

Programowanie 1R – Ćwiczenia 12

Całkowanie numeryczne (kwadratury)

Mikołaj Marcinkowski

2 czerwca 2026

Zadania możesz wykonywać jako oddzielne skrypty `.py` lub w notatniku jupytera `.ipynb`, chyba że treść zadania wskazuje inaczej.

Zadania wzorowane na Serii 13 zadań autorstwa Bartłomieja Zglinickiego (<https://www.fuw.edu.pl/~bzglinicki/teaching/p1r/zadania/seria-13/>), udostępnionej na licencji CC BY 4.0..

1 Kwadratury

Kwadratury to metody numeryczne oszacowania całki

$$I(f, a, b) = \int_a^b f(x) dx. \quad (1)$$

Zazwyczaj sprowadzają się do podzielenia odcinka $[a, b]$ na punkty $x_0 < x_1 < \dots < x_N$, gdzie $x_0 = a$ i $x_N = b$. Wartość całki $I(f, a, b)$ jest szacowana na podstawie wartości funkcji $f(x_k)$, dodawanych z odpowiednimi wagami i mnożonych przez odległości między punktami x_k . W szczególności nie potrzebujemy mieć postaci funkcji $f(x)$, wystarczą nam same wartości w punktach x_k , więc możemy na przykład całkować dane eksperymentalne czy wyznaczone numerycznie.

Jest bardzo wiele sposobów na rozwiązanie tego problemu, jak przybliżanie funkcji f różnego rodzaju wielomianami, różne strategie doboru punktów x_k itp. Kilkadziesiąt z nich jest opisanych w [1] (głównie w rozdziale 4).

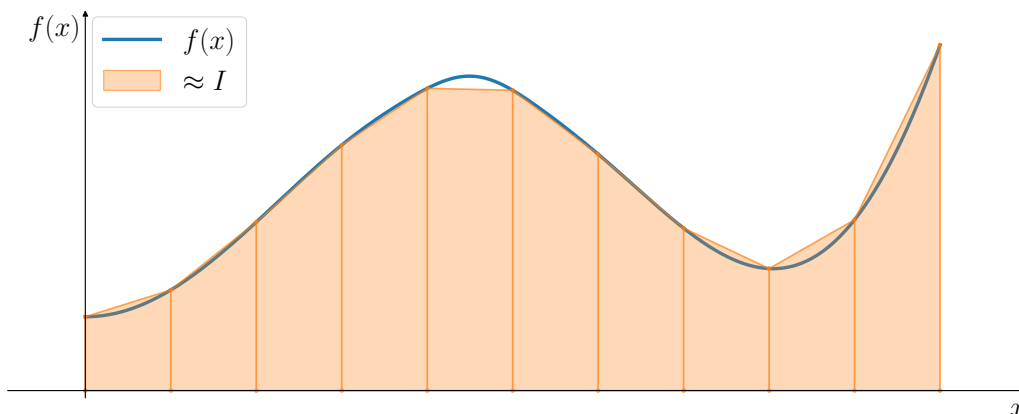
2 Zadania

Zadanie 1. Metoda trapezów

Metoda trapezów polega na równomiernym rozstawieniu punktów x_k :

$$x_k = a + \frac{b-a}{N} \cdot k = a + \Delta x \cdot k, \quad (2)$$

i oszacowaniu całki z funkcji f przez sumę pól trapezów o bokach $f(x_k)$ i podstawach Δx :



Rysunek 1: Oszacowanie całki $I(f, a, b) = \int_a^b f(x)dx$ metodą trapezów.

Robiąc serię przekształceń można pokazać, że metoda trapezów szacuje całkę I jako:

$$I(f, a, b) \approx \frac{\Delta x}{2} \sum_{k=1}^N (f(x_{k-1}) + f(x_k)). \quad (3)$$

Korzystając z tej metody oblicz całkę:

