Data: 23.04.2025

Laboratorium 7

Autorzy: Kosman Mateusz, Ludwin Bartosz

Temat laboratorium:

Celem niniejszego laboratorium jest porównanie skuteczności adaptacyjnych metod całkowania numerycznego na przykładzie całki $\int_0^1 \frac{4}{1+x^2} dx$. Zostaną wykorzystane dwie metody: adaptacyjna kwadratura trapezów oraz adaptacyjna kwadratura Gaussa-Kronroda. Porównamy ich dokładność, liczbę ewaluacji funkcji oraz wpływ ustawień tolerancji na wyniki. W drugiej części sprawdzimy te same metody na trudniejszych całkach – zawierających logarytm oraz funkcje z osobliwościami – i porównamy wyniki z wartościami analitycznymi.

Treść:

Zadanie 1.

Oblicz wartość całki z poprzedniego laboratorium

$$\int_0^1 \frac{4}{1+x^2} dx \tag{1}$$

korzystając z:

(a) kwadratur adaptacyjnych trapezów,

(b) kwadratur adaptacyjnych Gaussa-Kronroda.

Dla każdej metody narysuj wykres wartości bezwzględnej błędu względnego w zależności od liczby ewaluacji funkcji podcałkowej. Wyniki dodaj do wykresu uzyskanego w poprzednim laboratorium. Przydatna będzie funkcja `scipy.integrate.quad_vec`. Na liczbę ewaluacji funkcji podcałkowej można wpływać pośrednio, zmieniając wartość dopuszczalnego błędu (tolerancji). Przyjmij wartości tolerancji z zakresu od 10^0do 10^{-14}. Liczba ewaluacji funkcji podcałkowej zwracana jest w zmiennej `info.neval`.

Zadanie 2.

Powtórz obliczenia z poprzedniego oraz dzisiejszego laboratorium dla całek:

$$\text{(a)} \qquad \int_0^1 \sqrt{x}\, log(x) dx = -\frac{4}{9}, \qquad (2)$$

$$\text{(b)} \qquad \int_0^1 \left(\frac{1}{(x-0.3)^2+a} + \frac{1}{(x-0.9)^2+b} - 6\right) dx. \qquad (3)$$

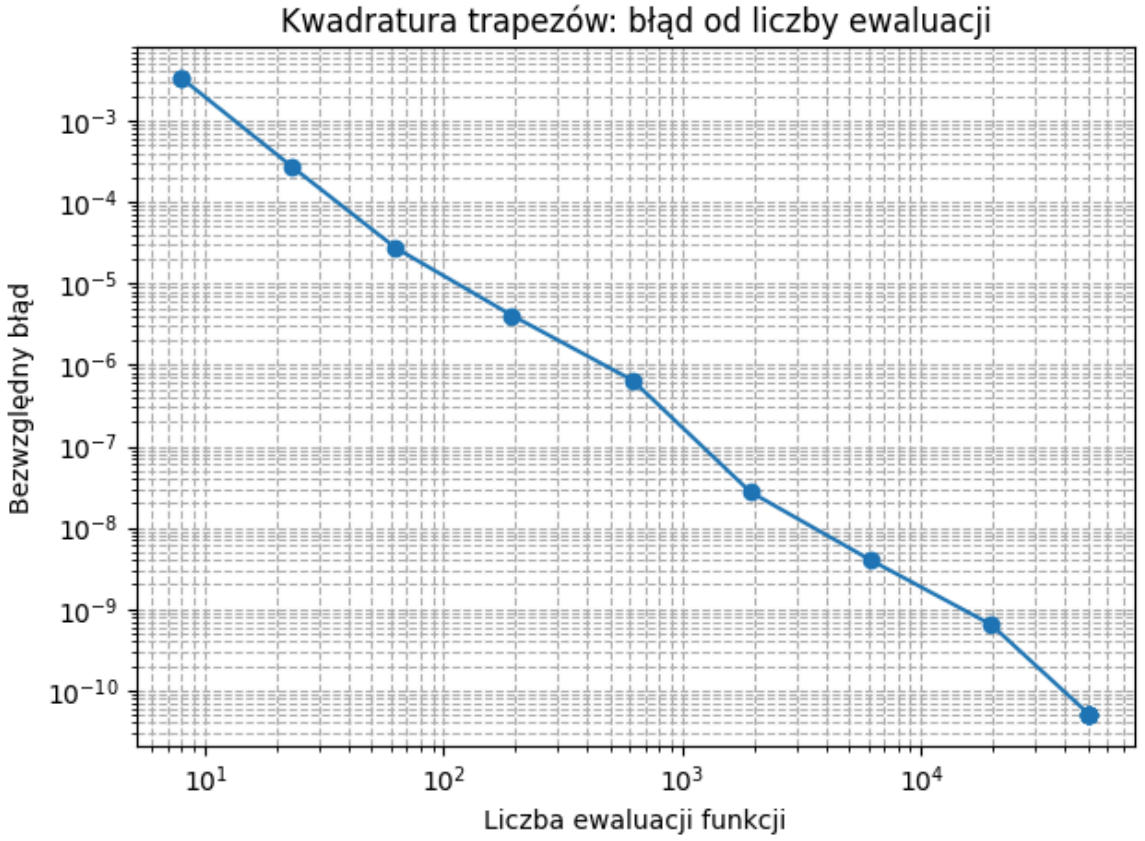
We wzorze (3) przyjmij $a = 0.001$ oraz $b = 0.004$. Błąd kwadratury dla całki (3) oblicz, wykorzystując fakt, że:

$$\int_0^1 \frac{1}{(x-x_0)^2+a} dx = \frac{1}{\sqrt{a}} \left(arctg \frac{1-x_0}{\sqrt{a}} + arctg \frac{x_0}{\sqrt{a}}\right) \qquad (4)$$

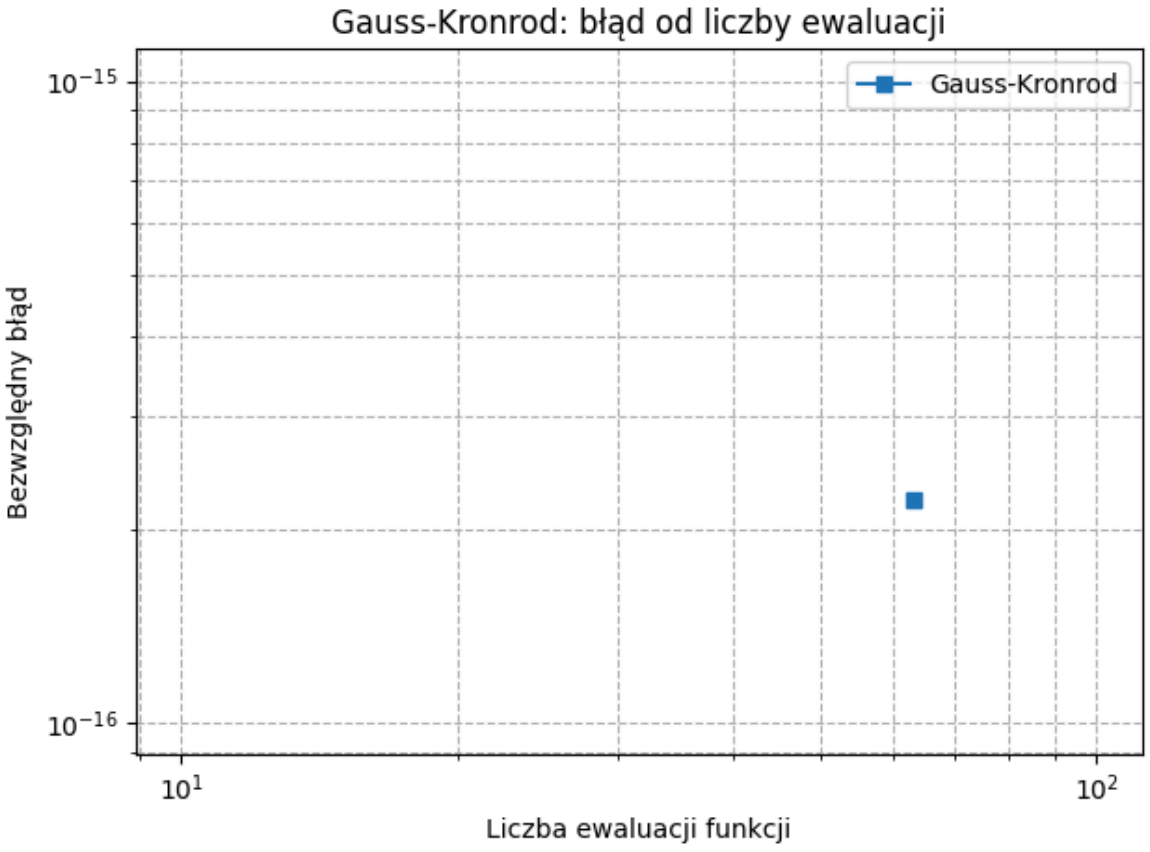
Rozwiązanie:

Zadanie 1.

Do wykonania zadania użyliśmy funkcji $quad\_vec$ z modułu $scipy$ odpowiednio z parametrami `quadrature='trapezoid'` dla adaptacyjnej kwadratury trapezów, oraz `quadrature='gk21'` dla adaptacyjnej kwadratury Gaussa-Kronroda. Wykonaliśmy je po 15 razy stosując parametr `epsrel` w zakresie od 1 do 10^{-14} z skokiem o jeden rząd wielkości. Rezultaty tych metod użyliśmy do obliczenia błędu względnego, a następnie zapisaliśmy go wraz z ilością ewaluacji potrzebnych do uzyskania wyniku. Dane przygotowaliśmy odfiltrowując z nich wartości równe 0 (z powodu użycia skali logarytmicznych na osiach zero nie mogło być wśród wyników), a następnie nałożyliśmy je na wykresy otrzymując:

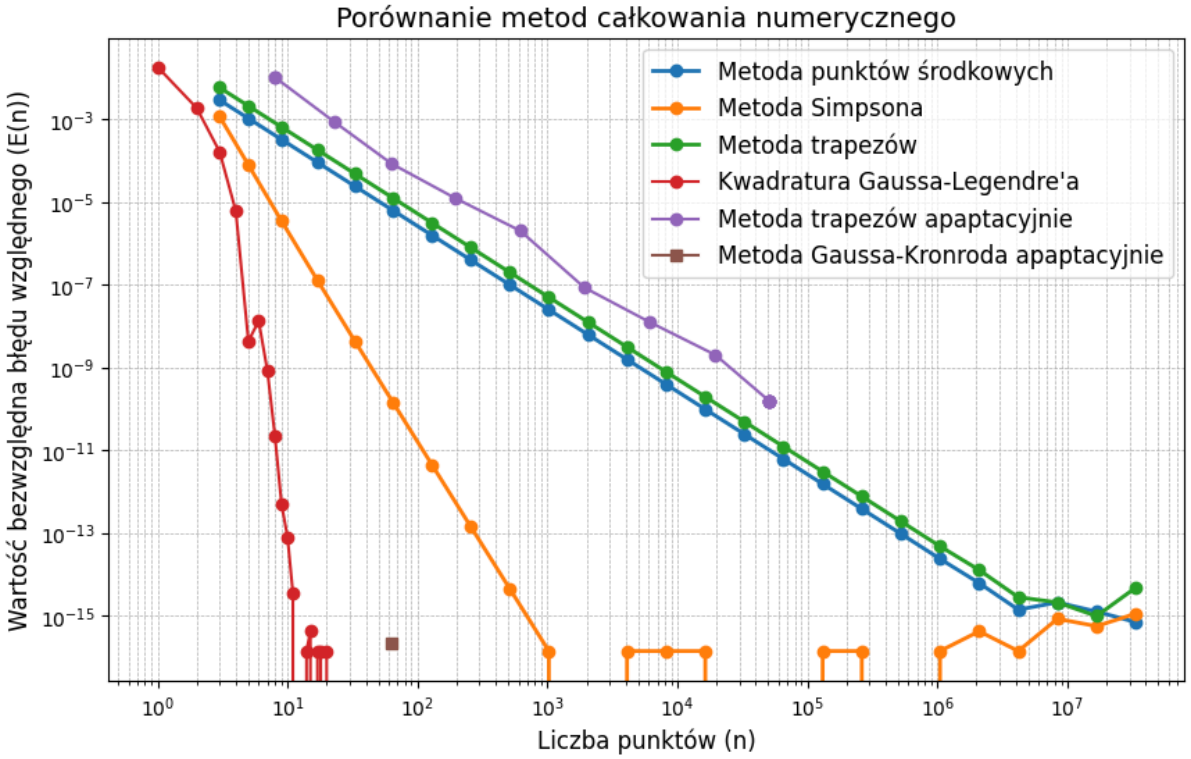


Wykres 1



Wykres 2

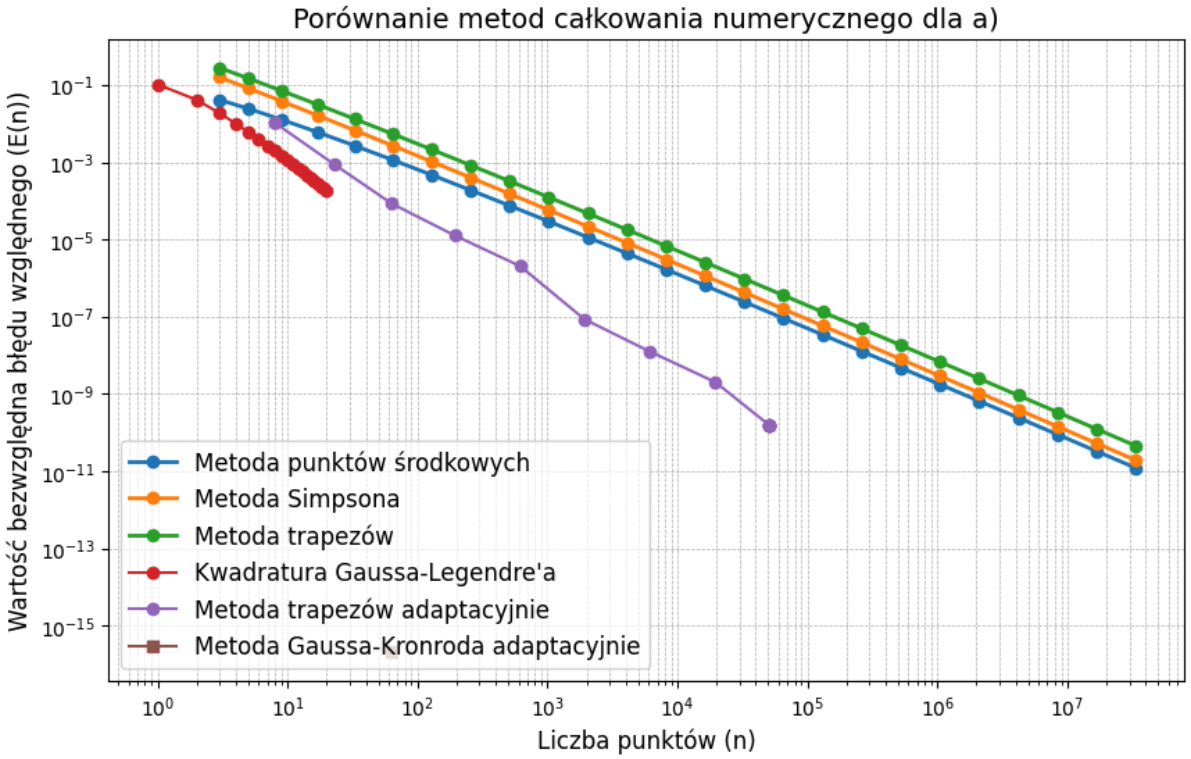
Następnie dodaliśmy otrzymane wyniki na wykres z poprzedniego laboratorium.



