Programowanie i metody numeryczne

zestaw zadań 8

Różniczkowanie, równania nieliniowe, całkowanie. Wszystko jednej zmiennej.

Opracowanie *Jędrzej Wardyn*

22 kwietnia 2024

Polecam: kody C++, jak ktoś potrzebuje sobie C++ powtórzyć: Rozwiązania na githubie Kurs na youtubie

Materiały z wykładu od M.M.(wykład 4.4 pdf):na githubie
(wykład 11-04.pdf):na githubie

1 Różniczkowanie I

1.1 Pochodna w punkcie

Napisz template (szablon):

```
template <typename T>
T diff(T (*f)(T), T x, T h, string method)
```

gdzie `T` oznacza nazwę zastosowanego typu (na przykład `double`, tylko nie macie go wpisywać w templatkę, a jedynie w funkcji main używacie konkretnego typu na wartości np a, b i wtedy kod przyjmie te wartości jako `T`.)

Ta funkcja ma zawierać metody obliczające pochodną zadaną metodą dla danego odwzorowania $y = f(x)$, określonego przez `f` w punkcie `x`. Wykorzystaj następujące wzory:

$$f'_{\text{forward}}(x) = \frac{f(x+h) - f(x)}{h},$$
$$f'_{\text{backward}}(x) = \frac{f(x) - f(x-h)}{h},$$
$$f'_{\text{central}}(x) = \frac{f(x+h) - f(x-h)}{2h},$$
$$f'_{\text{richardson}}(x) = \frac{-f(x+2h) + 8f(x+h) - 8f(x-h) + f(x-2h)}{12h}.$$

Źródła: Finite difference

Numerical differentiation

Napisz program testowy sprawdzający poprawność działania tych wzorów dla kilku przykładowych różniczkowanych odwzorowań (e^x, x^4, ($log(x)$ dla wartości $x > 1$)) i różnych wartości $h = 0.1, 0.001, 0.000000001$.
`diff` przyjmuje jako `method` jedno ze słów: forward, backward, central lub richardson, określające metodę różniczkowania, funkcję odwzorowania $f(x)$, punkty x oraz liczbę rzeczywistą określającą wartość h.

Następnie, wypisz wartości znane pochodnych tych funkcji i porównaj je z obliczonymi pochodnymi. Sprawdź też jak zmieni się wartość funkcji, jeżeli zamiast `double` użyjesz `float`.
*dla ciekawskich: typ float o obniżonej precyzji w C++

1.2 Aplikowanie pochodnej do wektorów punktów x,$f(x)$

Skopiuj kod poprzedniego punktu do nowego pliku `.cpp` i zmodyfikuj funkcje C++ tak, by przyjmowały wektory `vector<double> x` oraz `vector<double> y` (gdzie `y` zawiera wartości `f(x)` dla wektora `x`), a następnie zwracały wektor ze zróżniczkowanymi wartościami. Jest pewna istotna różnica: tym razem h należy obliczyć wewnątrz funkcji, a nie wprowadzić.

1.3 * dla chętnych

*Dla chętnych: spróbuj zaaplikować metodę różniczkowania wyższego rzędu lub z większą ilością punktów metody wyższych rzędów i z większą ilością punktów

2 Znajdowanie miejsc zerowych w równaniach nieliniowych jednej zmiennej

Autor zadania: B.Z.
Napisz 3 szablony funkcji (+1 dla chętnych)

```
template < typename F >
double rootFunkcja ( F f , double a , double b , double eps )
```

W nazwie `rootFunkcja` w miejscu `Funkcja` mamy mieć

1. `Bisection`
2. `Falsi`
3. `Newton`
4. *[dla chętnych]Secant

Czyli robimy po jednym szablonie dla każdego z punktów. Szablon ma spełniać warunki opisane poniżej.

Napisz szablony, które posługując się metodą (1. Bisecji, 2.Falsi , 3. Newtona) znajdują miejsce zerowe ciągłej funkcji $f : \mathcal{R} \to \mathcal{R}$ (czyli przyjmuje `double` i oddaje `double`) położone w przedziale $[a, b] \in X$, przy czym $f(a)f(b) < 0$, z dokładnością ϵ. Funkcja `rootFunkcja` przyjmuje argument f – implementację funkcji f oraz trzy argumenty a, b i ϵ typu double, odpowiadające kolejno liczbom a, b i ϵ. Wartością zwracaną przez funkcję `rootFunkcja` powinno być znalezione przez nią miejsce zerowe odwzorowania f.

2.1 Założenia i wytyczne:

Załóż, że w przedziale $[a, b]$ znajduje się co najwyżej jedno miejsce zerowe odwzorowania f .

Funkcja `rootFunkcja` powinna sprawdzać, czy wartości jej argumentów są poprawne oraz czy spełniają założenia (a więc np. czy odpowiadają $f(a)f(b) < 0$, $a < b$, $0 < \epsilon \leq (b - a)$). Jeśli okaże się, że tak nie jest, funkcja powinna zgłosić odpowiedni wyjątek (pochodzący z biblioteki standardowej lub napisany specjalnie na jej potrzeby) opatrzony komunikatem wyjaśniającym przyczynę jego wystąpienia.

2.2 Co trzeba zrobić:

Napisz program testowy (to w `main()` można) sprawdzający poprawność działania tego szablonu dla trzech przykładowych odwzorowań: $f(x) = (x^2 - 2)$, $g(x) = (\log(x + 6) - x^3)$ oraz $h(x) = (e^x + x - 1)$, poszukujący dla każdej z nich miejsca zerowego w przedziale $[-1, 6]$ z różnymi przykładowymi dokładnościami: 0.1, 0.01, 0.001, 0.0001, 0.0000001. Następnie, korzystając z tego szablonu, napisz program "Funkcja", który przyjmuje jako argumenty wywołania trzy liczby zmiennoprzecinkowe określające wartości a, b i ϵ. Program powinien wczytywać ze standardowego wejścia liczby zmiennoprzecinkowe aż do napotkania znaku końca pliku, następnie konstruować wielomian z tymi liczbami jako współczynnikami i znajdować miejsce zerowe tego wielomianu w zadanym przedziale i z zadaną dokładnością. Możesz założyć, że w zadanym przedziale znajduje się co najwyżej jedno miejsce zerowe tego wielomianu.

3 Całkowanie I: metodami Newtona-Cotesa

Mając całkę oznaczoną:

$$I = \int_a^b f(x)\mathrm{d}x$$

dla odwzorowania $y = f(x)$. Napisz szablon do różnych metod całkowania za pomocą templatki:

```
template <typename T>
T integrate(T (*f)(T), T x, T h, string method) {
```

gdzie `T` oznacza nazwę zastosowanego typu (na przykład `double`, tylko nie macie go wpisywać w templatkę, a jedynie w funkcji main używacie konkretnego typu na wartości np a, b i wtedy kod przyjmie te wartości jako `T`.) Wykorzystaj wzory Newtona-Cotesa:

1. Rectangular
2. Trapezoid
3. Simpson

Porównaj dokładność metod dla odwzorowania $\sin(x)$ w przedziale $[0, \pi]$