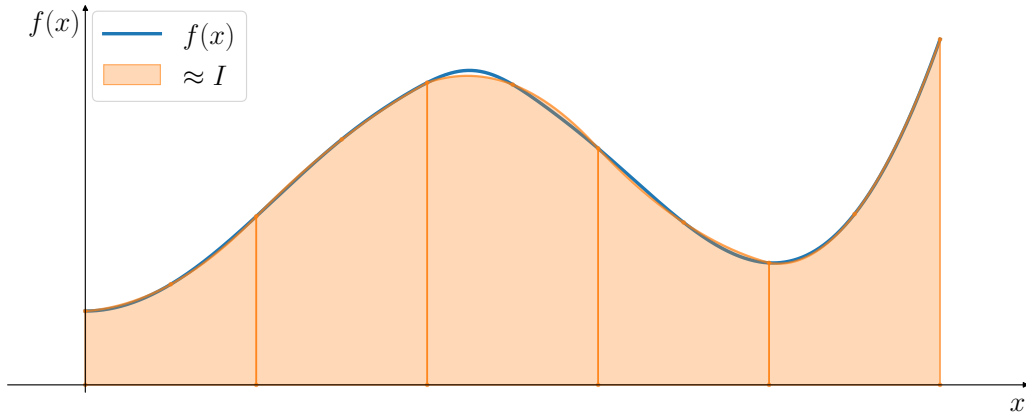
$$\log(10) = \int_1^{10} \frac{1}{x} dx, \quad (4)$$

i porównaj wartość z daną przez funkcję `np.log`, dla różnych wartości N .

Zadanie 2. Metoda Simpsona $\frac{1}{3}$

Metoda Simpsona $\frac{1}{3}$ wykorzystuje ten sam rozstaw punktów, co metoda trapezów (2), przy czym liczba punktów **musi być nieparzysta**. W tej metodzie punkty x_k grupuje się w trójki, po czym przez każdą trójkę punktów przepuszcza się parabolę mającą przybliżyć kształt funkcji $f(x)$. Całkę z $f(x)$

szacuje się wtedy przez sumę całek z tych parabol, które można już obliczyć analitycznie:



Rysunek 2: Oszacowanie całki $I(f, a, b) = \int_a^b f(x)dx$ metodą Simpsona $\frac{1}{3}$.

Robiąc serię przekształceń można pokazać, że metoda Simpsona $\frac{1}{3}$ szacuje całkę I jako:

$$\begin{aligned}
 I(f, a, b) &\approx \Delta x \sum_{k=1}^{N/2} \left(\frac{1}{3}f(x_{2k-2}) + \frac{4}{3}f(x_{2k-1}) + \frac{1}{3}f(x_{2k}) \right) = \\
 &= \Delta x \left(\frac{1}{3}f(x_0) + \frac{4}{3}f(x_1) + \frac{2}{3}f(x_2) + \frac{4}{3}f(x_3) + \frac{2}{3}f(x_4) + \cdots + \frac{4}{3}f(x_{N-1}) + \frac{1}{3}f(x_N) \right) = \\
 &= \frac{\Delta x}{3} \left(f(a) + 4 \sum_{k=1}^{N/2} f(x_{2k-1}) + 2 \sum_{k=1}^{N/2-1} f(x_{2k}) + f(b) \right), \quad (5)
 \end{aligned}$$

gdzie korzystam z faktu, że $x_0 = a$ i $x_N = b$.

Korzystając z tej metody oblicz całkę:

$$\frac{\pi}{4} = \int_0^1 \frac{1}{1+x^2} dx, \quad (6)$$

i porównaj z wartością $\pi/4$ daną przez `numpy`, dla różnych wartości N . Pamiętaj, że aby liczba punktów była nieparzysta, N musi być parzyste (x_k numerujemy od zera).

Na przyszłość: Istnieje więcej metod Simpsona, zwykle nazywanych od ułamka, który da się wyciągnąć przed sumę (na przykład metoda $\frac{3}{8}$), które przybliżają funkcję $f(x)$ wielomianami wyższego rzędu. Metody te mają jednak inne wymogi co do liczby punktów x_k (na przykład w metodzie $\frac{3}{8}$ to musi być (liczba podzielna przez 3) + 1).

Zadanie 3. quad

Biblioteka `scipy` posiada funkcję do całkowania numerycznego (kwadratur) `scipy.integrate.quad`, która implementuje szereg algorytmów do całkowania z Fortranowej biblioteki `QUADPACK`. Funkcja ta sama dobiera sobie rozkład punktów x_k , szacując błąd i starając się utrzymać go poniżej zadanego poziomu. Zwraca krotkę (szacunek_całki, niepewność). Korzystając z tej funkcji, oblicz całki z poprzednich zadań.

