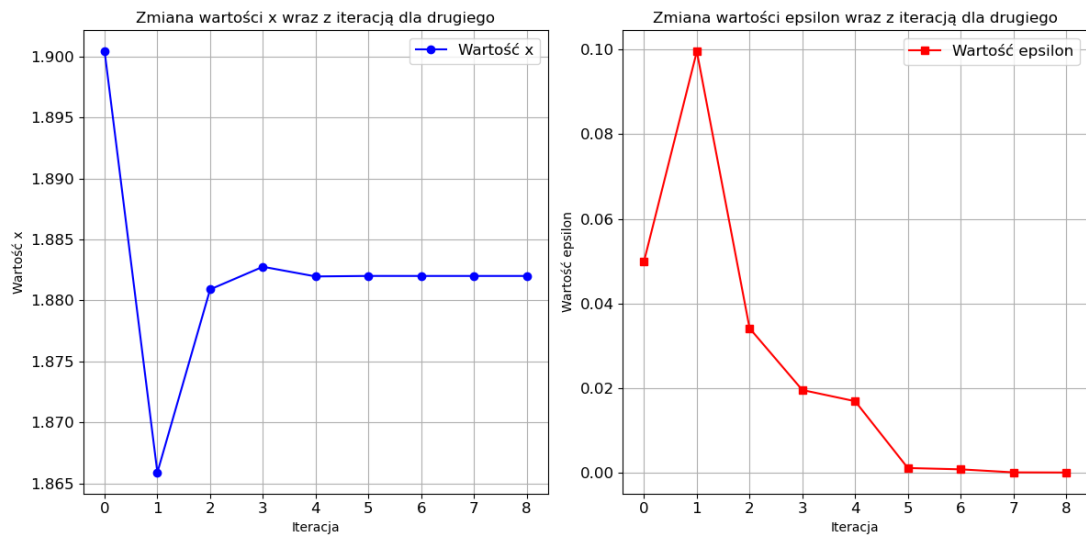
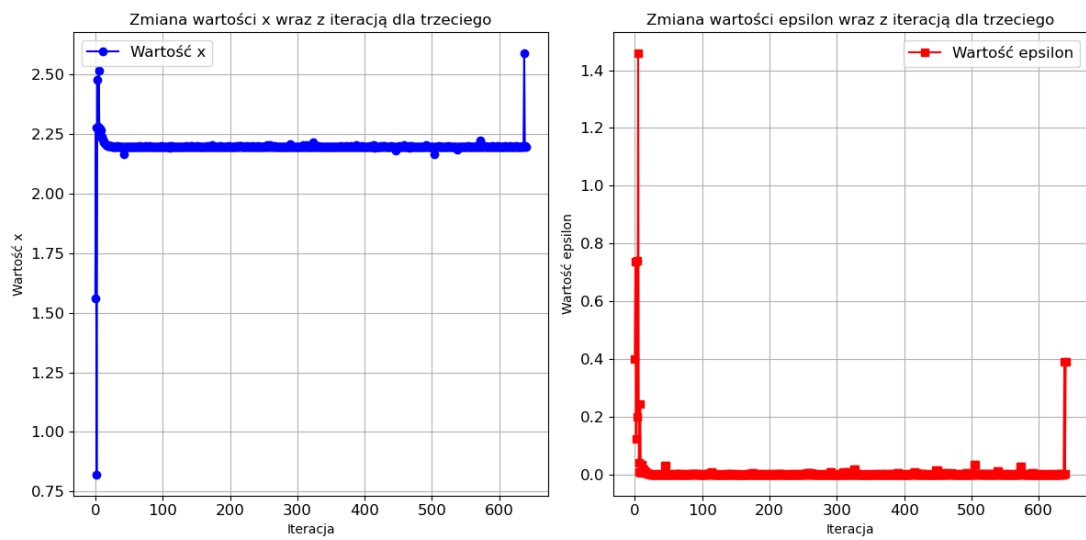


Figure 4: Wykres prezentujący zmianę przybliżenia pierwiastka 2 w zależności od iteracją



