Rozwiązywanie równań nieliniowych metodami połowienia, *regula falsi* i siecznych w arytmetyce przedziałowej

Mikołaj Rozwadowski

27 maja 2017

1 Informacje ogólne

1.1 Zastosowanie

Każda z niżej opisanych metod rozwiązuje w arytmetyce przedziałowej równanie $f(x) = 0$, gdzie f jest dowolną funkcją rzeczywistą. Do znalezienia rozwiązania potrzebny jest początkowy przedział $[a, b]$, który jest potem sukcesywnie zawężany.

1.2 Sposób wczytania funkcji

Wymagana przez każdą metodę funkcja f jest przekazywana jako wskaźnik do obiektu klasy `Function`. Można samemu napisać podklasę realizującą jakąkolwiek funkcję albo posłużyć się klasą `SOFunction`, która umożliwia załadowanie jej z biblioteki dynamicznej `.so` pod systemami operacyjnymi z rodziny GNU/Linux. Sposób utworzenia i skompilowania kompatybilnej biblioteki opisany jest na stronie projektu w serwisie GitHub.

1.3 Identyfikatory nielokalne

W pliku nagłówkowym `common.h` znajdują się deklaracje pomocniczych funkcji i stałych, z których korzystają wszystkie metody:

`Function`
: klasa udostępniająca metodę `interval evaluate(interval x)`.

`WRONG_INTERVAL`
: stała o wartości liczbowej 1; jest to wyjątek, którym funkcja sygnalizuje, że w podanym przedziale początkowym lewy koniec jest większy lub równy od prawego końca,

`NO_REAL_ROOTS`
: stała o wartości liczbowej 2; jest to wyjątek, którym funkcja sygnalizuje brak pierwiastków rzeczywistych w podanym przedziale początkowym,

`check_interval`
: pomocnicza funkcja, która sprawdza warunki dla przedziału początkowego, a niespełnienie ich sygnalizuje zgłoszeniem jednego z powyższych wyjątków.

2 Metoda połowienia przedziału

2.1 Zastosowanie

Funkcja `Bisection` znajduje wartość pierwiastka równania $f(x) = 0$ metodą połowienia przedziału w arytmetyce przedziałowej.

2.2 Opis metody

Metoda połowienia wykorzystuje własność Darboux, która mówi, że jeśli dana jest funkcja ciągła f i przedział rzeczywisty $[a, b]$ takie, że $f(a) = y_a$ i $f(b) = y_b$, to funkcja ta przyjmuje w przedziale (a, b) wszystkie wartości pośrednie między y_a a y_b. W szczególności oznacza to, że jeśli taka funkcja ma na końcach przedziału różne znaki, to istnieje tam miejsce zerowe, czyli punkt $x_0 \in (a, b)$, dla którego $f(x_0) = 0$.

Algorytm rozpoczyna pracę z danym przedziałem $[a, b]$, o którym wiadomo, że $f(a) \cdot f(b) < 0$. W każdej iteracji następuje wyznaczenie środka przedziału $m = \frac{a+b}{2}$ i podział go na dwie połowy $[a, m]$ i $[m, b]$. Jeżeli $f(a) \cdot f(m) < 0$, to przeszukiwanie jest kontynuowane w przedziale $[a, m]$, w przeciwnym razie $[m, b]$.

Pętla powtarza się aż do momentu, w którym środek przedziału m będzie miejscem zerowym funkcji. Ponieważ w praktyce może to jednak nie nastąpić, do przerwanie algorytmu dojdzie też, gdy szerokość przedziału będzie mniejsza niż zadana tolerancja ϵ lub po wykonaniu określonej liczby iteracji. W takim wypadku wynikiem będzie najwęższy uzyskany przedział zawierający pierwiastek.

2.3 Wywołanie funkcji

`Bisection(a, b, func, tolerance, iterations, reached)`

2.4 Dane

`a, b`
: lewy i prawy koniec przedziału zawierającego pierwiastek,

`func`
: funkcja, której miesce zerowe należy znaleźć,

`tolerance`
: największa akceptowalna szerokość przedziału wynikowego,

`iterations`
: maksymalna liczba iteracji.

2.5 Wyniki

`Bisection(a, b, func, tolerance, iterations, reached)`
: przedział zawierający miejsce zerowe

2.6 Inne parametry

`reached`
: określa, czy w `iterations` iteracjach udało się zmieścić w wymaganej tolerancji.

2.7 Typy parametrów

`interval a, interval b, Function* func, long double tolerance, int iterations, bool& reached`

2.8 Identyfikatory nielokalne

`NOT_ENOUGH_ITERATIONS`

stała o wartości liczbowej 3; jest to wyjątek, którym funkcja sygnalizuje, że parametr `iterations` nie jest liczbą dodatnią.

2.9 Przykłady

Równanie	a	b	liczba iteracji	ϵ	x_0
$x^2 - 2 = 0$	1	2	60	10^{-16}	[1.41421356237309503, 1.414213562
$xe^{\sqrt{x+1}} - 1 = 0$	-1	1	60	10^{-16}	[3.17347582146508266, 3.173475821
$\sin x \cdot (\sin x + \frac{1}{2}) - \frac{1}{2} = 0$	0	1	60	10^{-16}	[5.2359877559829803e-01, 5.235987755

3 Metoda *regula falsi*

3.1 Zastosowanie

3.2 Opis metody

3.3 Wywołanie funkcji

3.4 Dane

3.5 Wyniki

3.6 Inne parametry

3.7 Typy parametrów

3.8 Identyfikatory nielokalne

3.9 Przykłady

4 Metoda siecznych

4.1 Zastosowanie

4.2 Opis metody

4.3 Wywołanie funkcji

4.4 Dane

4.5 Wyniki

4.6 Inne parametry

4.7 Typy parametrów

4.8 Identyfikatory nielokalne

4.9 Przykłady

5 Bibliografia

- A. Marciniak, D. Gregulec, J. Kaczmarek: *Podstawowe procedury numeryczne w języku Turbo Pascal*. Wydawnictwo Nakom, Poznań, 2000

r.