Wiktor Bechciński 229840 229840@edu.p.lodz.pl
Kamil Budzyn 229850 229850@edu.p.lodz.pl

Zadanie 1.: Metody wyznaczania miejsca zerowego

1. Cel

Celem zadania było zaimplementowanie i porównanie ze sobą dwóch metod (siecznych oraz falsi) rozwiązywania (znajdowania miejsca zerowego) równań nieliniowych. .

2. Wprowadzenie

Metoda Siecznych - polega na wyliczeniu miejsca przecięcia siecznej poprowadzonej pomiędzy krańcami podanego przedziału z osią OX, następnie punkt ten traktuje jako przybliżone miejsce zerowe i staje się on nowym krańcem przedziału. Czynność ta powtarzana jest do momentu uzyskania zadanej dokładności(uzyskiwanej dzięki wzorowi wybranemu przez użytkownika) lub osiągnięciu maksymalnej liczby iteracji. Algorytm:
-Przez punkty $X_0 = (x_0,f(x_0))$ oraz $X_1 = (x_1,f(x_1))$ przeprowadzana jest prosta(sieczna)
-Wzór(1) na wyliczenie punktu przecięcia z osią OX:

$$x = x_0 - (x_0 - x_1) * f(x_0)/f(x_1) - f(x_0)$$

-Jeśli wartość funkcji w wyliczonym punkcie przecięcia lub wartość bezwględna różnicy pary ostatnich punktów(zależnie od wyboru użytkownika) jest większa niż podany epsilon to wyliczony punkt staje się nowym krańcem przedziału i powtarzane zostaje obliczanie punktu przecięcia
-Gdy wartość funkcji w punkcie będzie mniejsza niż podany epsilon lub osiągnięta zostanie maksymalna ilość iteracji to program kończy swoje działanie i zwraca obliczony punkt jako miejsce zerowe.

Regula Falsi (Metoda Fałszywej Pozycji) -ma bardzo podobne działanie jak metoda siecznych, jednak przed wyliczeniem każdego nowego przecięcia siecznej z osią OX jest sprawdzany warunek, czy funkcja przyjmuje różne znaki na krańcach przedziału. Punkt przecięcia siecznej z osią OX brany jest jako pierwsze przybliżenie miejsca zerowego. Obliczamy go ze wzoru(2):

$$x = x_0 * f(x_1) - x_1 * f(x_0)/f(x_1) - f(x_0)$$

3. Opis implementacji

Do wykonania zadania wykorzystaliśmy język programowania Python w wersji 3.9, skorzystaliśmy z dostępnych bibliotek:
— "matplotlib.pyplot" - umożliwiająca stworzyć wykres dla określonej funkcji wraz z zaznaczonymi punktami wyjściowymi liczonych metod,
— "sys" - w celu zakończenia działania programu,
— "numpy" - wykorzystywana do obliczenia funkcji trygonometrycznych.

Cały program podzieliliśmy na dwie części, gdzie użytkownik:
— w pierwszej części może wyświetlić wykresy zaimplementowanych w programie funkcji nieliniowych, określając nawet dokładność wykresu (gęstość punktów),
— w drugiej części może przystąpić do realizacji porównania metod, określając:
  — z której z dostępnych funkcji nieliniowych chce skorzystać
  — przedział, na którym obie metody będą stosowane
  — maksymalną liczbę iteracji, jako jeden z warunków zatrzymania programu
  — ϵ, użytkownik może wybrać z której metody porównania ϵ chce skorzystać:

$$|x_i - x_{i-1}| < \epsilon$$

$$|f(x)| < \epsilon)$$

Po poprawnym podaniu wszystkich danych wejściowych użytkownik otrzymuje podsumowanie wyników dla obu metod: -wartość znalezionego przez metodę miejsca zerowego i wartości badanej funkcji nieliniowej w tym punkcie oraz przy którym przejściu funkcji liczącej w wskazanej przez użytkownika liczbie iteracji, lub wykorzystując porównanie ϵ, które ma wyższy priorytet niż liczba iteracji, -wartość znalezionego miejsca zerowego i wartości badanej funkcji nieliniowej w wcześniejszej iteracji funkcji liczącej, -wartość bezwzględną różnicy pomiędzy ostatnią parą punktów, -wykres przedstawiający badaną funkcję nieliniową wraz z naniesionymi punktami miejsc zerowych wyliczonych przez obie metody. Aby ułatwić interpretację wykresu zastosowaliśmy kolory punktów, a dokładnie kolor pomarańczowy oraz kolor czerwony, które reprezentują odpowiednio miejsce zerowe wyliczone metodą siecznych oraz miejsce zerowe wyliczone za pomocą regula falsi .