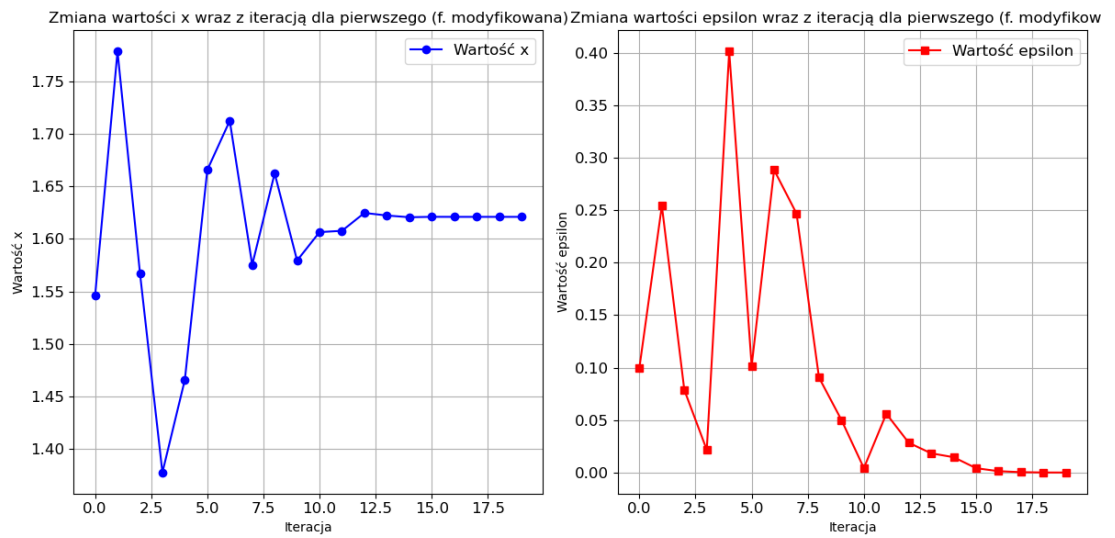


Figure 6: Wykres prezentujący zmianę przybliżenia pierwiastka 1 w zależności od iteracją przy użyciu zmodyfikowanej metody siecznych

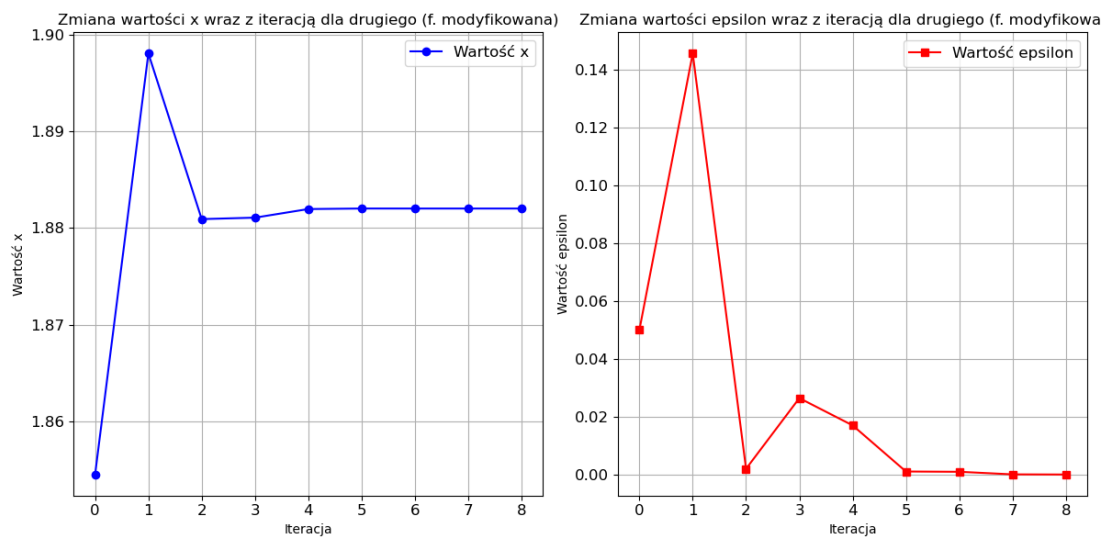


Figure 7: Wykres prezentujący zmianę przybliżenia pierwiastka 2 w zależności od iteracją przy użyciu zmodyfikowanej metody siecznych

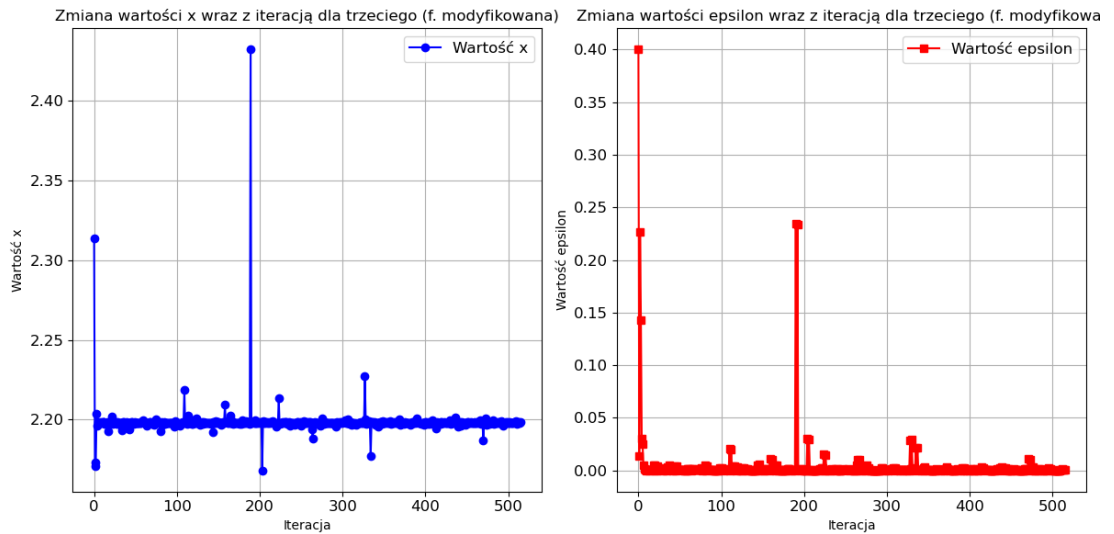


Figure 8: Wykres prezentujący zmianę przybliżenia pierwiastka 3 w zależności od iteracją przy użyciu zmodyfikowanej metody siecznych

