Laboratorium 7: Kwadratury Adaptacyjne

Jan Sarba, Dariusz Rozmus

20 maja 2025

1 Treść zadań

1.1 Zadanie 1

Oblicz wartość całki z poprzedniego laboratorium:

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

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 10^0 do 10^{-14}. Liczba ewaluacji funkcji podcałkowej zwracana jest w zmiennej `info.neval`.

1.2 Zadanie 2

Powtórz obliczenia z poprzedniego oraz dzisiejszego laboratorium dla całek:

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

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

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

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

2 Wstęp Teoretyczny

Całkowanie numeryczne, zwane również kwadraturą numeryczną, jest zbiorem technik służących do aproksymacji wartości całki oznaczonej. Jest to niezbędne, gdy funkcja podcałkowa nie ma znanej funkcji pierwotnej wyrażalnej za pomocą funkcji elementarnych, lub gdy funkcja jest znana tylko w dyskretnych punktach.
Podstawowe metody kwadratur, takie jak reguła prostokątów, trapezów czy Simpsona (metody Newtona-Cotesa) oraz kwadratury Gaussa, dzielą przedział całkowania na ustaloną liczbę podprzedziałów i stosują na nich odpowiednie formuły. Choć skuteczne dla funkcji "gładkich", mogą wymagać bardzo dużej liczby podprzedziałów (a tym samym ewaluacji funkcji) dla funkcji o nieregularnym zachowaniu, np. z ostrymi pikami lub osobliwościami.
Kwadratury adaptacyjne stanowią rozwinięcie tych metod, dążąc do osiągnięcia zadanej dokładności przy minimalnym wysiłku obliczeniowym. Ich główna idea polega na nierównomiernym podziale przedziału całkowania – gęściej tam, gdzie funkcja zmienia się gwałtownie, a rzadziej tam, gdzie robi to powoli. Algorytm adaptacyjny estymuje błąd aproksymacji na danym podprzedziale. Jeśli błąd jest zbyt duży, podprzedział jest dzielony na mniejsze części, na których procedura jest powtarzana rekurencyjnie. Jeśli błąd jest akceptowalny, wartość całki na tym podprzedziale jest dodawana do sumy całkowitej.
W ramach niniejszego laboratorium wykorzystane zostały dwie metody kwadratur adaptacyjnych:

- **Adaptacyjna kwadratura trapezów**: Jest to metoda rekurencyjna. Dla danego przedziału $[a, b]$ oblicza się przybliżenie całki S_{ab} za pomocą reguły trapezów. Następnie przedział dzieli się na dwa podprzedziały $[a, c]$ i $[c, b]$ (gdzie $c = (a + b)/2$) i oblicza sumę przybliżeń $S_{ac} + S_{cb}$. Różnica $|S_{ab} - (S_{ac} + S_{cb})|$ służy jako estymata błędu. Jeśli błąd przekracza zadaną tolerancję (skalowaną odpowiednio do długości podprzedziału), procedura jest stosowana rekurencyjnie do podprzedziałów.

- **Adaptacyjna kwadratura Gaussa-Kronroda**: Metody Gaussa-Kronroda wykorzystują parę formuł kwadraturowych: regułę Gaussa niższego rzędu (G_n) i regułę Kronroda wyższego rzędu (K_{2n+1}), która ponownie wykorzystuje n węzłów z reguły Gaussa. Różnica między wartościami całek uzyskanymi z tych dwóch reguł ($|K_{2n+1} - G_n|$) dostarcza efektywnej estymaty błędu. Podobnie jak w metodzie trapezów, jeśli błąd jest zbyt duży, przedział jest dzielony, a procedura powtarzana. Funkcja `scipy.integrate.quad_vec` wykorzystuje ten schemat.

Kwadratury adaptacyjne są szczególnie użyteczne dla funkcji, których trudne do całkowania fragmenty są zlokalizowane w niewielkich częściach całego przedziału całkowania.

