Raport Zadanie NUM9

Tomasz Dziób

18.01.2024

1 Wstęp techniczny

Poniższy raport dotyczy zadania numerycznego NUM8 z listy 9. Załączone programy o nazwie *metody.cpp*, *num9_a.cpp* i *num9_b.cpp* zostały napisane w języku *C++*, oraz korzystają z bibiotek *GNU Plot*. Przed skompilowaniem programu należy zainstalować powyższe biblioteki.

1.1 Jak uruchomić program?

Razem z załączonymi plikami znajdziemy *Makefile* który służy do uruchomienia programów.

- *make runA* uruchamia program dotyczący podpunktu a
- *make runB* uruchamia program dotyczący podpunktu b

2 Nakreślenie problemu

Szukanie miejsc zerowych funkcji, zwane również rozwiązywaniem równań nieliniowych, to kluczowy problem w matematyce numerycznej i ma szerokie zastosowanie w wielu dziedzinach nauki. Miejsce zerowe funkcji to punkt, w którym wartość funkcji wynosi zero. Innymi słowy, jest to rozwiązanie równania f(x) = 0.

Istnieje wiele metod numerycznych do szukania miejsc zerowych, w tym metoda Regula Falsi, metoda Newtona, metoda Siecznych i metoda Bisekcji. Każda z tych metod jest metodą iteracyjną pokazującą inne podejście do odnajdywania pierwiastków zadanej funkcji.

W zadaniu mieliśmy odnaleźć pierwiastek funkcji $f(x)$ oraz $g(x)$ dla:

a) $f(x) = sin(x) - 0.4$

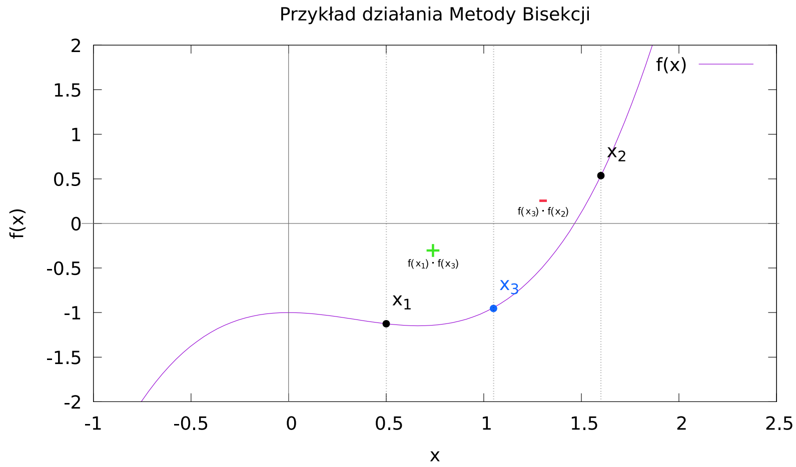
b) $g(x) = f(x)^2 = (sin(x) - 0.4)^2$

3 Użyta metoda

3.1 Metoda Bisekcji

Metoda ta polega na wzięciu dwóch liczb x_1, x_2 między którymi podejrzewamy, że znajduje się nasz szukany pierwiastek.

1. Jako trzeci punkt bierzemy $x_3 = \frac{x_1+x_2}{2}$, czyli środek przedziału.
2. Ustalamy, w którym z przedziałów $[x_1, x_3]$, $[x_3, x_2]$ funkcja zmienia znak $f(x_1) \cdot f(x_2) < 0$, po czym powtarzamy cała procedurę dla tego przedziału.
3. Warunkiem stopu jest $|x_{n+1} - x_n| \leq \epsilon$.

