

AKADEMIA GÓRNICZO-HUTNICZA

WYDZIAŁ INFORMATYKI
KIERUNEK INFORMATYKA



METODY OBLICZENIOWE W NAUCE I TECHNICE

Laboratorium 7

Kwadratury adaptacyjne

Wojciech Michaluk, Kyrylo Iakymenko

Kraków, 19 kwietnia 2024

1 Wprowadzenie

Celem zadań w ramach tego laboratorium jest wykorzystanie metod całkowania numerycznego (tak jak w poprzednim laboratorium, **laboratorium 6**) oraz nowo poznanych metod całkowania adaptacyjnego do obliczenia wartości wybranych całek, w tym całki $\int_0^1 \frac{4}{1+x^2} dx$ z poprzedniego laboratorium oraz porównanie różnych kwadratur.

W pierwszym zadaniu obliczymy przytoczoną całkę, korzystając z kwadratur adaptacyjnych: trapezów i Gaussa-Kronroda. Obliczymy wartości bezwzględne błędu względnego w zależności od liczby ewaluacji funkcji podcałkowej $f(x) = \frac{4}{1+x^2}$. Następnie przedstawimy wyniki na oddzielnych wykresach dla każdej metody i umieścimy je na wspólnym wykresie z metodami z poprzedniego laboratorium.

W drugim zadaniu powtarzamy obliczenia z poprzedniego i obecnego laboratorium dla dwóch kolejnych całek:

$$\int_0^1 \sqrt{x} \log x dx \quad (1)$$

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

Dla nich przy obliczeniach błędu względnego skorzystamy z odpowiednich tożsamości, żeby móc porównać rezultat obliczeń z rzeczywistą wartością.

2 Zadanie 1

2.1 Opis zadania

Istotą całkowania adaptacyjnego jest fakt, że we wzorach złożonych węzły są równoodległe, ale funkcje podcałkowe mogą mieć przedziałami różny przebieg. Przebieg funkcji może być np. **wolnozmienny** lub **oscylacyjny**. Stąd rozwiązaniem tego problemu jest dobranie odpowiedniego kroku w danym podprzedziale. Główną ideą metody jest podział przedziału na fragmenty, dla których zastosowanie jednej z metod klasycznych da dostatecznie dokładny wynik.

W ramach tego zadania wykorzystujemy kwadratury adaptacyjne:

1. trapezów,
2. Gaussa-Kronroda.

Dla każdej z nich obliczymy wartość bezwzględną błędu względnego w zależności od liczby ewaluacji funkcji podcałkowej, wiedząc, że rzeczywista wartość wspomnianej całki wynosi π oraz przedstawimy je na wykresie. Następnie przedstawimy te wyniki na wspólnym wykresie wraz z wykresami metod całkowania numerycznego z laboratorium 6, aby porównać te metody.

2.2 Opracowanie zadania

Procedurę całkowania adaptacyjnego można samodzielnie np. zaimplementować rekurencyjnie, przyjmując jako warunek stopu odpowiednią tolerancję błędu, natomiast w ramach tego zadania nie będziemy implementować przydatnych funkcji ręcznie - skorzystamy z zaproponowanej funkcji `scipy.integrate.quad_vec`. Szczególnie interesującymi dla nas argumentami, które ona przyjmuje są:

- funkcja f , z której liczymy całkę na zadanym przedziale,
- a i b - dwa argumenty, wskazujące na krańce przedziału (w tym zadaniu odpowiednio $a = 0$ i $b = 1$),
- poziom tolerancji błędu $epsabs$,

- rodzaj kwadratury *quadrature* - rozważamy kwadratury *trapezoid* oraz *gk-21* - domyślna wersja kwadratury Gaussa-Kronroda dla przedziałów skończonych,
- oraz flaga `full_output` wskazująca, czy chcemy obsługiwać dodatkową wartość zwracaną `info` - jest to struktura z informacjami o parametrach procedury, w tym szczególnie interesuje nas jej pole `neval`, które zawiera informację o liczbie ewaluacji funkcji podcałkowej. Dlatego ustawiamy tę flagę na `True`.