3 Argumentacja

W celu realizacji zadań laboratoryjnych wykorzystano język Python wraz z bibliotekami NumPy, Matplotlib oraz SciPy.
Implementacja metod z poprzedniego laboratorium (Lab6): Dla celów porównawczych, na wykresach umieszczono również wyniki uzyskane metodami z poprzedniego laboratorium:

- Reguła prostokątów (Mid-point Rule)
- Reguła trapezów (Trapezoidal Rule)
- Reguła Simpsona (Simpson's Rule)
- Kwadratura Gaussa-Legendre'a

Liczba ewaluacji funkcji dla tych metod zależy od liczby podprzedziałów $n = 2^m$ (dla metod Newtona-Cotesa) lub liczby węzłów kwadratury (dla Gaussa-Legendre'a).
Adaptacyjna kwadratura trapezów (Lab7): Została zaimplementowana funkcja `recursive_adaptive_trapezoid`.

- Funkcja przyjmuje jako argumenty funkcję podcałkową opakowaną w klasę `FuncCounter` (służącą do zliczania liczby ewaluacji funkcji), granice całkowania (a, b), docelową tolerancję `tol`, początkowe przybliżenie całki S_{ab} oraz wartości funkcji na krańcach f_a, f_b.
- W każdym kroku rekurencji, punkt środkowy $c = (a + b)/2$ jest obliczany, a funkcja jest ewaluowana w c (co zwiększa licznik ewaluacji).
- Obliczane są przybliżenia całki na podprzedziałach S_{ac} i S_{cb}. Suma $S_{acb} = S_{ac} + S_{cb}$ jest porównywana z S_{ab}.
- Jeśli $|S_{acb} - S_{ab}| < \text{tol}$, zwracana jest wartość S_{acb}.
- W przeciwnym razie, funkcja jest wywoływana rekurencyjnie dla lewego podprzedziału (a, c) z tolerancją tol/2 oraz dla prawego podprzedziału (c, b) z tolerancją tol/2.
- Wprowadzono maksymalną głębokość rekursji (`max_depth`) oraz globalny limit liczby ewaluacji funkcji (`global_max_evals`) dla pojedynczego wywołania całkowania z daną tolerancją, aby zapobiec zbyt długiemu działaniu dla trudnych przypadków.

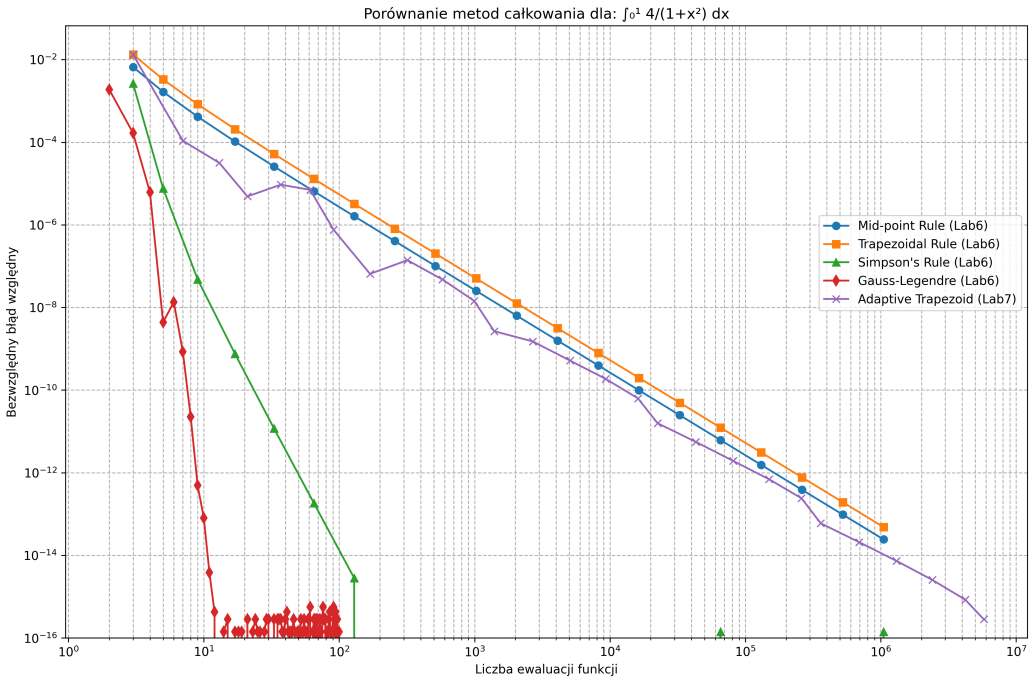
Liczba ewaluacji funkcji oraz błąd względny były zapisywane dla różnych wartości tolerancji z zakresu 10^0 do 10^{-14}.
Adaptacyjna kwadratura Gaussa-Kronroda (Lab7): Wykorzystano funkcję `scipy.integrate.quad_vec` z biblioteki SciPy.

