Wojciech Myszka

Laboratorium 6: Funkcje. Metoda Newtona-Raphsona

2022-04-21 18:22:28 +0200

1. Postawienie problemu

Załóżmy, że mamy wyznaczyć pierwiastek stopnia n z liczby w, czyli znaleźć taką liczbę x, że:

$$x^n = w \tag{1}$$

lub inaczej:

$$x^n - w = 0 \tag{2}$$

Jeżeli oznaczymy $f(x) = x^n - w$ to zadanie to można zapisać ogólniej: należy znaleźć takie x, że $f(x) = 0$.

Jeżeli zadanie dodatkowo uprościmy zakładając:
— funkcja ma dokładnie jedno miejsce zerowe,
— jest różniczkowalna,
— jej pochodna w całym przedziale jest albo dodatnia albo ujemna;
to możemy naszkicować rysunek (rys. 1) ilustrujący nasze zadanie.

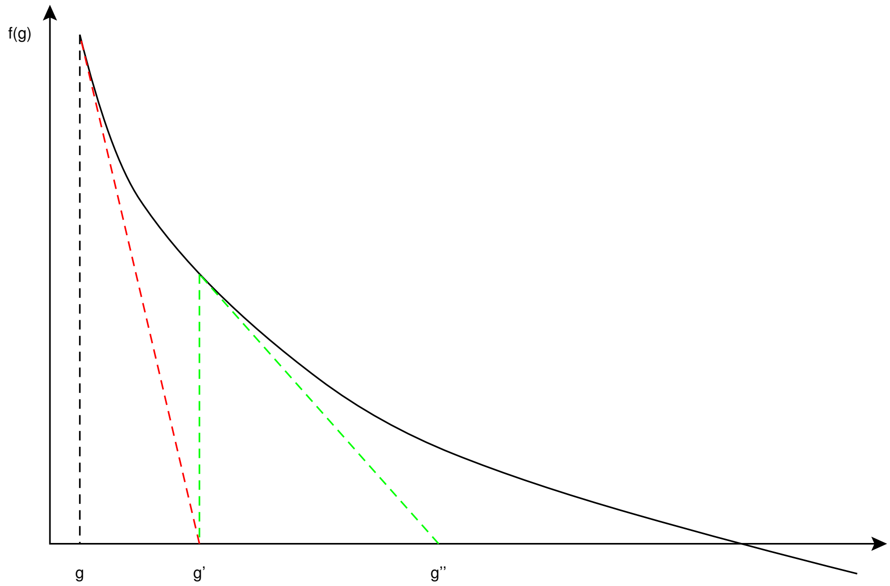
Zaczynamy w punkcie g; wartość funkcji w tym punkcie wynosi $f(g)$. Musimy w jakiś sposób zdecydować w którym kierunku należy wykonać następny krok. Zauważmy, że możemy w tym celu wykorzystać pochodną (czerwona, przerywana linia na poprzednim rysunku). Jeżeli przybliżymy funkcję za pomocą pochodnej (stycznej do) funkcji, przechodzącej przez punk $(g, f(g))$ to następnym przybliżeniem miejsca zerowego będzie punkt przecięcia stycznej z osią x.

Z równania prostej mamy:

$$\frac{f(g) - 0}{g - g'} = f'(g) \tag{3}$$

czyli

$$\frac{f(g)}{f'(g)} = g - g' \tag{4}$$



Rysunek 1. Przykład funkcji spełniającej założenia oraz ilustracja pierwszych kroków algorytmu

i dalej

$$g' = g - \frac{f(g)}{f'(g)} \tag{5}$$

Jeżeli zauważymy, że $f(x) = x^n - w$ oraz, że $f'(x) = nx^{n-1}$ to kolejne przybliżenie wyliczane będzie ze wzoru:

$$g' = g - \frac{g^n - w}{ng^{n-1}} \tag{6}$$

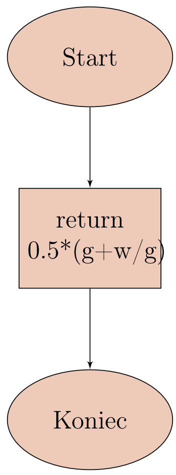
albo

$$g' = \frac{ng^n - g^n + w}{ng^{n-1}} = \frac{(n-1)g^n + w}{ng^{n-1}} = \frac{1}{n}\left((n-1)g + \frac{w}{g^{n-1}}\right) \tag{7}$$