Funkcja ta zwraca również wartość `error`, ale jej nie używamy - jest to pewne oszacowanie błędu, którego nie potrzebujemy - bo znamy rzeczywistą wartość.

Przedstawmy nasze funkcje wykorzystujące funkcję biblioteczną:

```
# Funkcja obliczająca całkę metodą adaptacyjną trapezów
# zwracamy obliczoną wartość i liczbę ewaluacji funkcji
def trapezoidal_adaptive(f, tolerance):
    value, error, info = quad_vec(f, a, b, epsabs = tolerance,
                                  quadrature = 'trapezoid', full_output = True)
    return value, info.neval

# Funkcja obliczająca całkę metodą adaptacyjną Gaussa-Kronroda
# zwracamy obliczoną wartość i liczbę ewaluacji funkcji
def Gauss_Kronrod_adaptive(f, tolerance):
    value, error, info = quad_vec(f, a, b, epsabs = tolerance,
                                  quadrature = 'gk21', full_output = True)
    return value, info.neval
```

Listing 1: Kwadratury adaptacyjne: trapezów i Gaussa-Kronroda

Na liczbę ewaluacji funkcji podcałkowej w tym przypadku nie mamy bezpośredniego wpływu - zależy ona m. in. od przyjętej tolerancji, stąd uzyskujemy tę wielkość poprzez `info.neval`. Rozważamy wartości tolerancji błędu 10^{-i} dla $i = 0, 1, \dots, 14$. Dla każdej z rozważanych wartości obliczamy poprzez przedstawione powyżej funkcje wynik oraz liczbę ewaluacji funkcji podcałkowej dla obu kwadratur adaptacyjnych. Wyniki dotyczące liczby ewaluacji funkcji podcałkowej dla każdej z metod przedstawiamy w poniższych tabelach.

Tolerancja 10^{-i} , i	0	1	2	3	4	5	6	7	8	9	10...14
Liczba ewaluacji n	8	13	33	103	348	1073	3363	11008	19648	19648	19648

Tabela 1: Wyniki dla adaptacyjnej kwadratury trapezów w zależności od tolerancji 10^{-i}

Tolerancja 10^{-i} , i	0	1	2...14
Liczba ewaluacji n	63	63	63

Tabela 2: Wyniki dla adaptacyjnej kwadratury Gaussa-Kronroda w zależności od tolerancji 10^{-i}

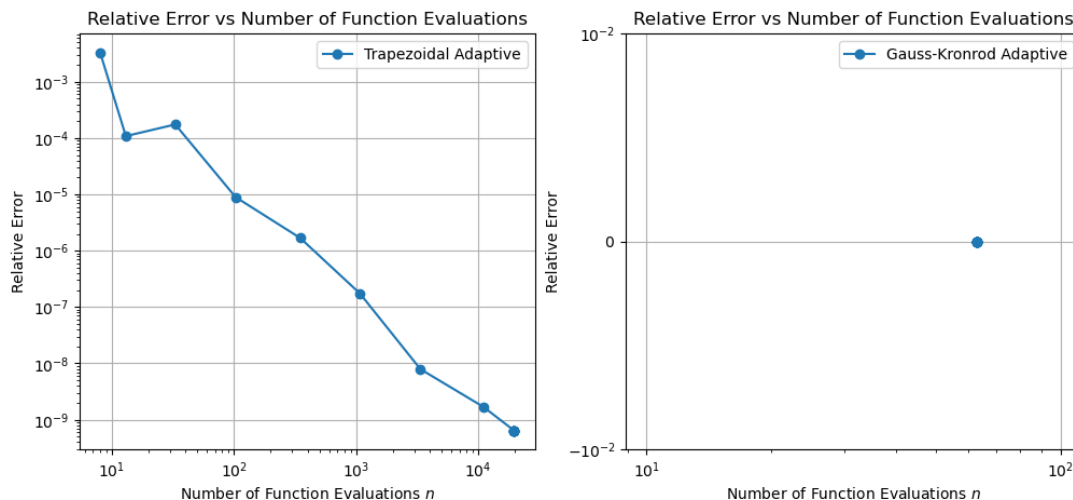
Jak można zauważyć, dla kwadratury trapezów dla $i = 8, 9, \dots, 14$ liczba ewaluacji funkcji podcałkowej jest taka sama. Z kolei dla kwadratury Gaussa-Kronroda dla wszystkich wartości $i = 0, 1, \dots, 14$ liczba ewaluacji jest taka sama i wynosi 63.