- Funkcja ta implementuje adaptacyjny algorytm oparty na parach kwadratur Gaussa-Kronroda (domyślnie para (15,7)-punktowa, ale może używać innych).
- Tolerancja błędu jest kontrolowana przez parametry `epsabs` (tolerancja absolutna) i `epsrel` (tolerancja względna); w laboratorium dla obu ustawiono tę samą wartość `tol`.
- Funkcja zwraca krotkę, z której obiekt `info_gk` zawiera pole `neval` przechowujące liczbę ewaluacji funkcji podcałkowej.
- Podobnie jak dla adaptacyjnej kwadratury trapezów, testowano różne wartości tolerancji, a także wprowadzono heurystyczny limit na maksymalną liczbę ewaluacji (`max_evals_adaptive_gk`) dla pojedynczego wywołania z daną tolerancją, przerywając pętlę po tolerancjach, jeśli limit został przekroczony.

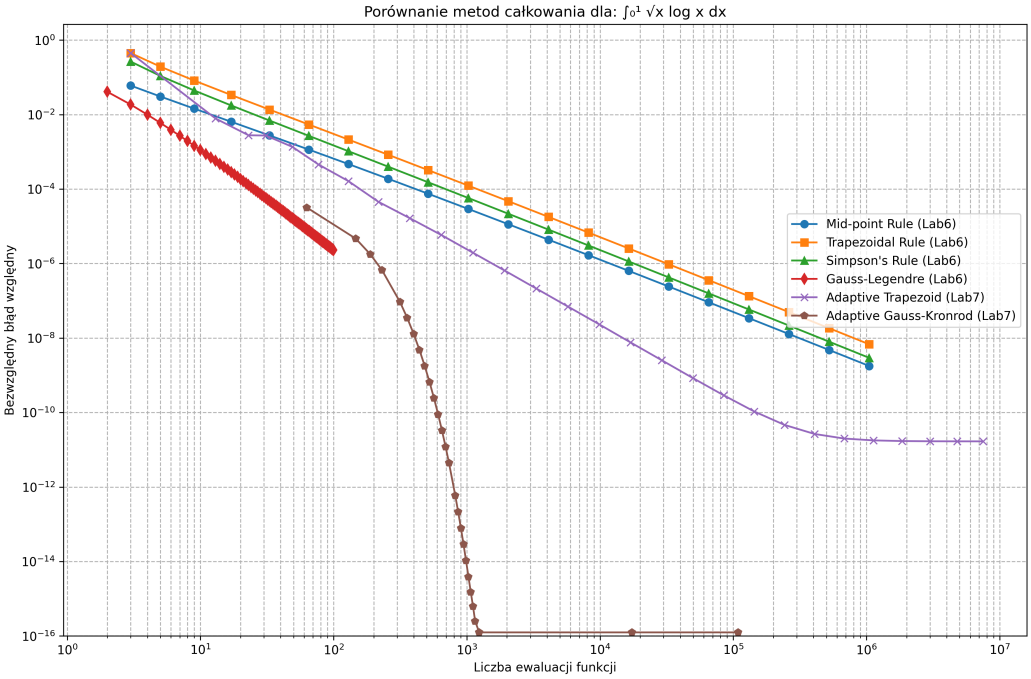
Generowanie wykresów: Dla każdej z badanych całek (Zadanie 1 oraz Zadanie 2a i 2b), wygenerowano wykresy zależności bezwzględnego błędu względnego od liczby ewaluacji funkcji w skali log-log. Na każdym wykresie umieszczono wyniki dla wszystkich metod z Lab6 oraz obu metod adaptacyjnych z Lab7.
Dla trzeciej całki (Zadanie 2b), ze względu na jej potencjalnie trudniejszy charakter (ostre piki), w kodzie zastosowano pewne ograniczenia na zakresy parametrów dla metod z Lab6 (mniejsze `m_values`, mniejszy zakres `n_gauss_range`) oraz niższe limity ewaluacji i mniejszą liczbę testowanych tolerancji dla metod adaptacyjnych, aby skrócić czas obliczeń.