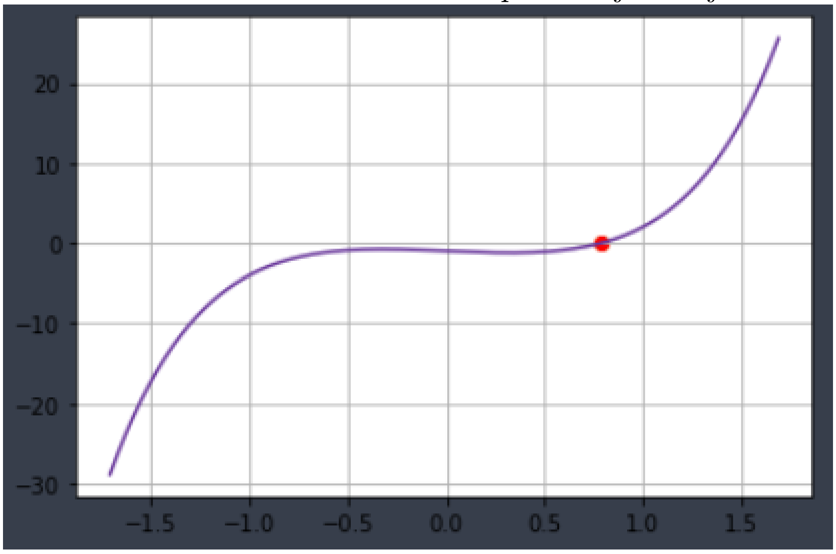
4. Materiały i metody

Sprawdziliśmy która z zaimplementowanych metod potrzebuje mniej iteracji do osiągnięcia zadowalającego wyniku poprzez ustawienie wysokiej(ponad 5000) ilości maksymalnych iteracji i obserwowaliśmy ile wykonań zajmie im osiągnięcie zadanej dokładności. Powtórzyliśmy podany eksperyment dla każdej z zdefiniowanych funkcji oraz dla obu warunków porównania epsilon. Aby poprawnie przeprowadzić podany eksperyment należy w plikach 'metodaSiecznych.py' oraz 'regulaFalsi.py' odkomentować funkcję wyświetlającą interesujące dla użytkownika wartości dla każdej iteracji, która znajduje się zaraz pod realizacją wzorów (1) oraz (2), ponadto użytkownik powinien wybrać drugą z podanych funkcji porównujących do wartości epsilon. Program zakończy swoją pracę wykonując wszystkie iteracje, lub gdy znajdzie dokładne miejsce zerowe.

