

METODY NUMERYCZNE

ZADANIE 11

Artur Guniewicz

11 O. Postępując się wzorem trapezów i metodą Romberga, oblicz całkę

$$I = \int_0^{\infty} \sin\left(\pi \frac{1+\sqrt{x}}{1+x^2}\right) e^{-x} dx \quad (8)$$

z dokładnością do 10^{-7} .

Wskazówka:

$$I = \underbrace{\int_0^A \sin\left(\pi \frac{1+\sqrt{x}}{1+x^2}\right) e^{-x} dx}_{I_1} + \underbrace{\int_A^{\infty} \sin\left(\pi \frac{1+\sqrt{x}}{1+x^2}\right) e^{-x} dx}_{I_{\text{ogon}}} \quad (9)$$

przy czym

$$|I_{\text{ogon}}| \leq \int_A^{\infty} \left| \sin\left(\pi \frac{1+\sqrt{x}}{1+x^2}\right) \right| e^{-x} dx \leq \int_A^{\infty} e^{-x} dx = e^{-A}. \quad (10)$$

Znajdź A takie, że $e^{-A} < 10^{-7}$, a następnie znajdź numerycznie wartość I_1 z odpowiednią dokładnością.

Metoda: Złożona metoda trapezów + metoda Romberga

Metoda Romberga opiera się na ekstrapolacji Richardsona oraz na złożonej metodzie trapezów i służy ona do przybliżania wartości całki:

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

Wzór trapezów:

$$\begin{array}{ccccccccccc} & A_{0,0} & & & & & & & & & & \\ & \swarrow & & & & & & & & & & \\ A_{0,1} & \rightarrow & A_{1,0} & & & & & & & & & \\ & \swarrow & & & & & & & & & & \\ A_{0,2} & \rightarrow & A_{1,1} & \rightarrow & A_{2,0} & & & & & & & \\ & \swarrow & & & & & & & & & & \\ A_{0,3} & \rightarrow & A_{1,2} & \rightarrow & A_{2,1} & \rightarrow & A_{3,0} & & & & & \\ & \swarrow & & & & & & & & & & \\ A_{0,4} & \rightarrow & A_{1,3} & \rightarrow & A_{2,2} & \rightarrow & A_{3,1} & \rightarrow & A_{4,0} & & & \\ & \swarrow & & & & & & & & & & \\ \dots & \rightarrow & \dots & \rightarrow & \dots & \rightarrow & \dots & \rightarrow & \dots & \rightarrow & \dots & \end{array}$$

Obliczanie kolejnych kolumn daje coraz lepsze przybliżenia.

Kolejne A liczyłem ze wzoru:

$$A_{n,k} = \frac{1}{4^n - 1} (4^n A_{n-1,k+1} - A_{n-1,k})$$

Natomiast wzór dla obliczania $A_{0,k}$ (złożona metoda trapezów) wygląda tak:

$$I_N \approx h \left(\frac{1}{2} f_0 + f_1 + f_2 + \dots + f_{N-1} + \frac{1}{2} f_N \right)$$

Kod programu:

```
/*
*****
*
*           Artur Guniewicz - Zadanie 11
*
*
*****
*/

#include <iostream>
#include <math.h> // zawiera funkcje pow, sqrt, sin, abs
#include <vector> // umożliwia tworzenie wektorów
#include <iomanip> // zawiera funkcję setprecision
#include <stdio.h>
#include <stdlib.h>

#define e 2.7182818284590452354
#define pi 3.14159265358979323846

using namespace std;

long double trapezoidArea(long double, long double, int);
long double function(long double);

int main(int argc, char const *argv[])
{
    // kontrola liczby argumentów programu
    if (argc != 4)
```

```

{
    cout << "Niepoprawna liczba argumentów wywołania programu!" <<
endl;

    cout << "Prawidlowe wywołanie programu: ./Zadanie11 <liczba
iteracji> <dolna granica całki> <przybliżenie 10^x>" << endl;

    exit(-1);
}

// dokładność 10^<argument wywołania>
int precision = stoi(argv[3]);
long double tolerance = pow(10, precision);

// szukanie A takiego, że  $e^{-A} < 10^{-7}$ 
int i = -1;
long double rest = 0;
while (true)
{
    rest = pow(e, i);

    if (rest < tolerance)
        break;

    i--;
}

cout << "A = " << -i << endl;

// górna granica całki
long double upper = -i;

// dolna granica całki
long double lower = stoi(argv[2]);

// limit iteracji
long double limitOfIters = stoi(argv[1]);

vector<long double> rombergs; // obliczanie przybliżeń do
obecnego wiersza
vector<long double> prev_rombergs; // zapamiętuje poprzednie
przybliżenia (poprzedni wiersz)

```

```

// ustawianie precyzji
cout << setprecision(-precision) << fixed;

// oblicz pierwszy wyraz
rombergs.push_back(trapezoidArea(lower, upper, 0));

cout << "k = 0 -> " << rombergs[0] << endl;

int tempK;
long double temp;
int k;

for (k = 1; k < limitOfIters - 1; k++)
{
    // zapamiętanie poprzedniego przybliżenia
    prev_rombergs = rombergs;

    // czyścimy obecny wektor
    rombergs.clear();
    cout << "k = " << k << " -> ";
    tempK = k;

    for (int n = 0; n <= k; n++)
    {
        if (n == 0)
        {
            rombergs.push_back(trapezoidArea(lower, upper, k));
            cout << rombergs[0] << " ";
            tempK--;
        }
        else
        {
            temp = (pow(4, n) * rombergs[n - 1] - prev_rombergs[n - 1]) / (pow(4, n) - 1);
            cout << temp << " ";
            rombergs.push_back(temp);
            tempK--;
        }
    }
}