4 Wyniki

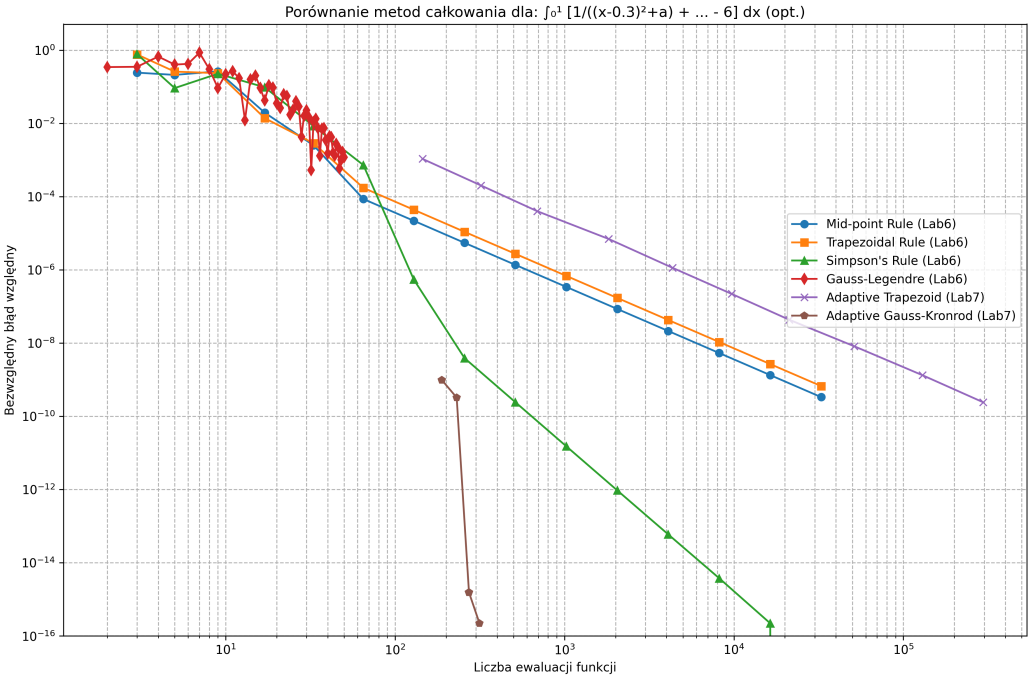
Poniżej przedstawiono wykresy porównujące metody całkowania numerycznego dla analizowanych funkcji. Na osi X znajduje się liczba ewaluacji funkcji podcałkowej, a na osi Y bezwzględny błąd względny. Obie osie są w skali logarytmicznej.



Rysunek 1: Porównanie metod całkowania dla całki $\int_0^1 \frac{4}{1+x^2} dx$.



Rysunek 2: Porównanie metod całkowania dla całki $\int_0^1 \sqrt{x} \log x dx$.



Rysunek 3: Porównanie metod całkowania dla całki $\int_0^1 \left(\frac{1}{(x-0.3)^2+a} + \frac{1}{(x-0.9)^2+b} - 6 \right) dx$ (opt.).