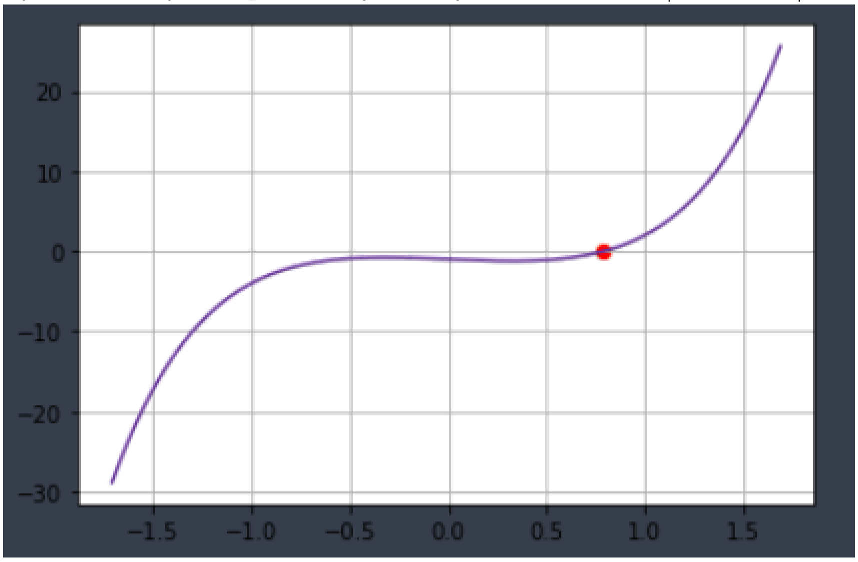
5. Wyniki

x**5 + 3 * x**3 - x - 1	|xi-xi-1|<e	|f(x)|<e
Zakres	[0.5 ; 2]	[0.5 ; 2]
Max iteracji	100	100
Wartosc epsilon	0.00001	0.00001
Wykorzystane iteracje- sieczne	9	9
Wykorzystane iteracje- falsi	56	78
x- sieczne	0.7904836	0.7904836
x - falsi	0.7904333	0.7904821
f(x)- sieczne	0.0000000	0.0000000
f(x) -falsi	-0.0003302	-0.0000092
x rzeczywiste	0.7904836	0.7904836
|Xi-Xi-1| -sieczna	0.0000053	0.0000053
|Xi-Xi-1| -falsi	0.0000088	0.0000002

Tabela 1. Podsumowanie dla pierwszej funkcji.



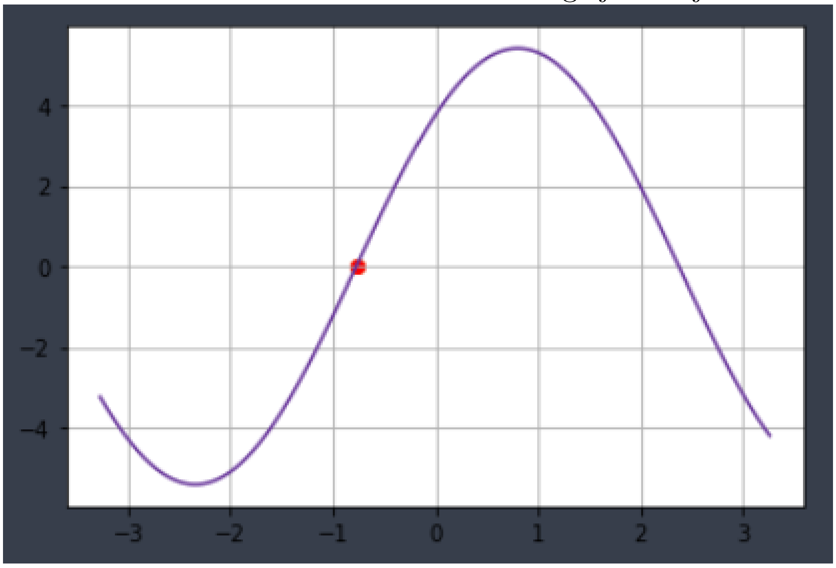
Wykres 1. Wykres pierwszej funkcji dla warunku $|x_i - x_{i-1}| < \epsilon$.



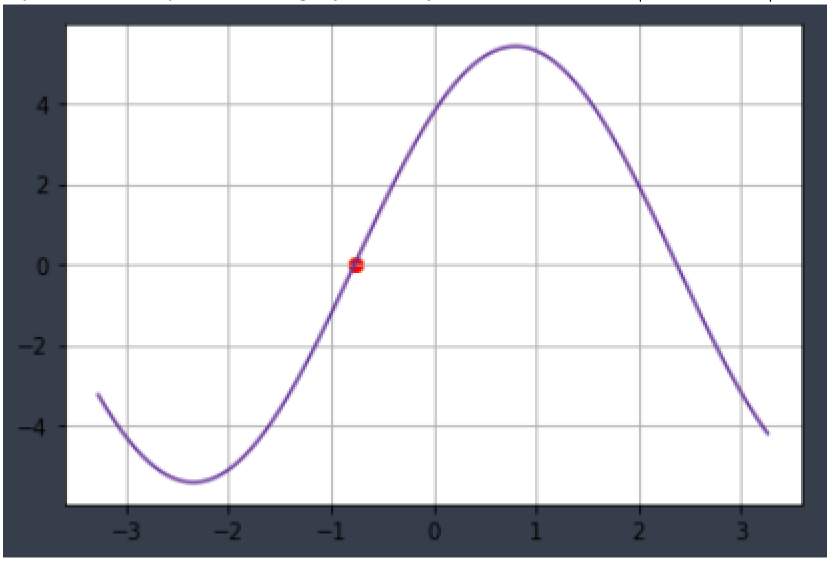
Wykres 2. Wykres pierwszej funkcji dla warunku $|f(x)| < \epsilon$.

