Savchenko Yehor, Metody Numeryczne, zadanie NUM 8

**Instrukcja do programy:**

1. Zainstalować [g++](https://www3.cs.stonybrook.edu/~alee/g++/g++.html);
2. Wpisać do terminalu „**make”** albo „**make all”**
3. Zapuścić program „ **./Yehor\_Savchenko\_NUM8.x** ”

**Wstęp:**

Musiałem zaimplementować dwie procedury biblioteczne obliczające całkę na przedziale **[a, b]** z tolerancją bezwzględną ε, bazujące na

* Złożonej kwadratury Newtona-Cotesa z n = 2 (wzór Simpsona)
* Metodzie Romberga.

Do obliczania miałem dwie całki (a) i (b) . Należało obliczyć z dokładnością do 10 miejsc po przecinku.

**Kwadratura złożona Newtona-Cotesa** - należy do metod z ustalonymi węzłami, polega na tym, że funkcja f(x) jest interpolowana wielomianem. Przedział całkowania dzielimy na **m** przedziałów i do każdego z nich stosujemy kwadraturę Simpsona:

)

h =

E =

**Metoda Romberga** – jest rozszerzeniem metody trapezów i daje lepsze przybliżenie całki poprzez zasadniczą redukcję błędu. Całkowanie metodą Romberga stosuje ten sam wzór

co ekstrapolacja Richardsona. Jednakże, metoda Romberga jest to algorytm rekurencyjny

Ogólne wyrażenie w metodzie Romberga:

Gdize **k** reprezentuje rząd ekstrapolacji:

k=1 odpowiada wartościom uzyskanym ze wzoru trapezów

k=2 odpowiada wartościom uzyskanym z błędem O(h­­­2)

Wskaźnik **j** reprezentuje dokładność; **j+1** daje całkę wyznaczoną dokładniej niż **j**.

**Wyniki:**

Newtona-Cotesa = 0.4596976941

Newtona-Cotesa = 0.4266771075

Romberg = 0.4596976941

Romberg = 0.4266771075

**Przedyskutowanie wyników:**

Z wyników i mojej obserwacji działania programy wyszło tak , że metoda Romberga zbiega szybciej (dla całki pojedynczej 4 iteracji, dla całki podwójnej 6 iteracji) niż metoda Newtona-Cotesa.