

Wojciech Ładyga - zadanie 14

Język technologia: c++

Aby obliczyć całkę metodą Romberga należy posłużyć się wzorem trapezów. W mojej implementacji metoda Romberga wykorzystuje wzór trapezów przekazując wartość A (granica górna całki) oraz wysokość trapezu $h = (b-a)/2$.

Metoda trapezów daje nam wzór:

$$h := \frac{b-a}{n}, \quad x_i = a + (i-1)h, \quad i = 1 \dots n+1.$$

$$S_n = \frac{y_1 + y_2}{2}h + \frac{y_2 + y_3}{2}h + \dots + \frac{y_n + y_{n+1}}{2}h = h \left(\frac{y_1}{2} + y_2 + y_3 + \dots + y_n + \frac{y_{n+1}}{2} \right)$$

następnie musiałem ją dostosować do naszego przypadku.

Lecz funkcja Romberga generuje wysokość h i przekazuje do metody trapezów i dana jest wzorem

$$h / \text{pow}(2, i-1) \text{ lub } h / \text{pow}(4, i-1)$$

w zależności od konfiguracji danych. Jest tak ponieważ metoda Romberga wykorzystuje 2^k podziały. Co oznacza, że wykorzystuje parzyste potęgi średnicy podziału.

Kod programu:

```
/*
 * @Author: Wojciech Ładyga
 * @Date: 2018-12-25
 * @Description: Zad 14
 */
#include <iostream>
#include <iomanip> //do setprecision
#include <cmath>   //biblioteka matematyczna c++
#include <algorithm>
using namespace std;

//przydatne linki
/*
    -wykłady
    //trapezy
    -https://www.p-programowanie.pl/cpp/calcowanie-numeryczne/
    -http://www.algorytm.edu.pl/algorytmy-maturalne/metoda-trapezow.html
    //Romberg
    -https://pl.wikipedia.org/wiki/Metoda_Romberga
    -https://en.wikipedia.org/wiki/Romberg%27s_method
    -https://www.wolframalpha.com/input/?
```

```

i=int+sin(pi(1%2B(x%5E(1%2F2)))%2F(1%2Bx%5E2))*exp(-x)+x%3D0..17
-http://th-www.if.uj.edu.pl/zfs/gora/metnum18/wyklad07.pdf
-
https://ps.uci.edu/~cyu/p231C/LectureNotes/lecture10:integration/html_version/

*/

const double epsilon = 0.0000001; //zadana dokladnosc
const int kroki = 22;             //ilosc krokow w metodzie Romberga

double f(double x);
double trapez(double A, double h);
double romberg(double A);
double znajdzA();

class Romberg
{
public:
    //funkcja szukajaca A
    double znajdzA()
    {
        double A = 0.0;
        //wiemy ze  $\exp(-A) < 10^{-7}$  wiec mozemy wyliczyc A
        while (exp(-A) > epsilon)
        {
            A++;
        }
        return A;
    }

    double romberg(double A)
    {
        double b = A, a = 0;
        double h = (b - a);
        double tab[kroki];
        fill(tab + 1, tab + kroki, 1.0); //wartosci kontrolne
        double tmp;

        for (int i = 1; i < kroki; i++)
        {
            for (int j = 1; j <= i; j++)
            {
                if (j == 1.0)
                {
                    tmp = tab[j];
                    tab[j] = trapez(A, h / pow(2, i - 1));
                }
                else
                {
                    tab[i] = (pow(4, j - 1) * tab[j - 1] - tmp) / (pow(4, j - 1) -
1);

```

```

        tmp = tab[j];
        tab[j] = tab[i];
    }
}

return tab[kroki - 1]; //zwracamy przedostatni el naszych obliczeń
}

private:
//funkcja do pokałkowania
double f(double x)
{
    return sin((M_PI + M_PI * sqrt(x)) / (1.0 + pow(x, 2))) * exp(-x);
}

double trapez(double A, double h)
{
    double Sn = 0.0;
    //h = (b - a)/n
    //Sn = (y1+y2)*2/h + (y2+y3)*2/h + ... +

    for (double i = h; i < A; i += h)
    {
        Sn += f(i);
    }
    //należy dodać przedziały
    Sn += f(0) / 2;
    Sn += f(A) / 2;

    Sn *= h;

    return Sn;
}

};

int main()
{
    //klasa licząca metodę Romberga z trapezami
    Romberg rb;
    cout << "A: " << rb.znajdzA() << endl;
    cout << "I1: " << setprecision(7) << fixed << rb.romberg(znajdzA()) << endl;

    return 0;
}

```

Wyniki działania programu to:

```
A: 17  
I1: -0.2172751
```