Wszystkie wykresy pokazują w jaki sposób zmieniały się x (pierwiastek równania) jak i epsilon. Daje się dostrzec pewne ważne zależności:

- Dla pierwiastków 1 i 2 niepojawiają się oscylacje a wartości szybko dążą do pożądaných wartości. Tak naprawdę można powiedzieć że akceptowalny wynik w przypadku metody nie zmodyfikowanej uzyskaliśmy już po 8 i 5 iteracjach. Poźniej następowało znaczące spowolnienie w zbieganiu do właściwej wartości.
- W przypadku pierwiastka 3 który jest pierwiastkiem podwójnym ilość potrzebnych iteracji jest o rząd wielkości większa. Wpływ może mieć na to zarówno fakt, że podstawowa metoda nie jest przewidziana do tego typu pierwiastków jak również fakt że wynik wszedł w pewnego rodzaju oscylację "względnie" w okół wartości dokładnej. Już od około kilkunastu iteracji wartość pierwiastka przybliżonego przestaje się gwałtownie zmieniać i z wyjątkiem kilku outlierów oscyluje kolejne 590 iteracji. Choć jest to z matematycznego punktu widzenia duże uproszczenie trend jakościowy zdaje się być widoczny.
- W zmodyfikowanych wariantach metody siecznych wartości wyraźniej zbiegają dużo szybciej. Wykres pierwiastka 2 (Fig 7.) idealnie to prezentuje. Już w 3 iteracji wartość pierwiastka przestaje oscylować i tylko stabilizuje się zyskując na dokładności. Od 5 iteracji już prawie wogóle się nie zmienia co potwierdza wykres epsilon od iteracji.
- Zauważalnym jest fakt, że w metodzie zmodyfikowanej jest potrzeba wykonania większej ilości iteracji dla pierwiastka 1 (dla tych parametrów początkowych)

- Wkres pierwiastka 3 (Fig. 8) w wariancie zmodyfikowanym wyraźnie szybciej osiąga mniejsze epsilon a zarazem dokładniejszy wynik, jednak również widoczne są oscylacje jak również (jak poprzednio) po kilku iteracjach wartość przestaje się drastycznie zmieniać. Jest to osiągane już w pierwszych iteracjach. Niepokojące wydają się być częstsze outliersy widoczne ok 200 i 320 iteracji.
- Na uwagę zasługuje fakt, że niezależnie od parametrów i ilości iteracji wynik rozwiązania równania zawsze okazywał się być poprawny.

6. Wnioski

1. Skuteczność metody siecznych: Wyniki eksperymentalne potwierdzają skuteczność metody siecznych w znajdowaniu miejsc zerowych funkcji nieliniowych. Metoda ta działa efektywnie nawet w przypadku braku znajomości pochodnych funkcji, co czyni ją użyteczną w praktyce.
2. Porównanie z metodą Falsi: Porównując metody siecznych i Falsi, zauważamy, że obie są skuteczne, ale mogą różnić się w zależności od charakterystyki funkcji. Metoda siecznych może być bardziej stabilna i mniej podatna na oscylacje w przypadku funkcji o dużej zmienności.
3. Wrażliwość na początkowe przybliżenia: Warto zauważyć, że skuteczność metody siecznych może być wrażliwa na dobranie początkowych przybliżeń miejsc zerowych. Dobór tych punktów może wpłynąć na szybkość zbieżności i stabilność procesu iteracyjnego.
4. Zastosowanie modyfikowanej metody siecznych: Warto rozważyć zastosowanie modyfikowanej wersji metody siecznych, która wykorzystuje iloraz różnicowy do obliczenia kolejnych przybliżeń. Ta modyfikacja może być przydatna szczególnie w przypadku miejsc zerowych o krotności parzystej.
5. Ograniczenia metody: Pomimo skuteczności, metoda siecznych może napotkać ograniczenia w przypadku funkcji o skomplikowanej strukturze lub ekstremalnych warunkach, takich jak ekstremalnie duże lub małe wartości funkcji w okolicy miejsca zerowego.
6. Zalecenia: W celu poprawy efektywności metody siecznych warto eksperymentalnie dobrać początkowe przybliżenia, monitorować zbieżność procesu iteracyjnego oraz stosować modyfikacje w zależności od charakterystyki rozwiązywanego problemu.

Źródła:

Wykres metody False: "https://www.cce.pk.edu.pl/mglowacki/rniel_w01INF.pdf"