7 * cos(x-1) - 2 * sin(x)	|xi-xi-1|<e	|f(x)|<e
Zakres	[-2 ; 1]	[-2 ; 1]
Max iteracji	100	100
Wartosc epsilon	0.00001	0.00001
Wykorzystane iteracje- sieczne	5	4
Wykorzystane iteracje- falsi	5	4
x- sieczne	-0.7712991	-0.7712979
x - falsi	-0.7712991	-0.7712995
f(x)- sieczne	-0.0000000	0.0000065
f(x) -falsi	0.0000000	-0.0000011
x rzeczywiste	-0.7712991	-0.7712991
|Xi-Xi-1| -sieczna	0.0000011	0.0001130
|Xi-Xi-1| -falsi	0.0000002	0.0004223

Tabela 2. Podsumowanie dla drugiej funkcji.



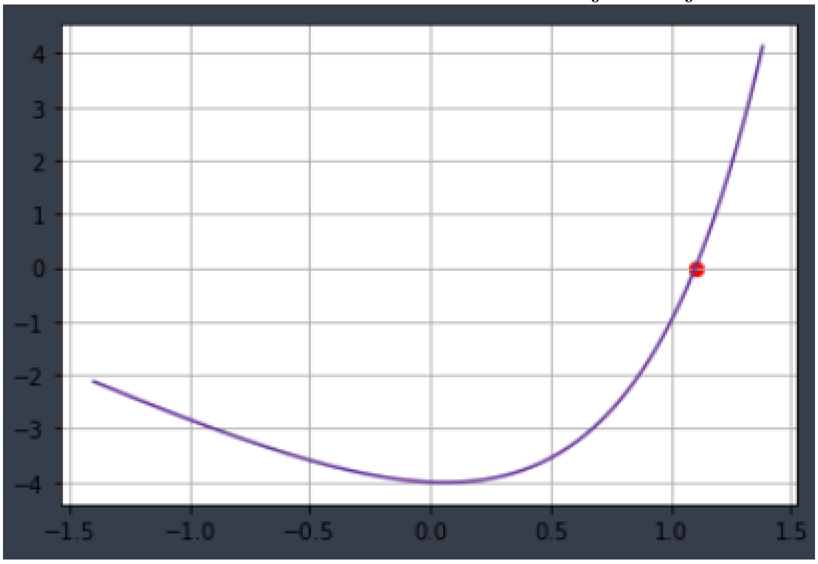
Wykres 3. Wykres drugiej funkcji dla warunku $|x_i - x_{i-1}| < \epsilon$.



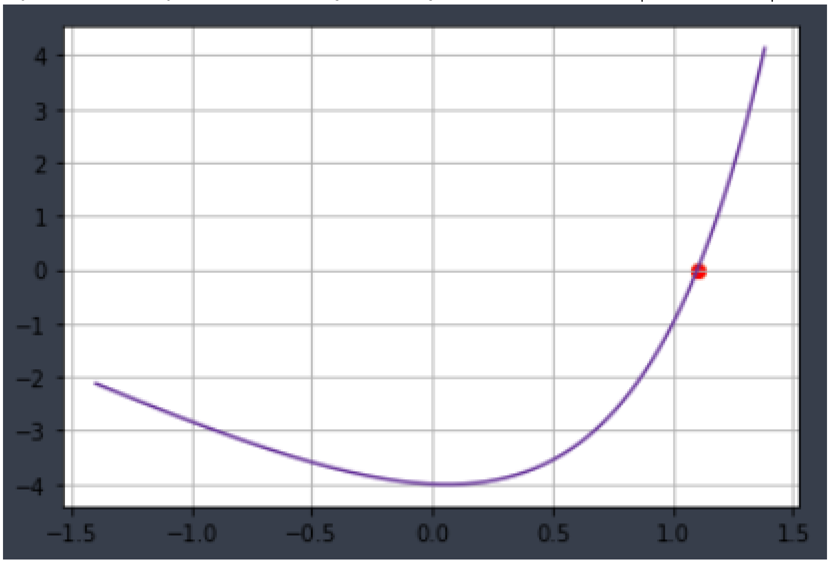
Wykres 4. Wykres drugiej funkcji dla warunku $|f(x)| < \epsilon$.

