

# **Rozwiązywanie równań nieliniowych - metoda Newtona i siecznych**

Dariusz Borkowski

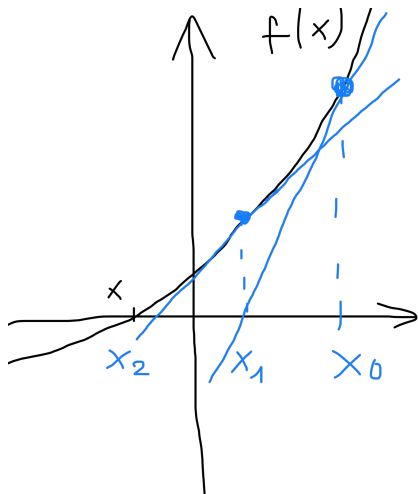
# Metoda Newtona

Niech  $f$  będzie funkcją, której zera musimy wyznaczyć. Niech  $x$  będzie takim zerem, a  $p$  jego przybliżeniem. Jeżeli  $f''$  istnieje to na mocy twierdzenia Taylora

$$0 = f(x) = f(p + h) = f(p) + hf'(p) + O(h^2),$$

gdzie  $h = x - p$ . Jeśli  $h$  jest małe to jest rozsądne pominięcie składnika  $O(h^2)$  i rozwiązanie otrzymanego równania względem  $h$ . Daje to  $h = -f(p)/f'(p)$ . Jeśli  $p$  jest przybliżeniem  $x$ , to  $p - f'(p)/f(p)$  powinno być lepszym przybliżeniem tego zera. Rozumowanie to możemy powtórzyć przyjmując  $x := p - f'(p)/f(p)$ . Dlatego metoda Newtona polega na rekurencyjnym stosowaniu wzoru

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



dla  $n \geq 0$  i pewnego punktu startowego  $x_0$ .

Metoda Newtona jest szybsza od metody bisekcji, gdyż jej zbieżność jest kwadratowa, a nie liniowa. Gdy tylko przybliżenia tworzone metodą Newtona są dostatecznie bliskie pierwiastka, staje się tak szybko zbieżna, że zaledwie kilka dodatkowych przybliżeń daje już maksymalną dokładność. Niestety metoda Newtona nie zawsze jest zbieżna. Wiemy tylko, że

**Twierdzenie 1.** *Istnieje otoczenie pierwiastka funkcji, takie że metoda Newtona startując z tego otoczenia zbiega do tego pierwiastka.*

natomiast nie znamy tego otoczenia. Dlatego często używa się jej w kombinacji z jakąś wolniejszą metodą, ale globalnie zbieżną (np. bisekcji).

**Zadanie 1.** Wykonaj dwie iteracje metody Newtona dla funkcji  $f(x) = x^2 + 1$  i punktu początkowego

a)  $x_0 = 1 + i$ ,

b)  $x_0 = 1$ ,

c)  $x_0 = 2i - 1$ .

**Zadanie 2.** Znajdź efektywny sposób obliczania pierwiastka 2-tego stopnia z liczby  $A$  tzn.  $\sqrt[2]{A}$ .

**Zadanie 3.** Napisz w MATLAB-ie program realizujący algorytm Newtona. Deklaracja funkcji powinna być następująca:

$[pn, err, ypn, k] =$   
 $newton(f, df, p0, delta, epsilon, max),$

gdzie

$f$  – nazwa funkcji,

$df$  – nazwa pochodnej funkcji,  
 $p_0$  – punkt początkowy,  
 $\delta$ ,  $\epsilon$  – dokładność rozwiązania; żądamy aby był spełniony warunek  $(|p_n - p_{n-1}| < \delta) \vee (|f(p_n)| < \epsilon)$ , gdzie  $p_n$  jest przybliżonym rozwiązaniem, a  $p_{n-1}$  poprzednim elementem ciągu,  
 $\max$  – ograniczenie na liczbę kroków; program nie może wykonać więcej niż  $\max$  iteracji,  
 $p_n$  – przybliżone rozwiązanie,  
 $err$  – błąd rozwiązania;  $err = |p_n - p_{n-1}|$ , gdzie  $p_n$  oraz  $p_{n-1}$  są dwoma ostatnimi elementami ciągu,  
 $yp_n$  – wartość funkcji w punkcie  $p_n$ ,  
 $k$  – liczba kroków po której wyznaczono rozwiązanie.

**Rozwiązanie:** `newton.m`



**Zadanie 4.** *Sprawdź program newton dla poniższych funkcji i punktów początkowych.*

a)  $f(x) = \sin(x)$ ,  $x_0 = 2$ ,

b)  $f(x) = \sin(x)$ ,  $x_0 = 1.57$ ,

c)  $f(x) = 1 - x^2 + \cos(x)$ ,  $x_0 = 2$ .

# Metoda siecznych

Jedną z wad metody Newtona jest wymóg znania pochodnej funkcji, której zera szukamy. Pomysł w metodzie siecznych polega na zastąpieniu pochodnej ilorazem różnicowym postaci

$$f'(x_n) \approx \frac{f(x_n) - f(x_{n-1})}{x_n - x_{n-1}}.$$

To przybliżenie daje metodę siecznych opisaną wzorem

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

dla  $n \geq 1$  i dwóch punktów początkowych  $x_0, x_1$ .

**Zadanie 5.** Napisz w *MATLAB*-ie program realizujący algorytm siecznych. Deklaracja funkcji powinna być następująca:

$[pn, err, ypn, k] =$   
 $secant(f, p0, p1, delta, epsilon, max),$

gdzie

$f$  – nazwa funkcji,

$p0, p1$  – punkty początkowe,

$delta, epsilon$  – dokładność rozwiązania; żądamy aby był spełniony warunek  $(|pn - p_{n-1}| < delta) \vee (|f(pn)| < epsilon)$ , gdzie  $pn$  jest przybliżonym rozwiązaniem, a  $p_{n-1}$  poprzednim elementem ciągu,

$max$  – ograniczenie na liczbę kroków; program nie może wykonać więcej niż  $max$  iteracji,

$pn$  – przybliżone rozwiązanie,

$err$  – błąd rozwiązania;  $err = |pn - p_{n-1}|$ , gdzie

$p_n$  oraz  $p_{n-1}$  są dwoma ostatnimi elementami ciągu,  
 $y_{pn}$  – wartość funkcji w punkcie  $p_n$ ,  
 $k$  – liczba kroków po której wyznaczono rozwiązanie.

**Rozwiązanie:** *secant.m*



**Zadanie 6.** Sprawdź program *secant* dla poniższych funkcji i wybranych dwóch punktów początkowych.

a)  $\sin(x/2) - 1$ ,

b)  $e^x - \operatorname{tg}(x)$ ,

c)  $x^3 - 12x^2 + 3x + 1$ .