Funkcja `quad` pozwala też na całkowanie po nieskończonych przedziałach. Robi to stosując zamianę zmiennych, która rzutuje nieskończony odcinek na skończony (na przykład funkcja $1/x$ rzutuje odcinek $[0, 1]$ na $[1, \infty]$). Korzystając z funkcji `quad` oblicz całkę:

$$\frac{\sqrt{\pi}}{2} = \int_0^{\infty} e^{-x^2} dx, \quad (7)$$

i porównaj z wartością $\sqrt{\pi}/2$ daną przez `numpy`.

Zadanie 4. Całkowanie analityczne biblioteką sympy

Biblioteka `sympy` pozwala na obliczenia symboliczne w pythonie, w podobny sposób jak Wolfram Mathematica. Uruchomiona w `jupyterze` wyświetla napisane równania za pomocą $\text{\LaTeX}$ a. Korzystając z tej biblioteki oblicz analitycznie całki z pozostałych zadań.

Kilka przydatnych obiektów/funkcji z biblioteki `sympy`:

- `x = sympy.symbols('x')` – tworzy zmienną x , z której można budować wyrażenia,
- `sympy.sqrt`, `sympy.exp` – odpowiedniki funkcji $\sqrt{\cdot}$ czy $\exp$,
- `sympy.oo` – nieskończoność ∞ ,
- `sympy.Integral(f, (zmienna, a, b))` – całka z wyrażenia f po zmiennej $zmienna$ od a do b . Wykonanie metody `.doit()` na tym obiekcie zwraca wartość całki.

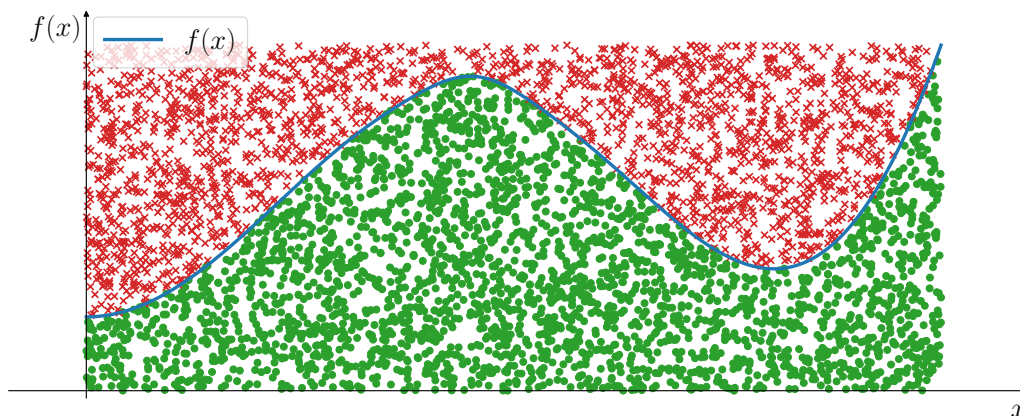
Zadanie 5. Całkowanie Monte Carlo

Całkowanie Monte Carlo polega na losowaniu wielu punktów na wybranym obszarze i sprawdzeniu, jaka ich część wypadła pod całkowaną krzywą. Wylosuj N punktów w kwadracie $[0, 1] \times [0, 1]$. Policz, ile z nich znajduje się w ćwiartce okręgu o promieniu 1 i środku w punkcie $(0, 0)$. Ten eksperyment odpowiada obliczeniu całki

$$\frac{\pi}{4} = \int_0^1 \sqrt{1-x^2} dx. \quad (8)$$

Powtórz całą procedurę M razy. Na podstawie M eksperymentów oblicz spodziewaną wartość π i odchylenie standardowe.

Do losowania punktów możesz skorzystać z `np.random.rand`.



Rysunek 3: Oszacowanie całki $I(f, a, b) = \int_a^b f(x)dx$ metodą Monte Carlo.

Bibliografia

- [1] William H. Press, B. P. Flannery, S. A. Teukolsky i W. T. Vetterling, *Numerical recipes in FORTRAN 77: Volume 1 of FORTRAN numerical recipes volume 1*, 2 wyd. Cambridge, England: Cambridge University Press, wrz. 1992.