Savchenko Yehor, Metody Numeryczne, zadanie NUM 9

**Instrukcja do programy:**

1. Zainstalować python: [**https://www.python.org/downloads/**](https://www.python.org/downloads/)
2. Zainstalować biblioteki NumPy: **pip install numpy**
3. Wpisać do terminalu: **python “FileName”**

**Wstęp:**

Musiałem znaleźć numerycznie pierwiastek **x­­­\***równań f(x) = 0 i g(x) = 0 dla

1. f(x) = sin(x) – 0.37,
2. g(x) = f(x)2 = (sin(x) – 0.37)2,

Miałem przedział x [0, ] z dokładnością 10-6 metodami:

1. bisekcji (połowienia, równego podziału),
2. Falsi,
3. Siecznych,
4. Newtona (stycznych).

**Bisekcji**

Żeby skorzystać z metody bisekcji w przedziale [a, b] funkcja musi spełniać poniższe warunki:

1. Funkcja f(x) **określona**.
2. Funkcja f(x) **ciągła.**
3. Funkcja f(x) na krańcach przedziału [a, b] przyjmuje **różne znaki**.

Jeżeli funkcja spełnia powyższe warunki, to w przedziale [a, b] istnieje pierwiastek i możemy go wyszukać algorytmem **bisekcji**:

W każdym przybliżeniu algorytm wyznacza środek x0 przedziału [a, b] .

Dalej sprawdzamy, czy odległość x0­ od krańców przedziału jest mniejsza od założonej dokładności. Jeśli tak, to zwracamy x0.

Jeśli nie x0 dzielimy przedział [a, b] na dwie równe połowy: [a, x0] i [x0, b]. Za nowy przedział [a, b] przyjmuję tę połówkę, w której funkcja zmienia znak na krańcach i kontynuujemy wyznaczanie pierwiastka funkcji.

**Falsi**

Żeby skorzystać z metody falsi w przedziale [a, b] funkcja musi spełniać poniższe

warunki:

1. Funkcja f(x) **określona**.
2. Funkcja f(x) **ciągła.**
3. Funkcja f(x) na krańcach przedziału [a, b] przyjmuje **różne znaki**.

Falsi jest bardzo podobna do metody bisekcji. Metoda ta wykorzystuje interpolację liniową funkcji, której zero jest poszukiwane. Prosta ta przechodzi przez punkty graniczne obszaru poszukiwań.

Wzór rekurencyjny:

Nowy obszar poszukiwań jest wyznaczany tak jak było w metodzie bisekcji.

**Siecznych**

Żeby skorzystać z metody siecznych w przedziale [a, b] funkcja musi spełniać poniższe

warunki:

1. Funkcja f(x) **określona**.
2. Funkcja f(x) **ciągła.**
3. Funkcja f(x) na krańcach przedziału [a, b] przyjmuje **różne znaki**.
4. Dodatkowo w przedziale [a, b] pierwsza pochodna f '(x) jest różna od zera

W falsi sprawdzamy, aby funkcja f(x) zawsze miała różne znaki na krańcach przedziału. Bo różne znaki gwarantują nam istnienie pierwiastka w tym przedziale. Ale w metodzie siecznych taki wymóg nie jest konieczny.

W celu obliczenia przybliżenia xi+1 korzystamy z dwóch wcześniej wyznaczonych punktów: xi i xi-1. Wzór określający ciąg przybliżeń jest następujący:

**Newtona**

Żeby skorzystać z metody newtona w przedziale [a, b] funkcja musi spełniać poniższe

warunki:

1. Funkcja f(x) **określona**.
2. Funkcja f(x) **ciągła.**
3. Funkcja f(x) na krańcach przedziału [a, b] przyjmuje **różne znaki**.
4. Dodatkowo w przedziale [a, b] pierwsza pochodna f '(x) jest **różna od zera.**

Zaczynamy obliczenia od punktu x­0 , który jest blisko poszukiwanego pierwiastka funkcji.

W przedziale [x0, x\*], gdzie x\* - docelowy pierwiastek pierwsza pochodna funkcji musi być niezerową. Pożądane, żeby druga pochodna w punkcie x0 miała ten sam znak co i funkcja.

Bo w innym przepadku metoda Newtona ucieknie od punktu pierwiastka zamiast tego, żeby zbliżać się do niego.

Wyznaczamy punkt x0 wykorzystując ostatnio wyliczony. Liczymy dopóki nie zbliżymy się do pierwiastka funkcji używając precison, który wyznamy(różnica pomiędzy dwoma kolejno wyznaczonymi pierwiastkami będzie dostatecznie mała).