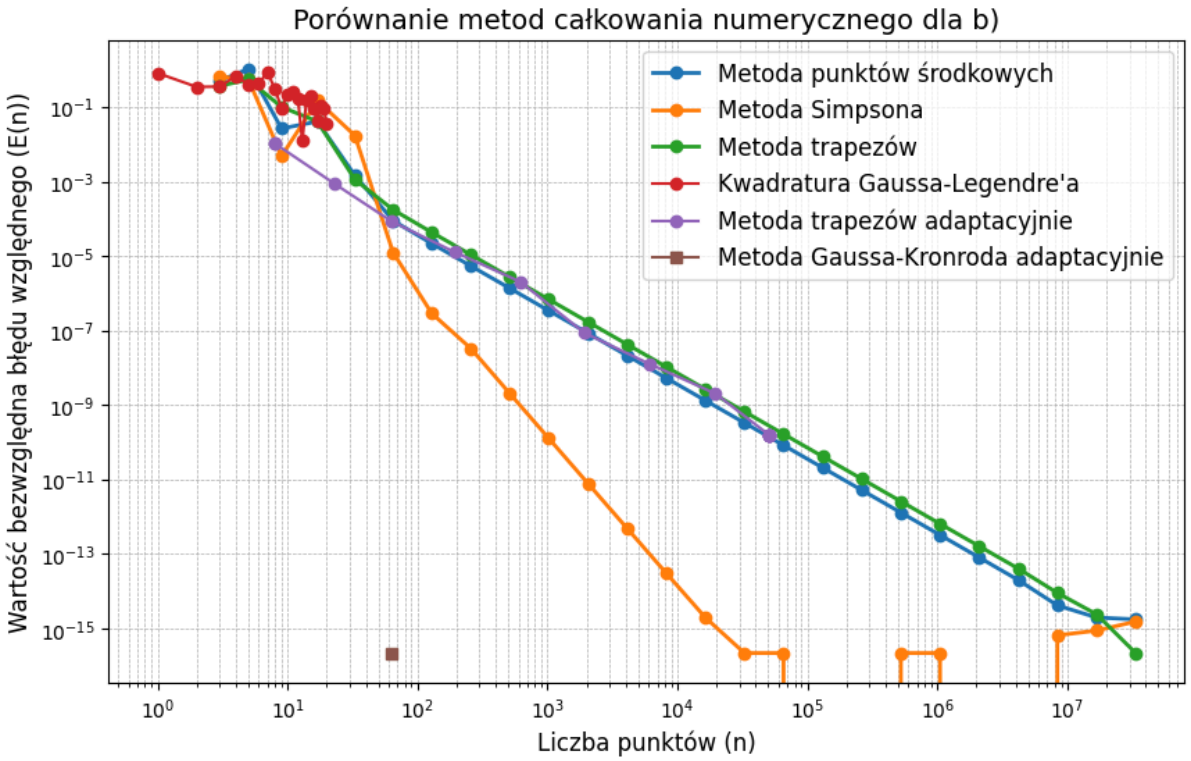
Wykres 3

Analizując wykres można zauważyć że adaptacyjna metoda trapezów charakteryzuje się większym błędem względnym. Dzieje się tak prawdopodobnie ponieważ adaptacja przeinwestowywuje błąd w miejscach gdzie nie jest to potrzebne. Z kolei metoda Gaussa-Kronroda okazała się być bardzo dokładna. Dla stosunkowo małej ilości punktów otrzymała błąd rzędu 10^{-14}, a nawet niektóre ewaluację miały błąd równy 0.

Zadanie 2.



Wykres 4 - Porównanie metod całkowania numerycznego dla wzoru a)



Wykres 5 - Porównanie metod całkowania numerycznego dla wzoru b)

W tym zadaniu dokonaliśmy tego samego porównania metod całkowania, co w zadaniu 1, lecz z zastosowaniem kolejno wzoru z podpunktu a) oraz z podpunktu b) tego zadania.

Wnioski:

- Kwadratura adaptacyjna Gaussa-Kronroda daje znacznie mniejszą wartość bezwzględną błędu względnego dla n rzędu $\sim 10^2$ niż pozostałe metody całkowania numerycznego, w tym z kwadraturą adaptacyjną trapezów.
- Kwadratury adaptacyjne sprawdzają się lepiej w przypadku skomplikowanych, oscylujących funkcji. W przypadku prostych funkcji może się okazać że błąd nieadaptacyjnej metody zbiega do zera szybciej.
- Zbyt restrykcyjny parametr tolerancji ($epsrel$) prowadzi do nadmiernego wzrostu liczby ewaluacji bez istotnego zmniejszenia błędu (ograniczenia maszynowe).

Źródła:

- Prezentacje z MS Teams (zespół MOwNiT 2025)
- pythonnumericalmethods.studentorg.berkeley.edu