Programowanie i metody numeryczne

Zadania – seria 9.

Całkowanie.

Naszym celem jest napisanie implementacji kilku kwadratur, czyli metod obliczania przybliżonej wartości całki oznaczonej

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

z pewnej całkowalnej funkcji $y = f(x)$.

W podanych w treściach zadań deklaracjach funkcji argument `f` reprezentuje implementację całkowanej funkcji, zaś argumenty `a` i `b` odpowiadają, odpowiednio, dolnej i górnej granicy całkowania.

Zadanie 1. `intnc` – Kwadratury Newtona–Cotesa.

W celu obliczenia przybliżonej wartości całki I posłużymy się złożonymi kwadraturami Newtona–Cotesa.

Wprowadźmy $N + 1$ równoodległych punktów

$$a = x_0 < x_1 < \cdots < x_{N-1} < x_N = b$$

dzielących przedział $[a,\, b]$ na N podprzedziałów długości $H = (b - a)/N$. W podanych poniżej deklaracjach funkcji argument `N` reprezentuje liczbę N.

Napisz zestaw szablonów funkcji obliczających przybliżoną wartość całki I:

- `double IntRectangular(const auto &f, double a, double b, int N)`

  – funkcja ta powinna stosować złożony wzór prostokątów (punktu środkowego):

  $$I \approx \frac{H}{2} \sum_{k=1}^{N} f\left(\frac{x_{k-1} + x_k}{2}\right);$$

  oznacza to, że całka I jest przybliżana przez sumę pól N prostokątów, przy czym wysokość prostokąta odpowiada wartości funkcji podcałkowej w środku właściwego przedziału $[x_k,\, x_{k+1}]$,

- `double IntTrapezoidal(const auto &f, double a, double b, int N)`

  – funkcja ta powinna stosować złożony wzór trapezów:

  $$I \approx \frac{H}{2} \sum_{k=1}^{N} \left(f(x_{k-1}) + f(x_k)\right) = \frac{H}{2}\left(f(x_0) + 2f(x_1) + 2f(x_2) + 2f(x_3) + \cdots + 2f(x_{N-1}) + f(x_N)\right);$$

  oznacza to, że całka I jest przybliżana przez sumę pól N trapezów,

- `double IntSimpson(const auto &f, double a, double b, int N)`

  – funkcja ta powinna stosować złożony wzór Simpsona:

$$
\begin{aligned}
I &\approx \frac{H}{3} \sum_{k=1}^{N/2} \Big(f(x_{2k-2}) + 4f(x_{2k-1}) + f(x_{2k}) \Big) \\
&= \frac{H}{3} \Big(f(x_0) + 4f(x_1) + 2f(x_2) + 4f(x_3) + 2f(x_4) + \ldots + 2f(x_{N-2}) + 4f(x_{N-1}) + f(x_N) \Big) \\
&= \frac{H}{3} \left(f(x_0) + 4 \sum_{k=1}^{N/2} f(x_{2k-1}) + 2 \sum_{k=1}^{N/2-1} f(x_{2k}) + f(x_n) \right).
\end{aligned}
$$

Następnie, korzystając z tych szablonów napisz program `intnc`, który przyjmuje jako argumenty wywołania jedno ze słów: `rectangular`, `trapezoidal` lub `simpson`, określające metodę całkowania, oraz pewną liczbę rzeczywistą $x > 1$ i dodatnią liczbę całkowitą odpowiadającą wartości N. Program powinien dwukrotnie obliczać wartość logarytmu naturalnego dla argumentu x, za pierwszym razem obliczając zadaną metodą numeryczną całkę

$$\ln x = \int_1^x \frac{\mathrm{d}t}{t},$$

zaś za drugim razem korzystając z odpowiedniej funkcji bibliotecznej. Nastęnie program powinien wypisać na standardowe wyjście obliczone wartości i moduł ich różnicy.

Zadanie 2. `intgauss` – Kwadratury Gaussa–Legendre'a.

Tym razem do obliczenia przybliżonej wartości całki I wykorzystamy kwadratury Gaussa–Legendre'a.

Napisz szablon funkcji

```
double IntGauss(const auto &f, double a, double b, std::vector<double> q,
    std::vector<double> w)
```

obliczającej przybliżoną wartość całki I za pomocą kwadratury Gaussa–Legendre'a. Wektor `q` jest zbiorem węzłow, zaś wektor `w` – zbiorem wag.

Napisz ponadto szablon funkcji

```
double IntGaussN(const auto &f, double a, double b, int n)
```

działającej podobnie jak funkcja `IntGauss`, przyjmującej jednak zamiast zbiorów węzłów i wag liczbę naturalną ze zbioru $\{2, 3, 4, 5\}$, określającą ilość węzłów. Funkcja ta powinna zawierać w swoim ciele stałe wektory zawierające wartości węzłów i wag dla przypadku dwóch, trzech, czterech i pięciu węzłów oraz wywoływać funkcję `IntGauss` z odpowiednimi argumentami. Wartości węzłów i wag należy obliczyć lub znaleźć w literaturze.

Następnie, korzystając z tych szablonów napisz program `intgauss`, który przyjmuje jako argumenty wywołania pewną liczbę rzeczywistą $x > 1$ oraz dodatnią liczbę całkowitą ze zbioru $\{2, 3, 4, 5\}$ określającą ilość węzłów. Program powinien dwukrotnie obliczać wartość logarytmu naturalnego dla argumentu x, za pierwszym razem obliczając za pomocą kwadratury Gaussa–Legendre'a całkę

$$\ln x = \int_1^x \frac{\mathrm{d}t}{t},$$

zaś za drugim razem korzystając z odpowiedniej funkcji bibliotecznej. Nastęnie program powinien wypisać na standardowe wyjście obliczone wartości i moduł ich różnicy.

Opracowanie: Bartłomiej Zglinicki.