Będzie miało to wpływ na reprezentację danych na wykresie, ale zanim do tego przejdziemy, obliczamy wartość bezwzględnej błędów względnego, korzystając z prostej funkcji, podobnie jak w poprzednim laboratorium:

```
def relative_error(approx_val, actual_val = np.pi):
    return np.abs((approx_val - actual_val) / actual_val)
```

Listing 2: Obliczanie błędów względnych przybliżeń

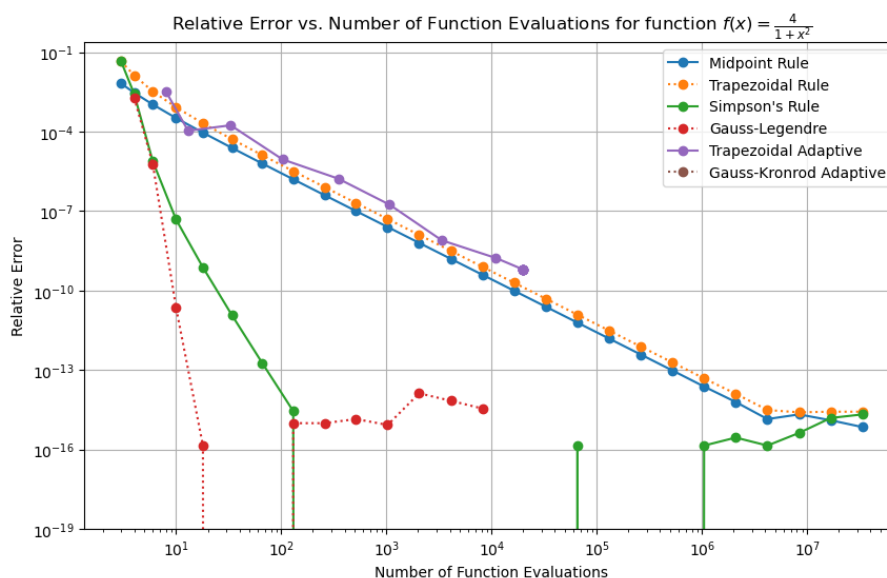
Możemy zatem przejść do przedstawienia wyników na okresie. Najpierw prezentujemy wyniki dla kwadratur adaptacyjnych na oddzielnych wykresach.



Rysunek 1: Błędy względne metod całkowania adaptacyjnego w zależności od liczby ewaluacji funkcji

Zauważamy bardzo ciekawe zjawisko w przypadku wykresu po prawej stronie. Dla rozważanej funkcji podcałkowej f w każdym z przypadków błąd metody wynosi 0, tzn. obliczona wartość jest (z dokładnością do precyzji reprezentacji `numpy.float64`) równa π . Na wykresie po lewej stronie przyjęliśmy na obu osiach skalę logarytmiczną, natomiast po prawej stronie przez wartość 0 używamy na osi Y opcji `symlog` - czyli skala logarytmiczna, ale w pobliżu 0 jest liniowa, żeby móc poprawnie przedstawić tę wartość na wykresie.

Teraz przejdziemy do przedstawienia wyników na wspólnym wykresie, wraz z metodami całkowania numerycznego z poprzedniego laboratorium (tzn. kwadratur złożonych - prostokątów, trapezów i Simpsona oraz kwadratury Gaussa-Legendre'a). Przyjmujemy na obu osiach skalę logarytmiczną, ponieważ przy wybraniu opcji `symlog` wykres stawał się bardzo nieczytelny. Dlatego podjęliśmy taką decyzję, kosztem czego nie widać punktu reprezentującego wynik dla kwadratury adaptacyjnej Gaussa-Kronroda, natomiast dobrze jest widoczny wynik dla kwadratury adaptacyjnej trapezów (oraz poprzednio badanych).