6**x - 2 * x - 5	|xi-xi-1|<e	|f(x)|<e
Zakres	[0.5 ; 3]	[0.5 ; 3]
Max iteracji	100	100
Wartosc epsilon	0.00001	0.00001
Wykorzystane iteracje- sieczne	10	10
Wykorzystane iteracje- falsi	87	100
x- sieczne	1.1020776	1.1020776
x - falsi	1.1019897	1.1020556
f(x)- sieczne	-0.0000001	-0.0000001
f(x) -falsi	-0.0009591	-0.0002404
x rzeczywiste	1.1020776	1.1020776
|Xi-Xi-1| -sieczna	0.0000079	0.0000079
|Xi-Xi-1| -falsi	0.0000099	0.0000025

Tabela 3. Podsumowanie dla trzeciej funkcji.



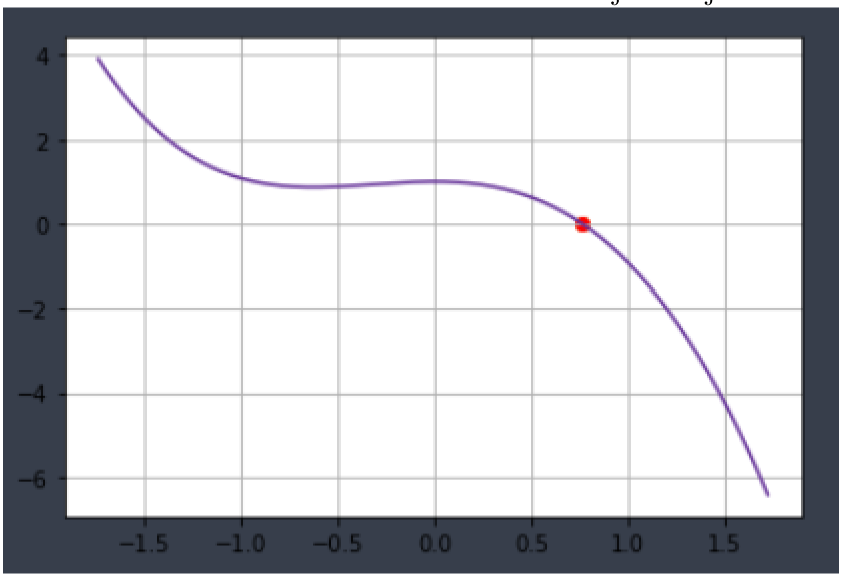
Wykres 5. Wykres trzeciej funkcji dla warunku $|x_i - x_{i-1}| < \epsilon$.



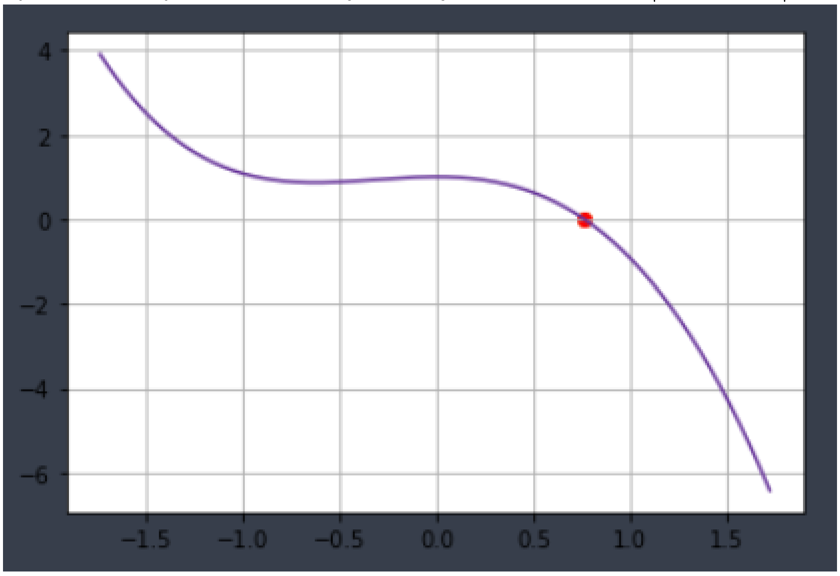
Wykres 6. Wykres trzeciej funkcji dla warunku $|f(x)| < \epsilon$.

