

# Program w języku Python realizujący metodę Newtona (stycznych)

Bartosz Beksa, Paweł Bukowski, Tomasz Domurad, Michalina Całus

kwiecień 2021

## 1 Teoretyczny opis metody numerycznej

Problem znalezienia miejsca zerowego funkcji  $f : [a, b] \rightarrow \mathbb{R}$  zamienimy na wyznaczenie miejsca zerowego stycznej do wykresu tej funkcji. Niech  $x_0 \in [a, b]$  będzie ustalonym startowym przybliżeniem miejsca zerowego  $f$ .

Za kolejne przybliżenie  $x_1$  przyjmujemy miejsce zerowe stycznej w punkcie  $(x_0, f(x_0))$  w poniższym równaniu:

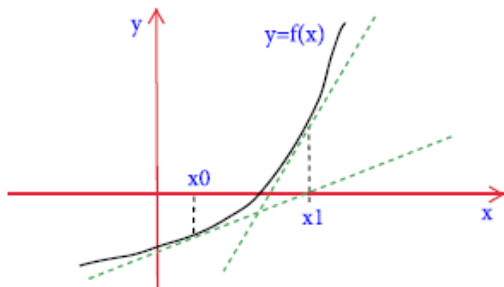
$$y = f'(x_0)(x - x_0) + f(x_0)$$

wstawiamy  $y = 0$  i otrzymujemy:

$$0 = f'(x_0)(x - x_0) + f(x_0) \Rightarrow x_1 := x_0 - \frac{f(x_0)}{f'(x_0)}$$

Otrzymamy tak iteracje nazywane metodą Newtona lub stycznych.

$$x_{k+1} = x_k - \frac{f(x_k)}{f'(x_k)}, k \geq 0$$



Rysunek 1: Metoda stycznych na wykresie

### Błąd przybliżenia

Dzięki kolejnym iteracjom uzyskujemy coraz dokładniejsze przybliżenie miejsca zerowego. Obliczenia można kontynuować do czasu gdy wynik będzie satysfakcjonujący. Błąd k-tego przybliżenia można oszacować poprzez nierówność:

$$|x^* - x_k| \leq \frac{f(x_k)}{m}$$

$$|x^* - x_k| \leq \frac{M}{2m}(x^* - x_{k-1})^2$$

gdzie stałe wylicza się ze wzorów:

$$m = \min_{x \in [a, b]} |f'(x)|$$

$$M = \max_{x \in [a, b]} |f''(x)|$$

## 2 Przykład ilustrujący metodę

Metodą Newtona wyznaczymy dodatni pierwiastek z równania  $x^2 = 2$ :

$$f'(x) = 2x$$

$$f''(x) = 2$$

za  $x_0$  uznamy 2

z metody Newtona wiemy, że  $x_1 = x_0 - \frac{f(x_0)}{f'(x_0)}$  a więc:

$$x_1 = 2 - \frac{f(x_2)}{f'(x_2)} = 2 - \frac{2}{4} = 1,5$$

$$x_2 = 1,5 - \frac{0,25}{3} \approx 1,4166$$

$$x_3 \approx 1,4142$$

Takie przybliżenie miejsca zerowego jest już dla nas wystarczająco dobre.

### 3 Opis implementacji algorytmu realizującego metodę

Program pobiera od użytkownika wzór funkcji, ustalone startowe przybliżone miejsce zerowe funkcji oraz ilość iteracji jakie ma wykonać żeby obliczyć miejsce zerowe funkcji, a następnie zwraca ułamek, który jest przybliżeniem miejsca zerowego podanej funkcji.

