Laboratorium 4 – całkowanie numeryczne

Tomasz Belczyk
05.06.2021

Metody Obliczeniowe w Nauce i Technice
Informatyka niestacjonarna 2020/2021
Wydział Informatyki, Elektroniki i Telekomunikacji
Akademia Górniczo-Hutnicza im. Stanisława Staszica w Krakowie

1. Treść zadań

Tematem zadania będzie obliczanie różnymi sposobami całki funkcji x^2 oraz 1/sqrt(x) w przedziale (0,1)
Proszę dla obydwu funkcji:

1. Znaleźć dokładną wartość całki (całkując ręcznie)

2. Napisać program obliczającą całkę metodą prostokątów. Program powinien mieć następujące parametry:

- float a - początek przedziału
- float b - koniec przedziału
- int n - ilość podprzedziałów, na które dzielimy przedział (a,b)

3. Zbadać, przy użyciu programu z poprzedniego punktu, jak zmienia się błąd całkowania wraz ze wzrostem liczby podprzedziałów. Kiedy błąd jest mniejszy niż 1e-3, 1e-4, 1e-5 i 1e-6?

4. Obliczyć wartość całki korzystając z funkcji gsl_integration_qag metodą GSL_INTEG_GAUSS15 dla zadanych dokładności takich jak w p. 3. Sprawdzić, ile przedziałów (intervals) potrzebuje ta procedura aby osiągnąć zadaną dokładność (1e-3, 1e-4, 1e-5 i 1e-6). Porównać, ile przedziałów potrzebuje metoda prostokątów do osiągnięcia podobnej dokładności. Patrz przykład w dokumentacji GSL

Podejście do rozwiązania zadań

1. Zaczynamy od policzenia całki ręcznie

$$f(x) = x^2$$

$$\int_0^1 x^2 dx = \frac{x^3}{3}\Big|_0^1 = \frac{1}{3} - 0 = \frac{1}{3}$$

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

$$\int_0^1 \frac{1}{\sqrt{x}} dx = \int_0^1 x^{-\frac{1}{2}} dx = \frac{x^{\frac{1}{2}}}{\frac{1}{2}}\Big|_0^1 = 2\sqrt{x}\Big|_0^1 = 2 - 0 = 2$$

2. Rozpoczynamy od implementacji dwóch funkcji

```
double f1(double x) {
    return x * x;
}

double f2(double x) {
    return 1/sqrt(x);
}
```

Następnie implementujemy funkcje liczącą całke

```c
double calculateIntegral(double (*f)(double x), float a, float b, int n) {
    const double dx = (b - a) / (float) n;

    double sum = 0;

    for(int i =1; i<= n; i++) {
        double x = a + i * dx;
        sum += f(x);
    }

    return dx * sum;
}
```

Funkcja jako parametr przyjmuje wskaźnik do jednej z funkcji f(x). Jest to cała implementacja a w celu obliczenia całki wystarczy wywołać funkcje i podać jedną z f(x) (w kodzie f1, f2).

3. Aby zbadać zmienność błędu całkowania zaczynamy od implementacji konkretnej funkcji liczącej

```c
void analyzeIntegral(double (*f)(double x), double expected, const char *funcName) {
    printf("%s: \n", funcName);

    for(int i = 3; i <= 6; i++) {
        double result, error, limit = pow10((-1)*i);
        int intervals = 1;

        while((error = fabs((result = calculateIntegral(f, 0, 1, intervals++)) - expected)) >= limit) {
            printf("\rf(x): %s | value: %.18f | error %.3e <= %.1e | %d intervals. ",
                funcName,
                result,
                error,
                limit,
                intervals);

            fflush(stdout);
        }
        printf("\n");
    }
}
```

Następnie w funkcji main wywołujemy

```c
int main() {
    analyzeIntegral(f1, 1/3.0, "x^2");
    analyzeIntegral(f2, 2.0, "1/sqrt(x)");

    return 0;
}
```

Jako wynik działania programu otrzymujemy

```
x^2:
f(x): x^2 | value: 0.385000000000000010 | error 5.167e-002 <= 5.0e-002 | 11 intervals.
f(x): x^2 | value: 0.352080475537265570 | error 1.875e-002 <= 1.8e-002 | 28 intervals.
f(x): x^2 | value: 0.340120525931336790 | error 6.787e-003 <= 6.7e-003 | 75 intervals.
f(x): x^2 | value: 0.335812665424958360 | error 2.479e-003 <= 2.5e-003 | 203 intervals.
```

```
1/sqrt(x):
f(x): 1/sqrt(x) | value: 1.950208161495852600 | error 4.979e-002 <= 5.0e-002 | 841 intervals.
f(x): 1/sqrt(x) | value: 1.981683521171350600 | error 1.832e-002 <= 1.8e-002 | 6303 intervals.
```

4. Wnioski

Wartości interwałów nie są dokładne, ale przybliżone z uwagi na nieuchwycone zaokrąglenia. Wynika to ze sposobu implementacji programu. Dla funkcji $\boldsymbol{f(x)}=\mathbf{1}\sqrt{\boldsymbol{x}}$ znalezienie poprawnej liczby interwałów trwało zdecydowanie zbyt długo. W związku z powyższym, można stwierdzić, że stosowanie algorytmu obliczania całek metodą prostokątów jest nieoptymalne.