2 * cos(x) - 1**x - x**3	|xi-xi-1|<e	|f(x)|<e
Zakres	[0 ; 2]	[0 ; 2]
Max iteracji	100	100
Wartosc epsilon	0.00001	0.00001
Wykorzystane iteracje- sieczne	10	9
Wykorzystane iteracje- falsi	24	27
x- sieczne	0.7634272	0.7634275
x - falsi	0.7634172	0.7634249
f(x)- sieczne	0.0000000	-0.0000010
f(x) -falsi	0.0000313	0.0000070
x rzeczywiste	0.7634272	0.7634272
|Xi-Xi-1| -sieczna	0.0000003	0.0000981
|Xi-Xi-1| -falsi	0.0000065	0.0000014

Tabela 4. Podsumowanie dla czwartej funkcji.



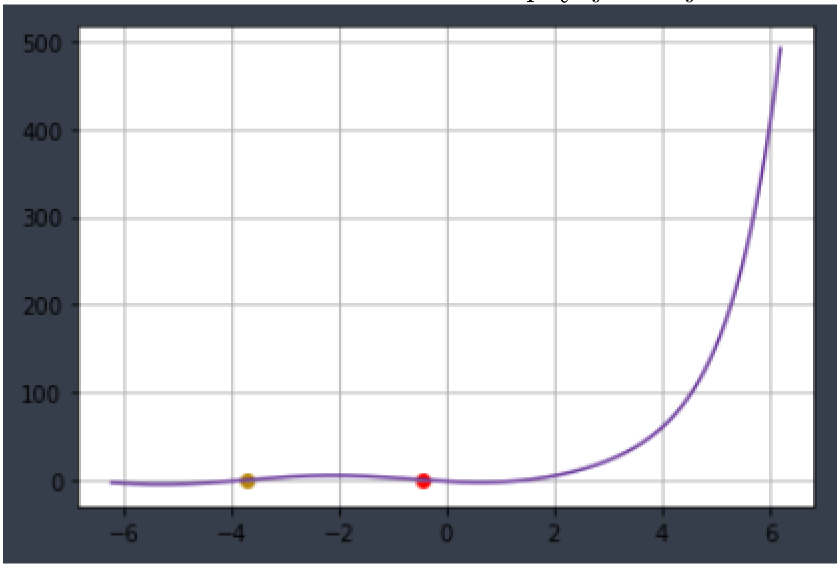
Wykres 7. Wykres czwartej funkcji dla warunku $|x_i - x_{i-1}| < \epsilon$.



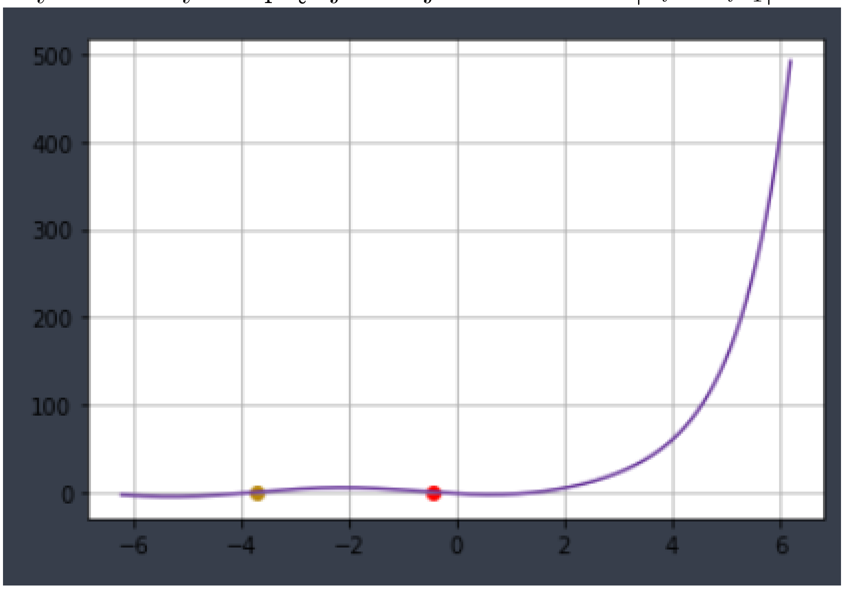
Wykres 8. Wykres czwartej funkcji dla warunku $|f(x)| < \epsilon$.