Rysunek 2: Wspólny wykres błędów względnych metod całkowania w zależności od liczby ewaluacji funkcji

Przeanalizujemy wyniki dla kwadratury adaptacyjnej trapezów (podpisane na wykresie - fioletowy kolor). Zauważamy, że wykres ma na wynikowym przedziale liczby ewaluacji funkcji n analogiczny kształt i przebieg, jak dla kwadratur złożonych: prostokątów i trapezów. Jako że przebiega nieco wyżej, oznacza to, że wartość błędu jest nawet trochę większa. Z kolei metoda Gaussa-Kronroda dała (zakładając, że nie popełniliśmy błędu) bardzo dokładny wynik. Warto wspomnieć, że ogólnie w przypadku dość prostych funkcji kwadratury adaptacyjne mogą sobie radzić nawet gorzej niż kwadratury złożone, co można zauważyć właśnie w powyższym przypadku dla kwadratury trapezów. Często prawdziwą użyteczność kwadratur adaptacyjnych widać dla bardziej skomplikowanych funkcji, co spróbujemy przebadać w ramach zadania drugiego.

3 Zadanie 2

3.1 Opis zadania

Celem tego zadania jest przeprowadzenie obliczeń z tego laboratorium oraz poprzedniego, tzn. wykorzystanie kwadratur złożonych (metody *prostokątów*, *trapezów*, *Simpsona* oraz *Gaussa-Legendre'a*), a także kwadratur adaptacyjnych (*trapezów* i *Gaussa-Kronroda*).

Aby obliczyć wartości błędów względnych w zależności od liczby ewaluacji funkcji podcałkowych $f_1(x) = \sqrt{x} \log x$ oraz $f_2(x) = \frac{1}{(x-0.3)^2+a} + \frac{1}{(x-0.9)^2+b} - 6$, korzystamy z faktu, iż wartość całki $\int_0^1 f_1(x) dx$ wynosi $-\frac{4}{9}$. Z kolei w f_2 przyjmujemy $a = 0.001, b = 0.004$, natomiast do wyznaczenia błędu obliczeń całki z tej funkcji na przedziale $[0; 1]$ korzystamy z podanej w poleceniu zależności:

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

Korzystając z własności operatora całkowania (*liniowość*) i podstawiając we wzorze w odpowiednie miejsce przyjęte stałe, wiemy, że rzeczywistą wartością dla całki z f_2 jest

$$f_2(x) = \int_0^1 f_2(x) dx = \frac{1}{\sqrt{0.001}} \left(\arctg \frac{0.7}{\sqrt{0.001}} + \arctg \frac{0.3}{\sqrt{0.001}} \right) + \frac{1}{\sqrt{0.004}} \left(\arctg \frac{0.1}{\sqrt{0.004}} + \arctg \frac{0.9}{\sqrt{0.004}} \right) - 6.$$

3.2 Opracowanie zadania

W ramach zadania 2 korzystamy z kodu przedstawionego w poprzednim zadaniu oraz w poprzednim laboratorium do obliczenia całek z funkcji f_1 i f_2 . W kodzie zapisujemy te funkcje następująco:

```
SMALL_CONST = 10 ** -13 # bo logarytm nie jest określony w zerze

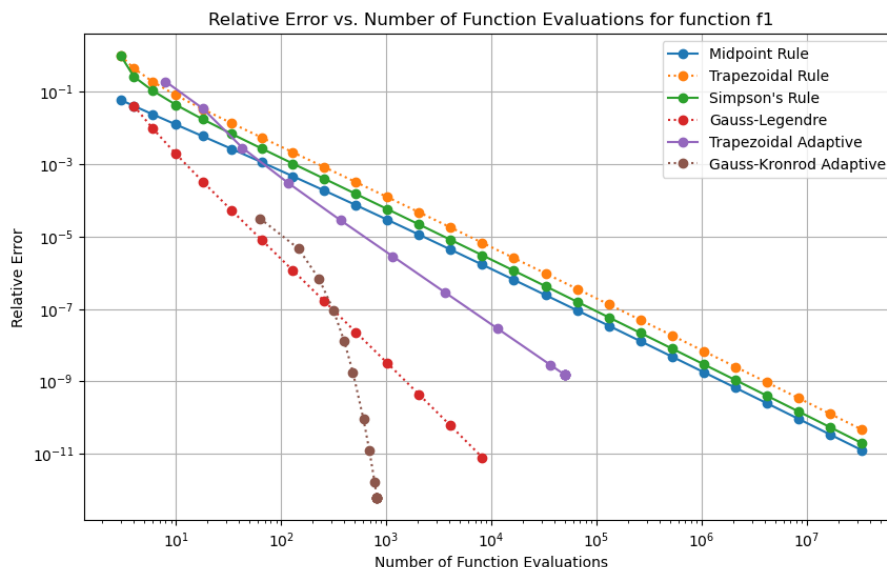
def f1(x):
    return np.sqrt(x + SMALL_CONST) * np.log(x + SMALL_CONST)

def f2(x):
    return 1 / ((x - 0.3) ** 2 + 0.001) + 1 / ((x - 0.9) ** 2 + 0.004) - 6
```