Program wykorzystuje funkcje pomocnicze do obliczeń:

- functionvalue - oblicza wartość funkcji dla podanej wartości
- derivativevalue - oblicza wartość pochodnej funkcji dla podanej wartości
- count - liczy przybliżenie miejsca zerowego z dokładnością do n iteracji

Metoda Newtona (stycznych)

**Instrukcja**

Jako zmiennej należy używać 'x'.  
 Funkcja powinna być zapisana jak wyrażenie matematyczne w pythonie  
 np.  $x^2$  powinno być zapisane w postaci  $x**2$   
 $2x$  powinno być zapisane w postaci  $2*x$   
 Zmienna 'n' musi być większa bądź równa 1.  
 $x_0$  należy podać jako liczbę zmiennoprzecinkową

**Podaj dane**

f(x) =

$x_0$  =

n =

Oblicz

Rysunek 2: prezentacja działania programu 1

Metoda Newtona (stycznych)

**Instrukcja**

Jako zmiennej należy używać 'x'.  
 Funkcja powinna być zapisana jak wyrażenie matematyczne w pythonie  
 np.  $x^2$  powinno być zapisane w postaci  $x**2$   
 $2x$  powinno być zapisane w postaci  $2*x$   
 Zmienna 'n' musi być większa bądź równa 1.  
 $x_0$  należy podać jako liczbę zmiennoprzecinkową

