# Metody Obliczeniowe i Symulacja, Laboratorium 5

## Całkowanie Numeryczne (cz. I i II), Kamil Klimowicz, 2020-12-05

### Zadania

Tematem zadania będzie obliczanie różnymi sposobami całki funkcji x^2 + x + 1 oraz 1/sqrt(x)  
w przedziale (0,1).

Cz. I:

1. Znaleźć dokładną wartość całki (całkując ręcznie)
2. Napisać program obliczający całkę metodą prostokątów. Program powinien mieć następujące parametry:
   1. float a - początek przedziału
   2. float b - koniec przedziału
   3. 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?

Cz. II:

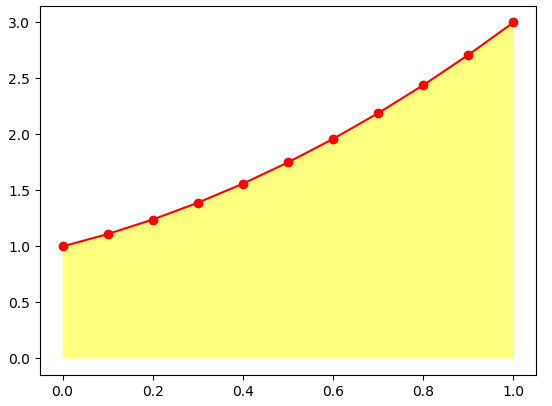
1. Obliczyć wartość całki korzystając z funkcji **gsl\_integration\_qag** metodą GSL\_INTEG\_GAUSS15 dla zadanych dokładności takich jak w p. 3. Sprawdzić, ile przedziałów (*intervals*) potrzebuje ta procedura, aby osiągnąć zadaną dokładność (1e-3, 1e-4, 1e-5 i 1e-6). Porównać, ile przedziałów potrzebuje metoda prostokątów do osiągniecia podobnej dokładności. Patrz przykład w [dokumentacji GSL](http://www.gnu.org/software/gsl/manual/html_node/Numerical-integration-examples.html).
2. Wykorzystać funkcję **gsl\_integration\_qag()** do obliczenia całki  z wartościami błędów mniejszymi niż 1e-3, 1e-4, 1e-5 i 1e-6 dla funkcji:
   1. sin(x) na przedziale [0, pi]
   2. tan(x) na przedziale (0, pi/2]
   3. log(x+x^2) na przedziale [1,4]

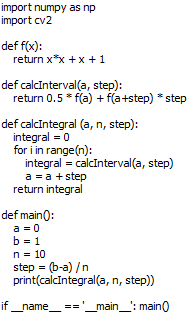
### Podejście do rozwiązania

Rozwiązania zostały wywołane/zaimplementowane w środowisku operacyjnym Linux Mint, Windows, wykorzystując języki skrypty odpowiednio przygotowane w C oraz Pyton, pozwalające na numeryczne obliczenia równań całkowych. Przygotowane pliki zostały skompilowane w ww systemach operacyjnych.

### Wyniki

Cz. I:

1. Dokładna wartość całki wynosi:
2. Program realizujący całkowanie funkcji z zadania:



Rys.1. Kod źródłowy. Rys.2. Wykres argumentów i wartości funkcji.

1. Analiza błędu całkowania została zamieszczona w poniższej tabeli.

W analizie została założona dokładna wartość całki S(f) wyznaczona ręcznie: wyrażona  
w zmiennej typu float. Błąd przybliżenia został wyrażony w postaci różnicy wartości dokładnej i wynikowej: .

Tabela 1. Wyniki błędu całkowania dla różnych wielkości podziałów funkcji całkowanej.

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| n | 2 | 4 | 6 | 8 | 10 | 20 | 50 |
| Err | -0.5417 | -0.0729 | 0.0694 | 0.1380 | 0.1783 | 0.2571 | 0.3031 |
| 1e-3=0.0498 1e-4=0.0182 1e-5=0.0069 1e-6=0.0025 | | | | | | | |

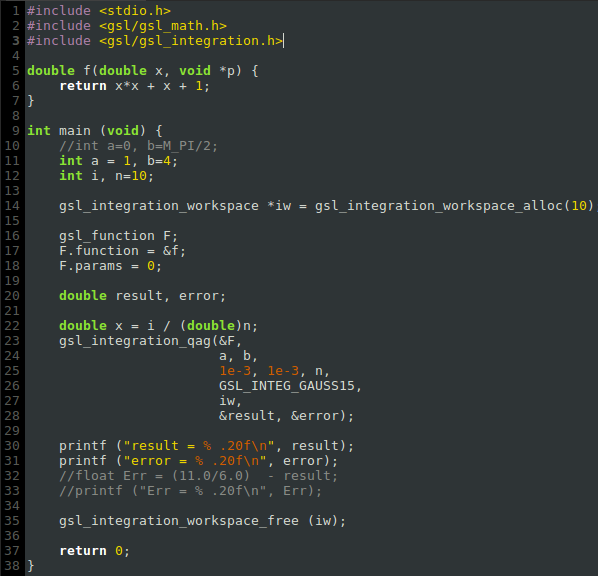
Legenda:  
 n – ilość przedziałów  
 Err – błąd przybliżenia

Zaimplementowane rozwiązanie nie pozwala na uzyskanie dokładności poniżej e-6. Najlepsza wartość przybliżenia wynosi e-2. Co więcej wraz ze wzrostem liczby przedziałów, wartość błędu rośnie. Jest to wynikiem niedokładności powodowanej przez zwiększającą się liczbę iteracji przy rosnącej liczbie przedziałów.