Listing 3: Funkcje podcałkowe do zadania 2

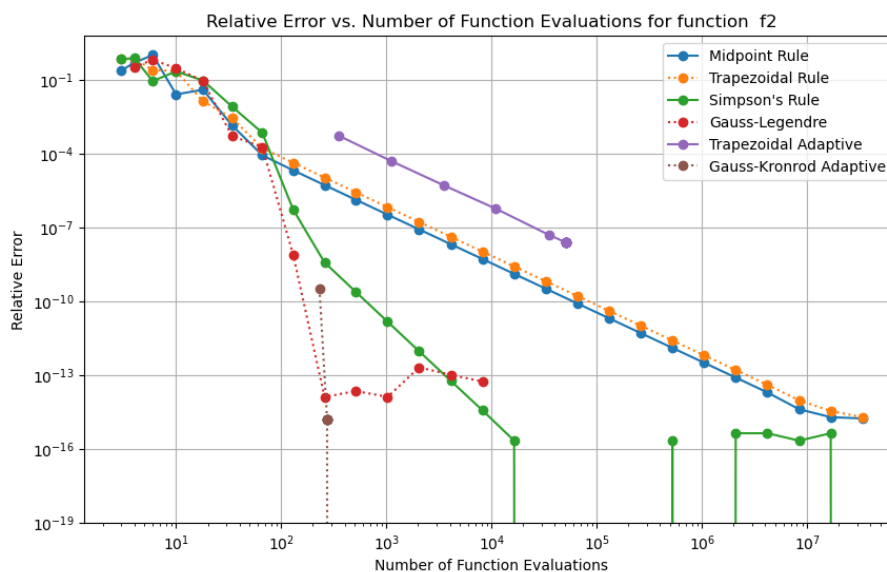
W przypadku funkcji f_1 używamy przesunięcia o wartość $SMALL_CONST$ ze względu na dziedzinę funkcji podcałkowej - matematycznie nie stanowi to problemu przy obliczaniu całki, ale metoda numeryczna w punkcie 0 nie obliczy wartości, stąd ten zabieg. Dobraliśmy wartość na tyle małą, żeby nie wpłynąć znacząco na wynik całkowania i błąd obliczeń.

Przedstawiamy wykresy zależności błędu względnego od liczby ewaluacji funkcji poniżej.



Rysunek 3: Wykres błędów względnych metod całkowania w zależności od liczby ewaluacji funkcji f_1