```

```

    cout << endl;

    if (abs(rombergs[k] - prev_rombergs[k - 1]) < tolerance)
    {
        cout << endl << "Pole całki w przedziale [" << lower << ", "
<< upper << "] wynosi: " << rombergs[k] << " (z dokładnością do " <<
tolerance << ")" << endl;
        break;
    }
}

long double result = rombergs[k] + rest;
cout << endl << "Pole łączne całki w przedziale [0,inf) wynosi: " <<
result << " (z dokładnością do " << tolerance << ")" << endl;
return 0;
}

// korzysta ze złożonej metody trapezow by podzielic całkę na k
przedziałów
long double trapezoidArea(long double low, long double high, int k)
{
    // ilość punktów podziału
    long double N = pow(2, k);
    long double area = 0;

    // przedział pomiędzy argumentami funkcji
    long double h = (high - low) / N;

    for (long double i = low; i <= high; i = i + h)
    {
        if (i == low || i == high)
            area += function(i) / 2;

        else
            area += function(i);
    }

    return area * h;
}

```

```
// oblicza wartość funkcji podcałkowej dla danego x
long double function(long double x)
{
    return sin(pi * (1 + sqrt(x)) / (1 + pow(x, 2))) * pow(e, -x);
}
```

Kompilacja:

np. g++ Zadanie11.cpp -o Zadanie11 && ./Zadanie11 50 0 -7

Wyniki:

```
A = 17
k = 0 -> 0.0000000
k = 1 -> 0.0002891 0.0003854
k = 2 -> 0.0294521 0.0391731 0.0417589
k = 3 -> 0.2657886 0.3445568 0.3649158 0.3700452
k = 4 -> 0.2188885 0.2031511 0.1937248 0.1918067 0.1903046
k = 5 -> -0.0284941 -0.1109283 -0.1318670 -0.1370351 -0.1383215 -0.1386427
k = 6 -> -0.1370753 -0.1732690 -0.1774251 -0.1781482 -0.1783095 -0.1783485 -0.1783582
k = 7 -> -0.1871272 -0.2038111 -0.2058472 -0.2062984 -0.2064088 -0.2064362 -0.2064431 -0.2064448
k = 8 -> -0.2063470 -0.2127536 -0.2133497 -0.2134688 -0.2134969 -0.2135039 -0.2135056 -0.2135060 -0.2135061
k = 9 -> -0.2133668 -0.2157068 -0.2159037 -0.2159442 -0.2159539 -0.2159563 -0.2159569 -0.2159571 -0.2159571
k = 10 -> -0.2158856 -0.2167252 -0.2167931 -0.2168073 -0.2168106 -0.2168115 -0.2168117 -0.2168117 -0.2168118 -0.2168118
k = 11 -> -0.2167825 -0.2170815 -0.2171052 -0.2171102 -0.2171113 -0.2171116 -0.2171117 -0.2171117 -0.2171117 -0.2171117
k = 12 -> -0.2171067 -0.2172068 -0.2172151 -0.2172169 -0.2172173 -0.2172174 -0.2172174 -0.2172174 -0.2172174 -0.2172174
k = 13 -> -0.2172134 -0.2172510 -0.2172539 -0.2172545 -0.2172547 -0.2172547 -0.2172547 -0.2172547 -0.2172547 -0.2172547
k = 14 -> -0.2172533 -0.2172666 -0.2172676 -0.2172678 -0.2172679 -0.2172679 -0.2172679 -0.2172679 -0.2172679 -0.2172679
k = 15 -> -0.2172674 -0.2172721 -0.2172724 -0.2172725 -0.2172725 -0.2172725 -0.2172725 -0.2172725 -0.2172725 -0.2172725
k = 16 -> -0.2172724 -0.2172740 -0.2172742 -0.2172742 -0.2172742 -0.2172742 -0.2172742 -0.2172742 -0.2172742 -0.2172742
k = 17 -> -0.2172741 -0.2172747 -0.2172748 -0.2172748 -0.2172748 -0.2172748 -0.2172748 -0.2172748 -0.2172748 -0.2172748
k = 18 -> -0.2172747 -0.2172750 -0.2172750 -0.2172750 -0.2172750 -0.2172750 -0.2172750 -0.2172750 -0.2172750 -0.2172750
k = 19 -> -0.2172750 -0.2172750 -0.2172750 -0.2172750 -0.2172750 -0.2172750 -0.2172750 -0.2172750 -0.2172750 -0.2172750

Pole całki w przedziale [0.0000000,17.0000000] wynosi: -0.2172750 (z dokładnością do 0.0000001)
Pole łączne całki w przedziale [0,inf) wynosi: -0.2172750 (z dokładnością do 0.0000001)
```

Komentarz:

Każde kolejne $A_{0,k}$ jest coraz bliższe dokładnemu wyniku.

Naszej całki nie jesteśmy w stanie od razu obliczyć numerycznie ponieważ ma nieskończoną górną granicę. Należy więc ją rozbić na dwie całki:

$$\int_0^{\infty} f(x) dx = \int_0^A f(x) + \int_A^{\infty} f(x)$$

Następnie szukamy takiego A, żeby:

$$\int_A^{\infty} \left| \sin\left(\frac{1 + \sqrt{x}}{1 + x^2}\right) \right| dx < \int_A^{\infty} e^{-x} dx$$

Dokładność z zadania wynosi 10^{-7} , a więc A, takie żeby prawa całka nie wpływała na wynik całości to $A = 17$.