5 Wnioski

Analiza przeprowadzonych eksperymentów i wygenerowanych wykresów pozwala sformułować następujące wnioski:

- **Ogólna charakterystyka metod adaptacyjnych**: O ile metoda adaptacyjna Gaussa-Kronroda w każdym wypadku okazała się lepsza, to dla dwóch pierwszych funkcji adaptacyjna metoda trapezów wziąż okazała się lepsza od metod nieadaptacyjnych. Wyjątkiem tu jest funkcja numer 3. Błąd zawsze maleje wraz z liczbą wywołań, choć jest to zachowanie nieregularne w przypadku adaptacyjnej metody trapezów dla funkcji nr 1.
- **Porównanie metod adaptacyjnych**:
  - **Adaptacyjna kwadratura Gaussa-Kronroda** (`scipy.integrate.quad_vec`) okazuje się być bardziej efektywna, tj. osiąga mniejszy błąd przy tej samej liczbie ewaluacji funkcji lub wymaga mniejszej liczby ewaluacji do osiągnięcia określonego błędu, w porównaniu do adaptacyjnej kwadratury trapezów. Jest to zgodne z teorią, gdyż metody Gaussa-Kronroda są metodami wyższego rzędu.
  - **Adaptacyjna kwadratura trapezów**, będąc metodą prostszą, może wymagać znacznie większej liczby ewaluacji funkcji, aby osiągnąć bardzo małe błędy, szczególnie dla funkcji gładkich, gdzie metody wyższego rzędu szybko zbiegają.
- **Porównanie z metodami nieadaptacyjnymi (Lab6)**:
  - Dla funkcji prostych (np. $\int 4/(1+x^2)dx$), metody Gaussa-Legendre'a oraz Simpsona pokazują bardzo dobrą zbieżność i są lepsze od adaptacyjnej kwadratury trapezów. Adaptacyjna kwadratura Gaussa-Kronroda utrzymuje swoją wysoką efektywność (w naszym wypadku błąd zerowy).
  - Dla funkcji z osobliwościami, ostrymi pikami lub gwałtownymi zmianami (np. $\int \sqrt{x}\log x dx$ przy $x = 0$, lub całka z Zadania 2b z ostrymi pikami wokół $x = 0.3$ i $x = 0.9$), metoda adaptacyjna Gaussa-Kronroda wykazała swoją przewagę. Koncentruje ona wysiłek obliczeniowy w "trudnych"regionach funkcji, podczas gdy metody nieadaptacyjne z równomiernym podziałem przedziału mogą marnować obliczenia w regionach, gdzie funkcja jest gładka, lub nie być w stanie dokładnie przybliżyć całki w regionach problematycznych bez bardzo dużej, globalnej liczby podziałów. Co ciekawe, adaptacyjna metoda trapezów radzi sobie najgorzej.
- **Wpływ limitów ewaluacji**: Wprowadzone w kodzie limity maksymalnej liczby ewaluacji dla metod adaptacyjnych (`max_evals_adaptive_trap`, `max_evals_adaptive_gk`)

mogą powodować, że krzywe błędów dla tych metod "urywają się"na wykresach, jeśli próba osiągnięcia bardzo niskiej tolerancji wymagałaby przekroczenia tego limitu. Jest to kompromis między dokładnością a czasem obliczeń.

- **Specyfika funkcji**:
  - Dla całki $\int 4/(1+x^2)dx$ (Rysunek 1), która jest gładka, wszystkie metody zbiegają, ale metody wyższego rzędu (Simpson, Gauss-Legendre, Gauss-Kronrod) robią to szybciej.
  - Dla całki $\int \sqrt{x}\log x dx$ (Rysunek 2), która ma osobliwość w pochodnej w $x = 0$, metody adaptacyjne lepiej radzą sobie z zachowaniem funkcji blisko granicy przedziału.
  - Dla całki z Zadania 2b (Rysunek 3), która posiada dwa ostre piki, metody adaptacyjne powinny wykazać znaczną przewagę nad metodami o stałym kroku, efektywnie zagęszczając obliczenia wokół tych pików. Dla metody trapezów tak się nie stało.

Podsumowując, wybór odpowiedniej metody całkowania numerycznego zależy od charakterystyki funkcji podcałkowej oraz wymaganej dokładności. Metody adaptacyjne, a w szczególności adaptacyjna kwadratura Gaussa-Kronroda, oferują wygodne i efektywne podejście dla szerokiej gamy problemów, zwłaszcza gdy zachowanie funkcji jest nierównomierne.