Gdy $n = 2$, wówczas

$$g' = \frac{1}{2}\left(g + \frac{w}{g}\right). \tag{8}$$

Istotą algorytmu jest powtarzanie powyższych kroków „w nieskończoność“. Procedura jest asymptotycznie zbieżna i umawiamy się, że program kończy pracę gdy kolejna poprawka g' nie różni się zbytnio od poprzednio wyliczonej wartości g, czyli $|g - g'| < \varepsilon$ albo gdy wartość funkcji w punkcie g' nie różni się zbytnio od zera ($|f(g')| < \delta$). Można stosować oba te kryteria:
— łącznie — połączone spójnikiem logicznym AND,
— które „zadziała” pierwsze (spójnik logiczny OR).



Rysunek 2. Schemat blokowy funkcji lepszy

2. Realizacja programowa

Program będzie się składał z trzech części (funkcji):

1. `blisko(g, gprim)` — funkcja o wartościach logicznych sprawdzająca czy $|g - g'| \leqslant \varepsilon$,
2. `lepszy(n, w, g)` — funkcja rzeczywista wyliczająca następne, lepsze przybliżenie pierwiastka,
3. `pierwiastek(n, w, g)` — funkcja (rzeczywista) wyliczająca pierwiastek stopnia n z w zaczynając od przybliżenia g.
4. i, oczywiście, funkcji main.

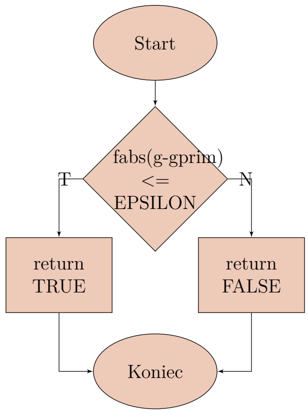
Uwaga: Dalszy przykład zakłada $n = 2$.

Na rysunkach 2–5 przedstawione są schematy blokowe poszczególnych funkcji. Funkcja pierwiastek realizowana jest w sposób rekurencyjny, ale można z rekurencji zrezygnować i obliczenia przeprowadzać w pętli (na przykład **while**).

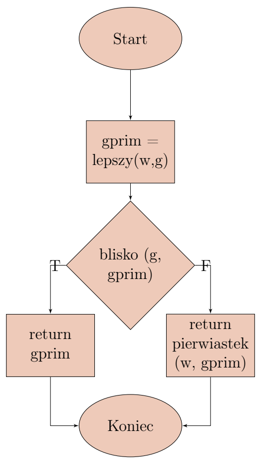
3. Zadania do wykonania

1. Zaprogramować metodę Newtona-Raphsona dla $n = 2$. Przetestować jej działanie.
2. Zmodyfikować schematy blokowe i programy aby metoda mogła działać dla dowolnego n (Uwaga: n nie musi być całkowite!). Przetestować.

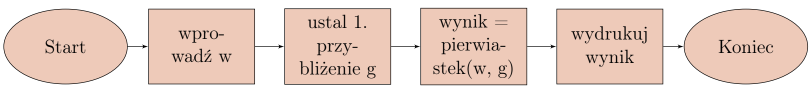
Program może być rekurencyjny lub nie.



Rysunek 3. Schemat blokowy funkcji blisko (TRUE to w języku C 1, a FALSE to 0)



Rysunek 4. Schemat blokowy rekurencyjnej wersji funkcji pierwiastek



Rysunek 5. Schemat blokowy funkcji main

4. Wersja PDF tego dokumentu...

... pod adresem.

Wersja: 61 z drobnymi modyfikacjami! data ostatniej modyfikacji 2022-04-21 18:22:28 +0200