Całkowanie Numeryczne

Wprowadzenie

Metody numeryczne rozpatrują trzy główne rodzaje całkowania:

1. numeryczne obliczanie wartości całki oznaczonej,
2. numeryczne rozwiązywanie równań różniczkowych zwyczajnych (gdzie znalezione rozwiązanie jest całką funkcji jednej zmiennej),
3. numeryczne rozwiązywanie równań różniczkowych cząstkowych (gdzie znalezione rozwiązanie jest całką funkcji dwóch lub więcej zmiennych).

Tematem bieżącego ćwiczenia jest pierwszy rodzaj całkowania, gdzie poszukujemy wartości:

gdzie *f(x)* jest funkcją, której wartości całki oznaczonej w przedziale *<a,b>* poszukujemy. Funkcja *f(x)* może być dostępna w kilku formach: w postaci analitycznej (wtedy zwykle obliczamy całkę numeryczne z powodu trudnego analitycznego całkowania) lub w postaci zbioru punktów *xi, f(xi)*. Ten drugi przypadek jest szczególnie istotny, gdy dokonujemy przetwarzania danych pomiarowych (np. z oscyloskopu).

Jedną z podstawowych metod całkowania numerycznego są metody Newtona-Cotesa. Polegają one na interpolacji danej funkcji wielomianem Lagrange’a. W zależności od liczby znanych punktów (*xi, fi*) w danym przedziale całkowania, mamy do czynienia z metodami Newtona-Cotesa różnych rzędów:

1. rząd zerowy – znamy jeden punkt (zwykle *b*, *f(b)*),
2. rząd pierwszy – znamy dwa punkty (na krańcach przedziału całkowania),
3. rząd trzeci – znamy trzy punkty, itd.

W metodzie rzędu zerowego, przybliżamy naszą funkcję poziomą linią prostą (wielomianem zerowego rzędu). W przypadku metody rzędu pierwszego, wielomian interpolacyjny również jest pierwszego rzędu (linia prosta).

Zwykle wykorzystujemy metody Newtona-Cotesa najwyżej piątego rzędu. Jakie mogą być tego powody?

Często dysponujemy dużą liczbą punktów, w których mamy spróbkowaną funkcję podcałkową. Żeby otrzymać dokładną wartość dzielimy punkty na podprzedziały, w których wykorzystujemy metodę Newtona-Cotesa niskiego rzędu. Taki sposób, nazywamy metodą złożoną Newtona-Cotesa. Po zsumowaniu wyników całkowania podprzedziałami otrzymujemy wynik. Przykładowo: posiadając spróbkowaną funkcję w 9 punktach: (*x1, f1*), (*x2, f2*), (*x3, f3*), (*x4, f4*), (*x5, f5*), (*x6, f6*), (*x7, f7*), (*x8, f8*), (*x9,  f9*) możemy policzyć szukaną całkę określoną od *x1* do *x9* obliczając trzy całki na podprzedziałach:

1. metodą Newtona-Cotesa trzeciego rzędu, znając: (*x1, f1*), (*x2, f2*), (*x3, f3*),
2. metodą Newtona-Cotesa trzeciego rzędu, znając: (*x3, f3*), (*x4, f4*), (*x5, f5*),
3. metodą Newtona-Cotesa trzeciego rzędu, znając: (*x6, f6*), (*x7, f7*), (*x8, f8*),
4. metodą Newtona-Cotesa drugiego rzędu, znając: (*x8, f8*), (*x9,  f9*).

Wynik jest sumą powyższych czterech wyników cząstkowych. Oczywiście można dokonać też innego podziału (np. rz. 4 + rz. 4 + rz. 3).

Zadanie nr 1

Celem tego ćwiczenia jest samodzielne zaimplementowanie prostej metody Newtona-Cotesa.

Zaimplementowana metoda powinna przyjmować następujące argumenty: wektor wartości funkcji *f*, długość kroku całkowania *h*. Przykładowe dane:

fi = [11, 0.0107421875, 0, 0.0107421875, 11];

h = 0.5;

w wyniku działania funkcji powinna zostać zwrócona wartość całki oznaczonej obliczonej według następujących wzorów Newtona-Cotesa:

1. dla jednoelementowego wektora fi należy zastosować wzór prostokątów (metoda otwarta),
2. dla dwuelementowego wektora fi należy zastosować wzór trapezów (metoda zamknięta),
3. dla trzyelementowego wektora fi należy zastosować wzór Simpsona,
4. dla czteroelementowego wektora fi należy zastosować regułę 3/8,
5. dla pięcioelementowego wektora fi należy zastosować wzór Boole’a,
6. dla dłuższych wektorów należy zwrócić wartość NaN (not-a-number).