<table>
<tr><td>exp(x) - 5 * cos(x-1)</td><td>|xi-xi-1|<e</td><td>|f(x)|<e</td></tr>
<tr><td>Zakres</td><td>[-2 ; 1]</td><td>[-2 ; 1]</td></tr>
<tr><td>Max iteracji</td><td>100</td><td>100</td></tr>
<tr><td>Wartosc epsilon</td><td>0.00001</td><td>0.00001</td></tr>
<tr><td>Wykorzystane iteracje- sieczne</td><td>8</td><td>7</td></tr>
<tr><td>Wykorzystane iteracje- falsi</td><td>6</td><td>5</td></tr>
<tr><td>x- sieczne</td><td>-3.7172491</td><td>-3.7172506</td></tr>
<tr><td>x - falsi</td><td>-0.4418733</td><td>-0.4418728</td></tr>
<tr><td>f(x)- sieczne</td><td>-0.0000000</td><td>-0.0000071</td></tr>
<tr><td>f(x) -falsi</td><td>-0.0000000</td><td>-0.0000019</td></tr>
<tr><td>x rzeczywiste</td><td>-0.4418733</td><td>-0.4418733</td></tr>
<tr><td>|Xi-Xi-1| -sieczna</td><td>0.0000014</td><td>0.0015591</td></tr>
<tr><td>|Xi-Xi-1| -falsi</td><td>0.0000004</td><td>0.0000589</td></tr>
</table>

Tabela 5. Podsumowanie dla piątej funkcji.



Wykres 9. Wykres piątej funkcji dla warunku $|x_i - x_{i-1}| < \epsilon$.



Wykres 10. Wykres piątej funkcji dla warunku $|f(x)| < \epsilon$.

6. Dyskusja

Dla pierwszej funkcji(wielomian) w podanym przedziale metoda siecznych okazała się wydajniejsza, udało jej się odnaleźć dokładniej miejsce zerowe oraz dokonała tego w dużo mniejszej ilości iteracji (9 vs 56 dla pierwszego warunku i 9 vs 78 dla drugiego warunku). Dodatkowo obie funkcje osiągneły zadaną dokładność w mniej niż maksymalną ilość iteracji.
Dla drugiej funkcji(trygonometryczna) obie metody uzyskały zadowalający wynik w taką samą ilość iteracji.
Ciekawie robi się przy trzeciej funkcji która jest funkcją wykładniczą, mianowicie metoda siecznych poradziła sobie w obu warunkach w zaledwie 10 iteracji, natomiast regula falsi w przypadku pierwszego warunku uzyskała oczekiwaną wartość dopiero w 87 przejściu, dla drugiego z warunków nie

udało jej się skończyć przed osiągnięciem maksymalnej ilości iteracji.
Czwarta funkcja niczym nie zaskakuje, metoda siecznych jest znacznie szybsza od metody fałszywej pozycji.
Najciekawszą funkcją jest ostatnia, ponieważ dla niej metoda siecznych nie znajduje miejsca zerowego w zadanym przedziale, zamiast tego wychodzi ona poza przedział i znajduje tam miejsce zerowe. Regula Falsi otrzymuje dla danej funkcji miejsce zerowe w mniej iteracji oraz w odpowiednim przedziale. Dzieje się tak ponieważ Regula Falsi sprawdza krańce zakresu w każdej swojej iteracji przez co jest ona wolniejsza ale mamy pewność że nie wyjdzie poza zakres.

7. Wnioski

— Metoda siecznych w większości przypadków jest szybsza względem metody fałszywej pozycji (regulafalsi)
— Regula falsi traktuje przedział jako bariere, której nie może przekroczyć, natomiast metoda siecznych nie traktuje przedziału "poważnie"
— Metoda siecznych nie radzi sobie, jeżeli w zadanym zakresie jest wiecej niż 1 miejsce zerowe
— Regula falsi zwróci tylko jedno miejsce zerowe, nawet jeśli w podanym przedziale jest ich wiecej

Literatura

[1] T. Oetiker, H. Partl, I. Hyna, E. Schlegl. *Nie za krótkie wprowadzenie do systemu LaTeX2e*, 2007, dostępny online. `https://ctan.org/tex-archive/info/lshort/polish/lshort2e.pdf`.
[2] B. Borowska, Wykład "Metody numeryczne i optymalizacja"
[3] A. Romanowicz, Materiały wykładowe "Metody numeryczne"