Dla funkcji f_1 widać, jak metody Simpson'a, trapezów oraz prostokątów zachowują się w bardzo podobny sposób. Zauważamy, że w tym przypadku metoda Simpsona już nie ma takiej przewagi nad innymi metodami, jak w poprzednim przykładzie i w zasadzie różnice tych trzech metod w tym przypadku sprowadzają się do różnego kosztu obliczeniowego. Adaptacyjne metody całkowania okazały się być lepsze od zwykłych. Wykres błędu metody adaptacyjnej trapezów ma podobny kształt do omówionych wcześniej 3 metod, ale jednak znacząco szybciej zbiega do 0. To samo można powiedzieć odnośnie metody Gaussa-Legendre'a, tylko w tym przypadku empiryczny rząd zbieżności wydaje się być większy w porównaniu do metod poprzednich. Znacząco inną charakterystykę ma błąd metody Gaussa-Kronroda. W tym przypadku na wykresie logarytmicznym to jest jedyna z metod, której spadek ma charakter nieliniowy. Gauss-Kronrod również osiąga najmniejszą wartość błędu względnego ze wszystkich badanych metod na poziomie 10^{-12} . Zatem można powiedzieć, że to była najskuteczniejsza metoda dla tego przypadku (oczywiście nie bierzemy pod uwagę kosztu obliczeniowego).



Rysunek 4: Wykres błędów względnych metod całkowania w zależności od liczby ewaluacji funkcji f_2

Dla funkcji f_2 mamy podobną sytuację, jeśli chodzi o metody trapezów i prostokątów. Ich wykresy błędu względnego są czymś w rodzaju prostej na wykresie w skali logarytmicznej. Tym razem metoda adaptacyjna trapezów jest trochę gorsza w porównaniu do metod złożonych i pokazuje siebie najgorzej, porównując do innych omówionych metod. Dla metod adaptacyjnych oraz metody Gaussa-Legendre'a widzimy podobną sytuację jak w przypadku funkcji $f(x) = \frac{4}{1+x^2}$. Znowu metody Simpson'a, Gaussa-Kronroda i Gaussa-Legendre'a pokazują najlepsze wyniki.

4 Podsumowanie i wnioski

W sprawozdaniu z tego laboratorium zaprezentowaliśmy wyniki eksperymentów z zastosowaniem metod całkowania numerycznego. Celem było porównanie różnych metod oraz analiza ich skuteczności dla różnych funkcji podcałkowych.

Metody adaptacyjne pokazały się najlepiej w przypadkach estymacji całki z funkcji, których wykres był bardziej skomplikowany. Na przykład używając metody Gaussa-Kronroda uzyskaliśmy najdokładniejsze i najszybciej zbiegające do 0 wyniki we wszystkich przypadkach. Warto również zauważyć, że metody adaptacyjne miały mniejszą przewagę nad innymi metodami w przypadkach funkcji stosunkowo prostych, takich jak $f_1(x) = \sqrt{x} \log(x)$. Dla funkcji f_1 metody prostokątów i trapezów dla mniejszych wartości h zbliżały się do dokładności adaptacyjnych metod całkowania numerycznego. Zwróciliśmy również uwagę na metodę Simpson'a, która pokazywała podobne zachowanie do metod adaptacyjnych w przypadkach bardziej skomplikowanych funkcji, natomiast dla f_1 jej wykres przypominał inne metody nieadaptacyjne. Spośród badanych metod adaptacyjnych o wiele gorzej pokazała siebie metoda adaptacyjna trapezów. W niektórych przypadkach była nawet mniej dokładna niż zwykła metoda trapezów.

Podsumowując, można powiedzieć, że generalnie metody adaptacyjne wykazują się większą dokładnością w porównaniu do nieadaptacyjnych, ale jeśli nasza funkcja podcałkowa ma stosunkowo prosty wykres, to użycie bardziej skomplikowanych algorytmów całkowania może być nieuzasadnione. Warto zauważyć również, że niekoniecznie mamy możliwość zmiany liczby ewaluacji funkcji podcałkowej dla algorytmów adaptacyjnych, co może się wiązać z nieoczekiwanym dużym kosztem obliczeniowym w porównaniu do metod takich jak metoda prostokątów czy metoda trapezów, w których koszt obliczeniowy jest liniowy ze względu na ilość ewaluacji funkcji podcałkowej i precyzja obliczeń jest bardziej przewidywalna (we wszystkich przypadkach wykres błędu względnego dla tych metod miał charakter liniowy na wykresie logarytmicznym). Stąd użycie metod adaptacyjnych w programie powinno być dobrze uzasadnione i może się wiązać z dłuższym czasem obliczeń.

Literatura

[1] Treść przedstawiona na wykładzie o kwadraturach adaptacyjnych.