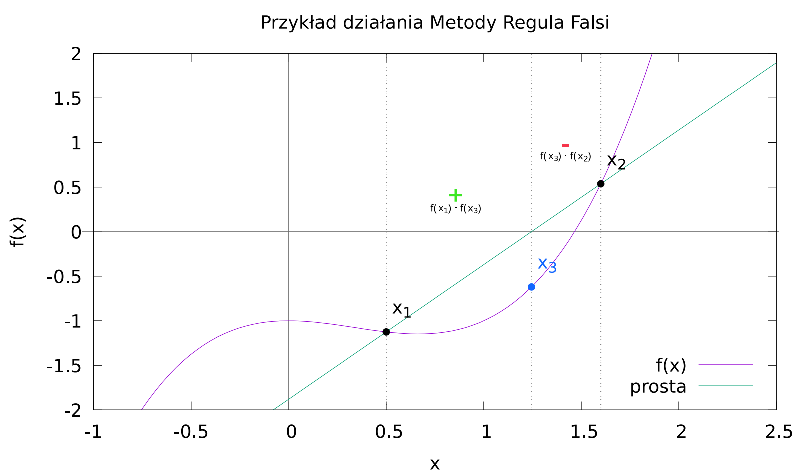


Rysunek 1: Przykład działania Bisekcji

3.2 Metoda *Regula Falsi*

Inaczej nazywana *Metoda Fałszywego Położenia*, polega na wzięciu dwóch liczb x_1, x_2 między którymi podejrzewamy, że znajduje się nasz szukany pierwiastek. Jeżeli znajdziemy dwa punkty, w których znak funkcji jest przeciwny, $f(x_1) \cdot f(x_2) < 0$, jako przyblizenie miejsca zerowego bierzemy punkt przecięcia siecznej przechodzącej przez punkty $(x_1, f(x_1))$, $(x_2, f(x_2))$ z osią OX.

1. Jako trzeci punkt bierzemy $x_3 = \frac{f(x_1) \cdot x_2 - f(x_2) \cdot x_1}{f(x_1) - f(x_2)}$
2. Ustalamy, w którym z przedziałów $[x_1, x_3]$, $[x_3, x_2]$ funkcja zmienia znak $f(x_1) \cdot f(x_2) < 0$, po czym powtarzamy cała procedurę dla tego przedziału.
3. Warunkiem stopu jest $|x_{n+1} - x_n| \leq \epsilon$.

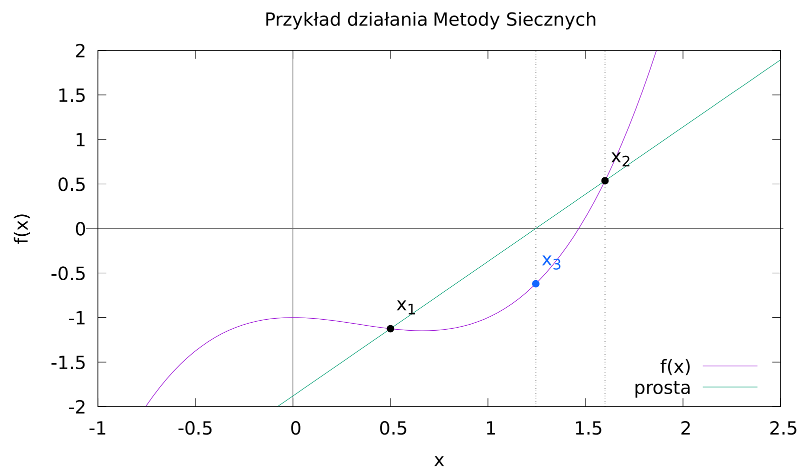


Rysunek 2: Przykład działania metody *Regula Falsi*

3.3 Metoda Siecznych

Bardzo podobna do metody *Regula Falsi* polega na wzięciu dwóch liczb x_1, x_2, dla których $f(x_1) \neq f(x_2)$. Bez względu na znak $f(x_1) \cdot f(x_2)$, jako przybliżenie miejsca zerowego bierzemy punkt przecięcia siecznej przechodzącej przez punkty $(x_1, f(x_1))$, $(x_2, f(x_2))$ z osią OX.

1. Jako trzeci punkt bierzemy $x_3 = \frac{f(x_1) \cdot x_2 - f(x_2) \cdot x_1}{f(x_1) - f(x_2)}$
2. Nie patrzymy na znaki przedziałów, zawsze bierzemy dwa ostatnie punkty $[x_3, x_2]$, po czym powtarzamy cała procedurę dla tego przedziału.
3. Warunkiem stopu jest $|x_{n+1} - x_n| \leq \epsilon$.



