Całkowanie numeryczne

Pliki do wykorzystania w poniższym ćwiczenie można pobrać za pomocą poniższych linków:

- Plik nagłówkowy kwad.h
- Plik źródłowy kwad.cpp

1. Słowem wstępu

Całkowanie numeryczne jest jednym z podstawowych algorytmów używanych w obliczeniach inżynierskich. Pamiętajmy, że całkowanie numeryczne zawsze dotyczy obliczania całki oznaczonej (jedno- bądź wielowymiarowej). Nigdy natomiast nie dotyczy obliczania całki nieoznaczonej. Wynikiem jego działania jest wartość całki, a w żadnym razie wzór funkcji pierwotnej.

2. Algorytmy całkujące

Wśród dwóch podstawowych algorytmów, jakim będziemy się zajmować jest metoda trapezów oraz metoda Simpsona.

Ćwiczenia

1. Napisz program obliczający metodą trapezów całkę oznaczoną o wzorze:

$$g(x) = \int_a^b f(x)dx$$

Algorytm całkowania metodą trapezów jest zaimplementowany. Znajdziesz go w funkcji `trapez` w pliku *kwad.cpp*. Nagłówek funkcji `trapez` ma następującą postać:

```
double trapez(double a, double b, double (*fun)(double x), int n)
```

Argumenty `a`, `b` i `n` to odpowiednio dolna i górna granica całkowania oraz liczba przedziałów, na które dzielony jest obszar całkowania. Zwróć uwagę na trzeci argument. Jest to wskaźnik do funkcji. Dzięki temu procedura trapez jest w stanie w taki sam, ogólny sposób liczyć całkę z dowolnej funkcji. Po nazwie (wskaźniku do funkcji) wie, której funkcji ma użyć do obliczania wartości funkcji podcałkowej. Poprawne wywołanie to np. `trapez(a, b, sin, n);` lub `trapez(1, 5, sqrt, 100);`. Możesz też użyć tej procedury do przecałkowania funkcji, którą wcześniej zdefiniowałeś, np. `trapez(a, b, MojaFunkcja, 50);` pod warunkiem, że we wcześniejszym miejscu w kodzie ta funkcja jest określona. Np. tak:

```
double MojaFunkcja(double x) // funkcja musi byc typu double i miec 1 argument
{
    return x*x+sin(x);
}
```

Aby zrealizować zadanie z punktu 1, wykonaj następujące czynności: - Napisz funkcję obliczającą $f(x)$ oraz funkcję obliczającą całkę w sposób analityczny (przecałkuj na papierze). Umieść ich prototypy przed funkcją główną oraz załącz plik nagłówkowy *kwad.h* z funkcjami całkującymi. - Czytaj z klawiatury `a`, `b` oraz `n`- liczbę podziałów. - Obliczaj całkę numerycznie `cn` oraz analitycznie ca przez wywołanie odpowiednich funkcji. - Obliczaj błąd $|c_n - c_a|$. - Wpisuj do pliku krok całkowania oraz wartość całki policzonej analitycznie oraz numerycznie.

2. Przetestuj ten program dla:

$$f(x) = \frac{1}{x^2}$$

$$f(x) = \frac{1}{x}$$

dla $a = 1, b = 5$ oraz $a = 0.1$ i $b = 5$.

3. Rozszerz ten program tak, aby zapisywał do pliku kolejne wierwsz odpowiadające $n = 2, 4, 8, ..., 2^m$.

4. Wynik przedstaw graficznie, korzystając z Excela.

5. Rozszerz program tak, aby dodatkowo używał metody Simpsona. Wymaga to tylko kosmetycznych zmian. Cały potrzebny szkielet już masz. Metoda Simpsona jest zaimplementowana w procedurze `simpson(double a, double b, double(*fun)(double x), int n)`.

6. Przedstaw wyniki graficznie.

Wskazówki odnośnie wskaźników do funkcji

Przeanalizuj poniższy kod ilustrujący użycie wskaźników do funkcji. Jakie wartości przyjmują zmienne `y1` i `y2`?

```
//funkcja: y = 2*x
double fun1(double x)
{
    return 2*x;
}

// funkcja: y = -x
double fun2(double x)
{
    return -x;
}

// funkcja zwaraca y^2
double kwadrat(double xx, double (*pf)(double))
{
    return pf(xx)*pf(xx);
}

void main()
{
    double y1 = kwadrat(2., fun1);
    double y2 = kwadrat(2., fun2);
}
```

3. Dla dociekliwych

Procedury trapezów i Simpsona to dość elementarne procedury. W programach obliczeniowych, które wymagają wykonywania całkowań wielokrotnie (często wiele milionów razy - o takich metodach, np. metodzie elementów skończonych - stosowanej choćby w wytrzymałości konstrukcji - będziesz się uczyć na wyższych latach) trzeba używać procedur najbardziej wydajnych obliczeniowo. Jedną klasę takich metod, które osiągają dużą dokładność przy niewielkiej liczbie punktów, w których obliczana jest funkcja podcałkowa (o tych punktach mówimy węzły kwadratury) stanowią kwadratury Gaussa.

Kwadratura Gaussa jest w oryginalnej postaci zdefiniowana dla następującej całki oznaczonej (zawsze w przedziale $x = [-1, 1]$).

$$I = \int_{-1}^{1} f(x)dx$$

Całkę w sposób przybliżony oblicza się wg następującego wzoru:

$$\int_{-1}^{1} f(x)dx \approx \sum_{i=1}^{n} w_i f(x_i)$$

gdzie w_i to kolejne wagi kwadratury, x_i to węzły kwadratury, a n oznacza liczbę węzłów, w których będzie obliczana wartość funkcji podcałkowej. Aby policzyć całkę kwadraturą Gaussa, trzeba znać położenia węzłów i wartości wag. Można je obliczyć (istnieją odpowiednie procedury) lub też dla wybranych wartości n można je znaleźć w internecie[1]. Położenia węzłów i wartości wag dla $n = 5$ są podane w poniższej tabeli:

x_i	w_i
-0.9061798459386639927976269	0.2369268850561890875142640
-0.5384693101056830910363144	0.4786286704993664680412915
0.0	0.5688888888888888888888889
0.5384693101056830910363144	0.4786286704993664680412915
0.9061798459386639927976269	0.2369268850561890875142640

Ćwiczenie

Zaimplementuj w dowolny sposób (choćby w pętli - niekoniecznie w osobnej procedurze) metodę całkowania za pomocą kwadratury Gaussa. Porównaj ją z poprzednimi metodami. Ilu podziałów w metodzie trapezów lub Simpsona musisz użyć, aby osiągnąć dokładność całkowania osiąganą przez kwadaturę Gaussa opartą na pięciu węzłach?

[1] Wystarczy do wyszukiwarki wpisać hasło „Legendre Gauss nodes and weights". Da się je znaleźć choćby pod adresem