Metoda BISEKCJI (płowienia)

Jeśli funkcja jest ciągła i na krańcach przedziału posiada różne znaki, to z twierdzenia twierdzenie o wartości średniej mamy, że jeżeli

f(*xp*) < 0 < *f*(*xk*) lub *f*(*xp*) > 0 > *f*(*xk*),

to istnieje w przedziale <xp, xk> punkt *x*o, taki, że

f(*x*o) = 0

Twierdzenie to gwarantuje istnienie przynajmniej jednego pierwiastka w rozważanym przedziale. **Metoda bisekcji (połowienia)** polega na dzieleniu zadanego przedziału argumentów na dwie równe połówki.

Metoda bisekcji (połowienia)

Punkt środkowy przedziału znajdujemy jako średnią arytmetyczną jego krańców

$$x_0 = \frac{x_p + x_k}{2}$$

Jeśli wartość funkcji w punkcie podziału jest równa zero (lub dostatecznie bliska zera), to punkt ten traktujemy jako pierwiastek funkcji i kończymy wykonywanie algorytmu.

Jeżeli wartość funkcji w punkcie podziału nie jest równa zero, więc musi być od niego większa lub mniejsza.

Sprawdzamy, w której z otrzymanych przez podział połówek funkcja zmienia znak na przeciwny na krańcach - jest tylko jedna taka połówka.

Wybraną połówkę traktujemy jako nowy przedział zawierający pierwiastek i znów dzielimy ją na dwie części sprawdzając wartość funkcji w środku przedziału.

Operacje te kontynuujemy, aż do znalezienia pierwiastka. Punkt *x*o w granicy nieskończonych podziałów jest zbieżny do punktu będącego poszukiwanym pierwiastkiem zadanej funkcji.

Ponieważ nie możemy wykonywać tych obliczeń w nieskończoność, to zadawalamy się **przybliżoną wartością pierwiastka**, którą otrzymamy po kilkunastu obiegach.

Przykład 1 (metoda bisekcji)

Obliczymy za pomocą metody **bisekcji** pierwiastek funkcji $f(x) = x^3 - x^2 + 2x - 1$

w przedziale <0,1> z dokładnością do 0,01.

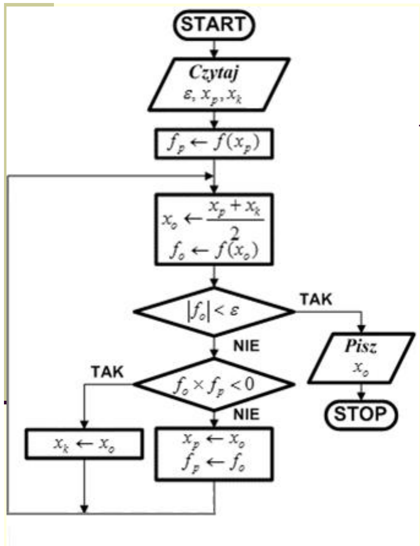
Funkcja jest ciągła w badanym przedziale.

Obliczamy wartość funkcji *f*(x) na krańcach przedziału

$f(x_p) = f(0) = 0 - 0 + 0 - 1 = -1$

$f(x_k)$ = f(1) = 1^3 – 1^2 + 2 - 1 = 1

Funkcja posiada różne znaki w punktach krańcowych przedziału, więc zgodnie z twierdzeniem o wartościach pośrednich w przedziale tym znajduje się pierwiastek funkcji.



1. wczytanie danych wejściowych: dokładności wyznaczania zera funkcji ***ε***, oraz początku ***xp*** i końca ***xk*** przedziału

2. obliczamy wartość funkcji w punkcie *xp* i umieszczamy wynik w zmiennej ***fp***.

3,. Pętla obliczająca pierwiastek

a) wyznaczamy środek przedziału *xo* oraz wartość funkcji w tym punkcie.

b) sprawdzamy, czy zbliżyliśmy się dostatecznie blisko do zera funkcji.

c) jeśli f(*x*o) będzie co do modułu mniejsza od *ε*, to przyjmujemy, iż *xo* jest pierwiastkiem przybliżonym funkcji *f*(*x*) i kończymy algorytm.

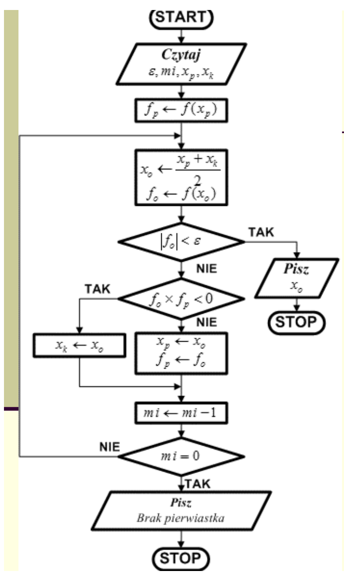
W przeciwnym razie zawężamy przedział, przyjmując za nowy przedział tę połówkę, w której funkcja zmienia znak i kontynuujemy pętlę.

Z matematycznego punktu widzenia jeśli warunki początkowe zostaną spełnione, to metoda połowienia zawsze da rozwiązanie, ale komputery nie wykonują obliczeń z nieskończoną dokładnością - to co jest zbieżne w matematyce, w obliczeniach numerycznych może być rozbieżne.

Jeśli założona dokładność będzie mniejsza od błędów zaokrągleń wyliczania wartości funkcji, to może się zdarzyć, iż algorytm nigdy nie da rozwiązania, zawieszając przy okazji program.

Może zdarzyć się, że krańce przedziału będą się do siebie coraz bardziej zbliżać, aż przekroczywszy dokładność zastosowanej arytmetyki liczb zmiennoprzecinkowych staną się sobie równe (w matematyce nigdy się to nie zdarzy). A wówczas w kolejnych krokach wartość x_o będzie już zawsze taka sama i w każdym następnym obiegu algorytmu nic się więcej nie zmieni - dojdzie do zablokowania algorytmu (uwagi te odnoszą się również do pozostałych algorytmów).

Rozwiązaniem tej sytuacji może być zastosowanie licznika obiegów, który przerwie jej wykonywanie po osiągnięciu zadanej wartości w przypadku, gdyby pierwiastek nie został znaleziony normalnie.



Warto wprowadzić ograniczenie maksymalnej liczby obiegów pętli obliczeniowej ***mi***

jeśli f(*x*o) będzie co do modułu mie jest mniejsza od *ε*, to zawężamy przedział, przyjmując za nowy przedział tę połówkę, w której funkcja zmienia znak.

Zmniejszamy o 1 licznik obiegów pętli i jeśli jest on różny od 0, pętlę kontynuujemy. W przeciwnym wypadku pętla jest przerywana z odpowiednim komunikatem dla użytkownika.

Osiągnięcie przez licznik *mi* wartości 0 oznacza, iż algorytm nie był w stanie wyznaczyć pierwiastka funkcji z założoną dokładnością w danej liczbie kroków.

Metoda relaksacyjna

odmiana metody połowienia, kiedy dzielimy na 10 części zamiast na dwie i szukamy w którym kawałku funkcja zmienia znak

Istnieją jeszcze inne odmiany, kiedy dzielimy na inną ilość części