**Podaj dane**

f(x) =

$x_0$  =

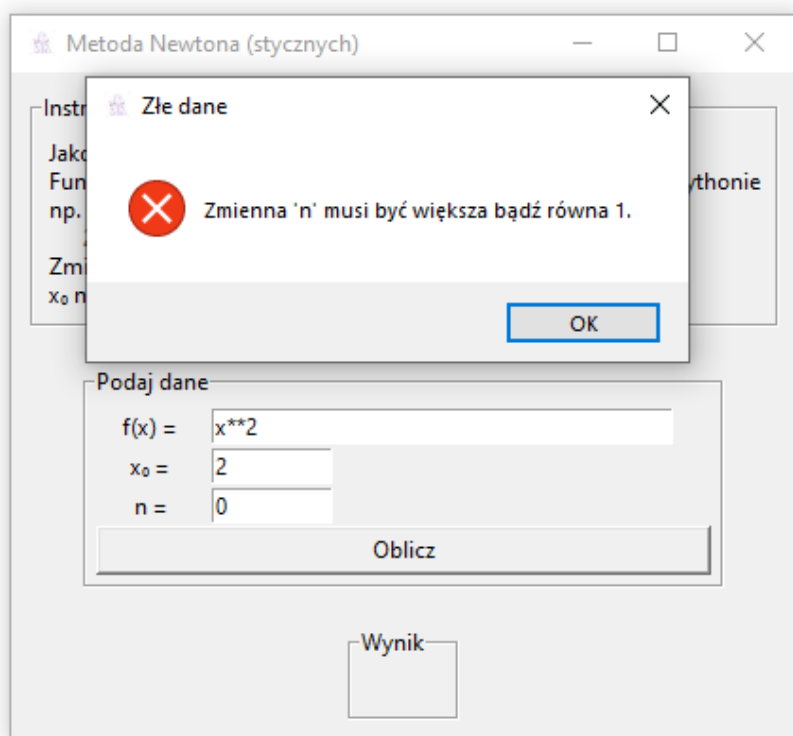
n =

Oblicz

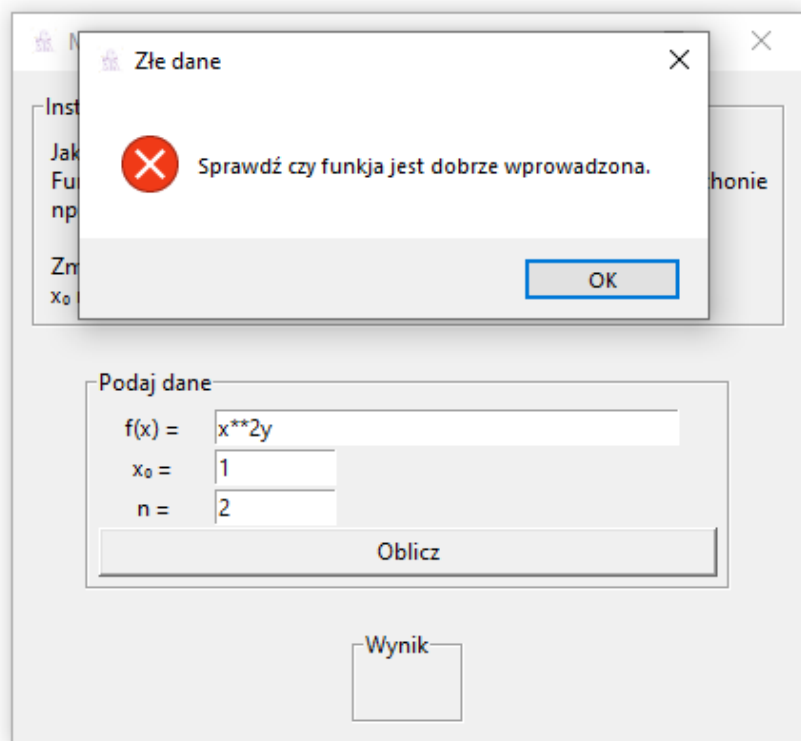
**Wynik**

0.50000

Rysunek 3: prezentacja działania programu 2



Rysunek 4: prezentacja działania programu 3



Rysunek 5: prezentacja działania programu 4

## 4 Kod programu

Kod załączony w archiwum razem z dokumentacją.

## 5 Dokładniejszy opis programu

Program używa wbudowanej w pythona biblioteki tkinter używanej do stworzenia interfejsu graficznego.

Program po uruchomieniu wyświetla instrukcję wprowadzania danych i prosi o podanie wzoru funkcji,  $x_0$  od którego zacznie przybliżanie i liczbę  $n$ , czyli ilość iteracji które ma wykonać. Zwraca przybliżone miejsce zerowe w postaci liczby zmiennoprzecinkowej.

Zabespieczeniem przed wprowadzeniem niepoprawnych danych jest m.in.

komunikat: "Zmienna 'n' musi być większa bądź równa 1." lub

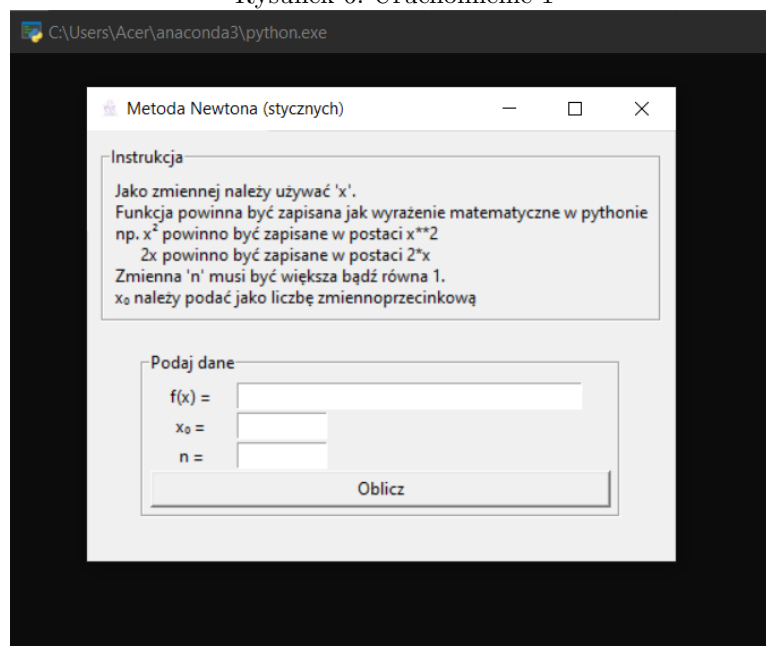
komunikat: "Sprawdź czy funkcja jest dobrze wprowadzona."

## 6 Uruchomienie programu

Do uruchomienia programu potrzebny będzie zainstalowany Python najlepiej w wersji 3.8.0. Należy umieścić wszystkie pliki w jednym miejscu i uruchomić app.pyw przy pomocy pythona.

__pycache__	07.05.2021 17:51	Folder plików	
app.pyw	07.05.2021 16:08	Plik PYW	3 KB
functions.py	07.05.2021 03:11	JetBrains PyCharm ...	1 KB
grafika.ico	25.03.2021 19:04	Ikona	120 KB

Rysunek 6: Uruchomienie 1



Rysunek 7: Uruchomienie 2