Cz. II:

1. Wyniki całkowania funkcji zrealizowane z wykorzystaniem biblioteki **gsl\_integration\_qag.**

Wykorzystana zasada kwadratury funkcji Gauss’a-Kronrod’a rzędu 1, zapewnia uzyskanie wyników obarczonych błędem poniżej e-10. Znakomita rezultat jest uzyskany bez względu na przyjęty stopień podziału, gdyż przyjęta metoda wykorzystuje algorytm adaptacyjny.

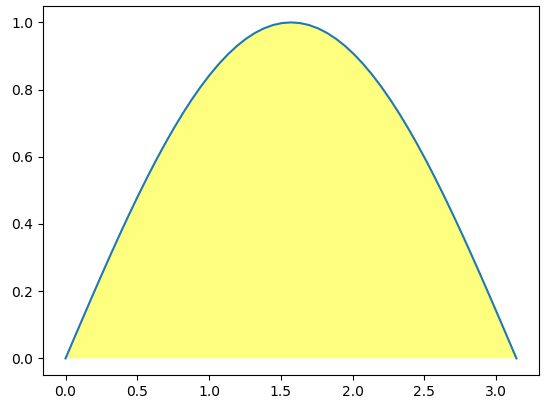
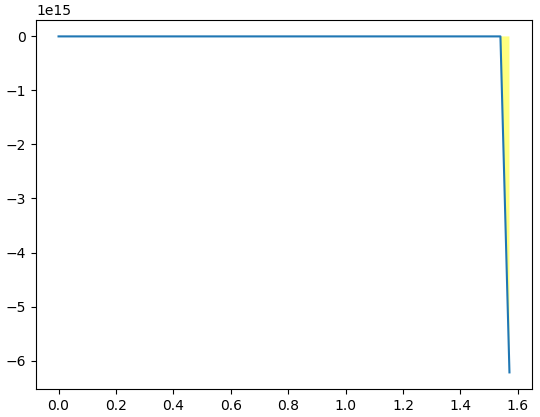
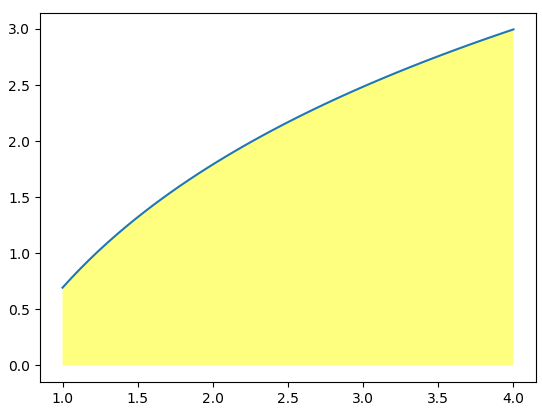


Rys.3. Implementacja biblioteki gsl\_integration\_qag.

1. Poniżej przedstawiono wyniki całkowania funkcji trygonometrycznych i logarytmicznej wykorzystując bibliotekę gsl\_integration\_qag.

Tabela 2. Funkcje całkowane i przyjęte wartości błędu ograniczenia.

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
|  | 1e-3 | 1e-4 | 1e-5 | 1e-6 |
| sin(x) ∈ [0,pi] | 0.00000000000018318680 | | | |
| tan(x) ∈ (0,pi/2] | 0.00000000163700368514 | | | |
| log(x+x2) ∈ [1,4] | 0.00000000163700368514 | | | |

Rys.5. Funkcje całkowe z tabeli 2 dla zadanej dziedziny.

### Wnioski

Metoda całkowania funkcji przez numeryczną analizę pola powierzchni funkcji dla zadanej dziedziny może wydawać się mało dokładna. Jednakże rozważając trudności w implementacji (a czasami wręcz całkowity jej brak) pochodnej funkcji podcałkowej, widzimy, że wspomniana metoda może okazać się zdecydowanie skuteczniejszą i jedyną dostępną opcją (szczególnie dla funkcji nieciągłych w całej dziedzinie).

Analizując stopień dokładności uzyskiwanych wyników w stosunku do metody klasycznej, widzimy, że popełniany błąd jest zależy od metody implementacji rozwiązania. Biblioteka GNU znakomicie rozwinięte metody adaptacyjne obliczania równań całkowych (szczególnie odnosząc się do własnej implementacji z wykorzystaniem aproksymacji trapezoidalnej).

### Bibliografia

[1] <https://artemis.wszib.edu.pl/~funika/mois/lab5/>

[2] Wykład Systemy i metody obliczeniowe, Dr inż. Włodzimierz Funika

[3] <http://www.gnuplot.info/>

[4] <https://www.wolframalpha.com>

[5] <http://wazniak.mimuw.edu.pl>