Rysunek 3: Przykład działania metody Siecznych

Metoda siecznych moze być zbieżna szybciej niż metoda *Regula Falsi*, ale w odróżnieniu od *Regula Falsi* i metody Bisekcji, w niektórych przypadkach zawodzi(nie jest zbieżna do miejsca zerowego).

3.4 Metoda Newtona

Metoda ta wymaga znania pochodnej funkcji f aby znaleźć szukane miejsce zerowe. Wybieramy losowy punkt x_1 oraz wyliczamy $x_2 = x_1 - \frac{f(x_1)}{f'(x_1)}$. Sprawdzamy warunek stopu oraz jeśli nie osignęliśmy jeszcze zadanej dokładności to do x_1 przypisujemy wartość x_2.

Jest to metoda której użyjemy w specyficznych przypadkach ze względu wymaganą znajomość pochodnej zadanej funkcji, która nie zawsze jest prosta w obliczeniu.

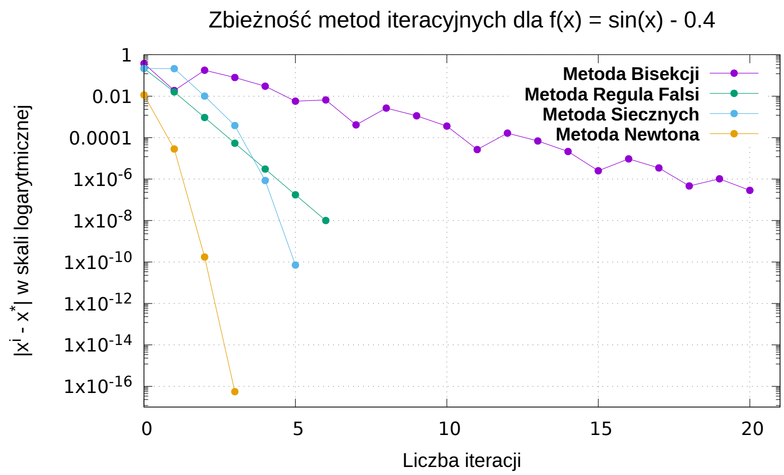
1. Wyznaczamy pochodną funkcji f oraz wyliczamy $f'(x_1)$
2. Jako drugi punkt bierzemy $x_2 = x_1 - \frac{f(x_1)}{f'(x_1)}$
3. Przypisujemy do x_1 wartość x_2 i powtarzamy
4. Warunkiem stopu jest $|x_{n+1} - x_n| \leq \epsilon$.

Metoda Newtona może byc rozbieżna, zawodzi ona w punktach, w których pochodna znika!

4 Uzyskany wynik

a) Wynik wykonania komendy *make runA*:

```
Metoda Bisekcji: 0.411517
Metoda Regula Falsi: 0.411517
Metoda Siecznych: 0.411517
Metoda Newtona: 0.411517
```



Rysunek 4: Wykres dla podpunktu a)

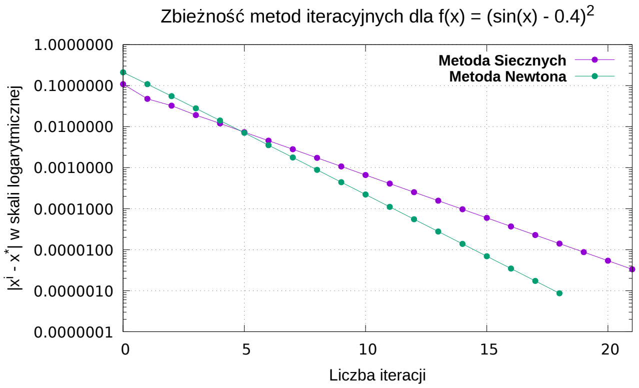
Jak widzimy dla wszystkich metod udało się osiągnąć zadaną dokładność $\epsilon = 10^{-6}$, 3 z nich w podobnym czasie ok. 5 iteracji, jednak wyraźnie odstającą funkcją jest *Metoda Bisekcji* która potrzebowała aż 20 iteracji.