Sprawdzić wyniki swojej funkcji na danych z Tabeli 1:

Tabela 1. Dane do sprawdzenia poprawności implementacji wzorów Newtona-Cotesa.  
Wyniki powinny być zgodne co najmniej z dokładnością do czterech miejsc po przecinku.

|  |  |  |  |
| --- | --- | --- | --- |
| Fi | h | *wynik* | *rząd* |
| [11] | 0.5 | 5.5 | 0 |
| [0, 11] | 1 | 5.5 | 1 |
| [0, 0.0107, 11] | 0.5 | 1.84046 | 2 |
| [0, 0.000186, 0.1907, 11] | 0.333 | 1.44514 | 3 |
| [0, 0.00001, 0.0107, 0.61945, 11] | 0.25 | 1.07724 | 4 |
| [11, 0.0107421875, 0, 0.0107421875, 11] | 0.5 | 3.43750 | 4 |

Zadanie nr 2

Celem tego ćwiczenia jest samodzielne zaimplementowanie złożonej metody  
Newtona-Cotesa. Należy przygotować funkcję, która otrzyma następujące argumenty:

wektor wartości funkcji *f*, długość kroku całkowania *h* oraz maksymalny rząd metody całkowania *d* (założyć *d > 0* i *d <* 5). Należy wziąć pod uwagę również sytuację, w której ostatni podprzedział będzie policzony metodą niższego rzędu niż *d* (wybrać największy możliwy).

Należy samodzielnie wygenerować 201 punktów w przedziale od -1 do +1, z krokiem całkowania 0.01, dla podanej poniżej funkcji *f*:

, .

Następnie, należy obliczyć wartość całki oznaczonej w podanym przedziale za pomocą metody Newtona-Cotesa. Zastosować następujące rzędy metody: 1, 2, 3, 4. Wyniki zapisać w Tabeli 2

Tabela 2. Wyniki obliczeń całki oznaczonej funkcji *f*(*x*), *h*=0.01.  
Podać wyniki z dokładnością do dwóch cyfr znaczących.

|  |  |
| --- | --- |
| *rząd* | *wynik* |
| 1 | 2.001833113355330 |
| 2 | 2.000000879560092 |
| 3 | 2.000001905563901 |
| 4 | 2.000000001406522 |
|  |  |

Zadanie nr 3

Celem tego ćwiczenia zbadanie szybkości zbieżności różnych rzędów metody Newtona-Cotesa. Badanym kodem będzie rozwiązanie zadania nr 2.

Należy obliczyć następujące warianty wzorami Newtona-Cotesa dla rzędów *d* 1, 2, 3, 4:

Liczba punktów , gdzie *.*

*Dla przypomnienia*: .

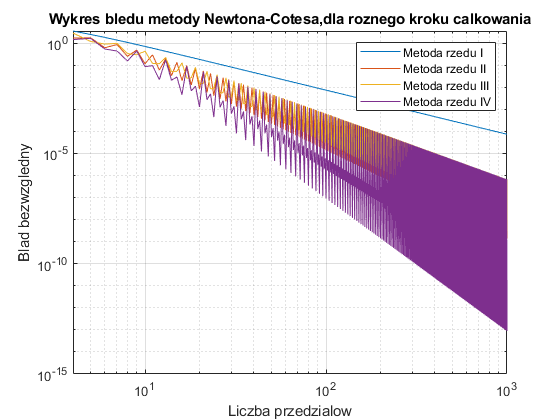
Narysować wykres z czterema przebiegami (dla każdego rzędu) wartości błędu bezwzględnego względem rozwiązania dokładnego, tj. 2.0. Z uwagi na konieczność porównania wartości na osiach o dużej rozpiętości należy wykorzystać skalę podwójnie logarytmiczną:

loglog([1:1000], abs(result\_nc\_1 – 2.0),...

[1:1000], abs(result\_nc\_2 – 2.0),...

[1:1000], abs(result\_nc\_3 – 2.0),...

[1:1000], abs(result\_nc\_4 - 2.0))



Rysunek 1. Wykres błędu dla czterech rzędów metody Newtona-Cotesa,  
dla różnego kroku całkowania. Skala podwójnie logarytmiczna.

Co podlega głównej ocenie

Najważniejszym elementem oceny jest umiejętność formułowania własnych wniosków z przeprowadzonych ćwiczeń oraz zdolność do samodzielnego, poprawnego zaimplementowania wskazanych metod całkowania numerycznego. Kolejnymi elementami oceny są również: staranność przygotowanego kodu oraz zamieszczonych ilustracji.

Wnioski:

Na podstawie obserwowanych wyników przedstawionych w powyższej *tabeli 2* możemy stwierdzić, że dokładność obliczeń całek wzrasta wraz wzrostem rzędu metody Newtona-Cotesa. Ze względu na wynik zbliżony do wyniku rzeczywistego, możemy też stwierdzić, że implementacja metody Netwona-Cotesa jest poprawna. Otrzymaliśmy bardzo podobne wyniki, właśnie na to oczekiwaliśmy.

W trzecim zadaniu na podstawie drugiego zbadaliśmy zależność liczby punktów od błędu względem rozwiązania dokładnego (to znaczy dla 2).

Na podstawie wykresu można zaobserwować, że wraz ze wzrostem liczby przedziałów błąd całkowania maleje i trochę wzrasta. Widać także, że błąd całkowania jest mniejszy dla metod wyższego rzędu. Z powodu braku możliwości policzenia całek metodami wyższych rzędów dla małej ilości przedziałów, najmniejsza liczba przedziałów została ustalona na 5.

Jak widać błąd całkowania maleje i wzrasta, ale całkowicie on maleje, to zależy od pewnej liczby przedziałów. Można założyć, że to powód kroku całkowania lub mamy gdzieś błąd w napisaniu laboratorium. Najgłówniejszy wniosek to błąd całkowania jest mniejszy dla metod wyższego rzędu.