Wzór na metodę newtona:

**Wyniki:**

**Bisekcji**:

f(x) = **0.379009**35178652406 | iteracji = 21

g(x) = Funkcja nie spełnia założeń | iteracji = 0

**Falsi**:

f(x) = **0.379009**09123943294 | iteracji = 6

(x) = Funkcja nie spełnia założeń | iteracji = 0

**Siecznych**:

f(x) = **0.379009**02069595077 | iteracji = 6

g(x) = **0.379007**7341527496 | iteracji = 28

u(x) = **0.379009**02069390163 | iteracji = 5

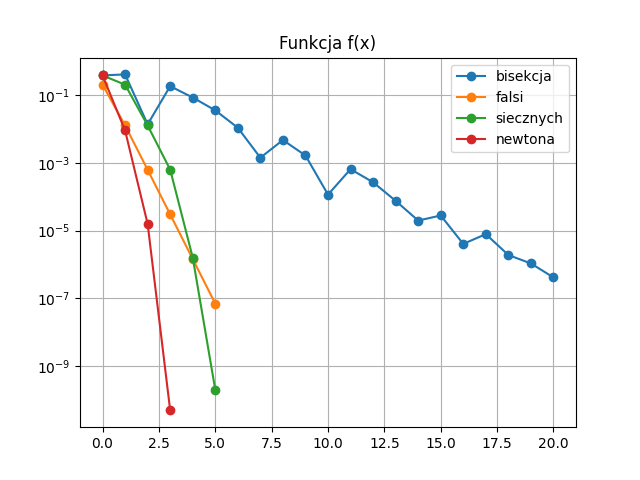
**Newtona**:

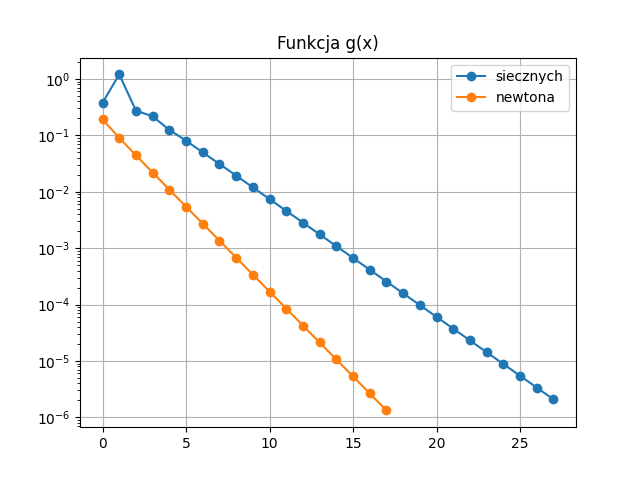
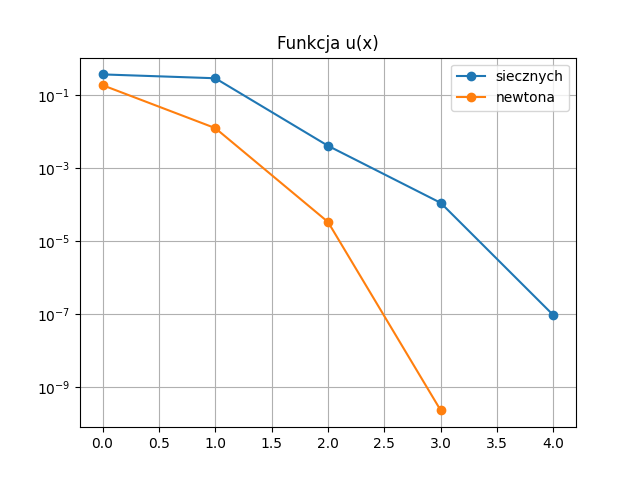
f(x) = **0.379009**0206959508 | iteracji = 4

g(x) = **0.379009**684416199 | iteracji = 18

g(x) = **0.379009**0206959508 | iteracji = 4

**Dokładna wartość** = 0206959508

****

****

**Przedyskutowanie wyników:**

Z wyników widać, że metoda **Bisekcji** potrzebuje najwięcej iteracji, żeby otrzymać wyniki. Metoda **Falsi** jest w większości przypadków szybciej zbieżna od metody bisekcji. No i metody **Siecznych** i **Newtona** potrzebują mniej iteracji, żeby zbiegnąć. Także widać, że funkcja **g(x)** nie spełnia warunki metody **Bisekcji** i **Falsi**, które były podane wyżej. Usprawnienie zadania przez funkcjępotrzebuje mniej iteracji niż funkcja **g(x)** w tych metodach.