To jest typowe dla Metody Bisekcji, która jest metodą stabilną, ale wolno zbieżną. W przeciwieństwie do niej, *Metoda Newtona*, *Siecznych* i *Regula Falsi* są zazwyczaj szybsze, ale mogą mieć problemy ze stabilnością w niektórych przypadkach.

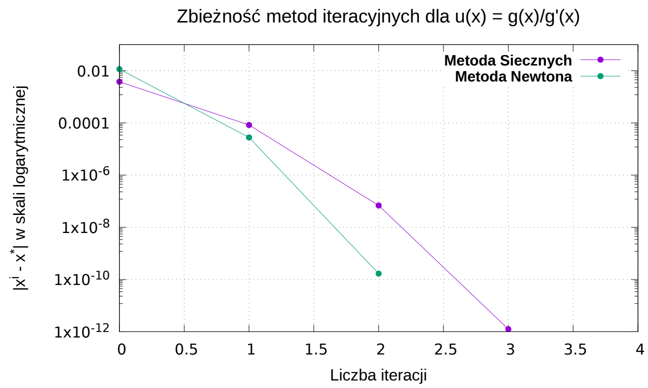
Jesteśmy również wstanie zauważyć mniejsze wzrosty i spadki dokładności które są normalnym zachowaniem nie wpływającym na ogólną liniową naturę tej metody.

b) Wynik wykonania komendy *make runB*:

```
Dla g(x):
Metoda Siecznych: 0.411518
Metoda Newtona: 0.411516
Dla u(x):
Metoda Siecznych: 0.411517
Metoda Newtona: 0.411517
```



Rysunek 5: Wykres dla dla funkcji g(x)



Rysunek 6: Wykres dla funkcji u(x)

Dla funkcji $g(x)$ zastosowanie *metod Bisekcji i Falsi* zakonczyło się niepowodzeniem ze względu na ich kryterium. Wymagają one dwóch punktów w których funkcja przyjmuje ujemne znaki, a $g(x)$ przyjmuje wyłącznie wartości nieujemne. Dwie metody które zostały rozwiązały ten problem to *metod Siecznych i Newtona*, jednak poradziły sobie one z tym w ok. 20 iteracji, wynika to z dwukrotności pierwiastka.

Zastosowanie funkcji $u(x) = \frac{g(x)}{g'(x)}$ poprawiło liczbę potrzebnych iteracji, ponieważ przekształciło oryginalną funkcję $g(x)$ na formę, która jest bardziej przyjazna dla metod takich jak Newtona i Siecznych. Dzielenie $g(x)$ przez jej pochodną sprawia, że nowa funkcja $u(x)$ ma tylko pierwiastki jednokrotne, dzięki czemu metody iteracyjne zbiegają szybciej. Jest to szczególnie korzystne przy funkcjach takich jak $g(x)$, które przyjmują tylko wartości nieujemne i dlatego nie mogą być rozwiązane za pomocą metod takich jak Bisekcja czy Falsi, które wymagają, aby funkcja przyjmowała zarówno wartości dodatnie, jak i ujemne. Po zastosowaniu funkcji $u(x)$, obie metody zbiegły znacznie szybciej, w mniej niż 4 iteracjach. To pokazuje, jak efektywne może być zastosowanie funkcji $u(x)$ w przyspieszaniu zbieżności metod iteracyjnych.

5 Podsumowanie

Ogólnie rzecz biorąc, wyniki te pokazują, że wybór odpowiedniej metody do rozwiązania danego problemu zależy od specyficznych właściwości funkcji, takich jak jej znaki, pierwiastki i znajomość pochodnej. Przekształcenie funkcji na bardziej "przyjazną" formę, taką jak $u(x)$, może znacznie przyspieszyć zbieżność metod iteracyjnych i umożliwić rozwiązanie problemów, które są trudne dla niektórych metod.

Metoda Bisekcji była najwolniejsza, ale najbardziej stabilna. *Metoda Newtona*, *Siecznych* i *Regula Falsi* były szybsze, ale mogą mieć problemy ze stabilnością. Zastosowanie funkcji $u(x)$ przyspieszyło zbieżność Metody Newtona i Siecznych. Wybór odpowiedniej metody zależy od specyficznych właściwości funkcji.