

Analiza Numeryczna (M) - Pracownia 2 - Zadanie P2.14

Oskar Bujacz

Instytut Informatyki UWr

24 stycznia 2020

1. Geneza całkowania numerycznego

Problem obliczenia pola był rozważany już w starożytności i od tamtej pory jest obszerną dziedziną analizy matematycznej. Starożytni Grecy problem ten rozwiązywali przez konstruowanie kwadratu o polu rozważanej figury. W ten sposób wiemy o kwadraturze koła, księżycach Hippokratesa czy kwadraturze paraboli. Z tego też powodu obliczanie pola określamy czasem mianem kwadratury. Aż do XVII wieku nie poczyniono większych postępów w dziedzinie obliczania pól figur, kiedy to powstało *Podstawowe twierdzenie rachunku całkowego* odkryte niezależnie przez Newtona i Leibniza. Mimo rozwoju, dalej nie potrafiono analitycznie obliczyć wielu całek, zatem potrzebne było nowe narzędzie.

Termin "całkowanie numeryczne" pojawił się stosunkowo niedawno, bo w 1915r. w publikacji brytyjskiego matematyka Davida Gibba pod tytułem *A Course in Interpolation and Numeric Integration for the Mathematical Laboratory*. Określa on problem przybliżonego obliczenia wartości całki oznaczonej.

$$S \approx \int_a^b f(x) dx$$

Metody całkowania numerycznego pozwalają obliczyć całki funkcji, których nie da się przedstawić analitycznie, jak również obliczyć pole znając tylko wartość w skończonej ilości punktów. Zajmę się różnymi metodami całkowania numerycznego, przedstawię zasadę ich działania, ich zalety i wady oraz pokażę, kiedy warto stosować każdą z nich.

2. Kilka definicji i twierdzeń

Definicja 1. *Metoda całkowania ma rząd dokładności równy n jeśli jest ona dokładna dla wszystkich wielomianów stopnia $\leq n$ dla wszystkich $w \in \Pi_{n-1}$ i istnieje wielomian $w \in \Pi_n \setminus \Pi_{n-1}$ dla którego jest ona niedokładna.*

Definicja 2. *Rozważamy zbiór funkcji całkowalnych F na przedziale $[a, b]$. Funkcjonał liniowy I_p odwzorowuje F na zbiór liczb rzeczywistych R i określamy go jako*

$$I_p(f) = \int_a^b p(x)f(x)dx \quad f \in F$$

Definicja 3. Kwadratura liniową nazywamy funkcjonal Q_n określony jako

$$Q_n[f] \approx \sum_{i=0}^n w_i f(x_i) \quad k = 0 : n$$

gdzie f jest całkowaną funkcją, w_i nazywamy wagami, natomiast x_k węzłami kwadratury Q_n . Resztą kwadratury Q_n to nazywamy różnicę

$$R_n(f) = I_p(f) - Q_n(f)$$

Definicja 4. Kwadraturę Q_n opartą na węzłach o łącznej krotności $n+1$ nazywamy interpolacyjną gdy

$$Q_n[f] \approx \int_a^b w_f(x) dx$$

gdzie w_f jest wielomianem interpolacyjnym funkcji f stopnia co najwyżej n , opartym na tych węzłach. Jej rząd to $n+1$.

Lemat 1. Jeśli kwadraturę Q_n określimy jak w definicji 3, to jej rząd nie przekracza $2n+2$.

Definicja 5. Wielomianami Czebyszewa nazywamy układ wielomianów ortogonalnych tworzących bazę przestrzeni wielomianów spełniających zależność

$$T_0(x) = 1 \quad T_1(x) = x \quad T_k = 2xT_{k-1} - T_{k-2}(x) \quad (1)$$

Posiadają one wiele ciekawych właściwości, ale nie będę ich rozpiszywał, gdyż byłby to temat na kolejną pracownię.

Twierdzenie 1. Jeśli funkcja $\in C[a, b]$ jest ortogonalna w tym przedziale z wagą w względem wszystkich wielomianów klasy Π_n , to w (a, b) funkcja zmienia znak co najmniej $n+1$ razy.

3. Zmiana przedziału całkowania

Dla wygody obliczeń będę przyjmował przedział całkowania $[-1, 1]$, ponieważ w łatwy sposób możemy zmienić przedział całkowania na dowolny inny poprzez liniowe przekształcenie zmiennej. Zastanawiamy wtedy rząd kwadratury. Będę chciał zmienić przedział z $[c, d]$ na $[a, b]$. Pierwotnie niech wzór będzie dokładny dla $f \in \Pi_m$

$$\int_c^d f(t) dt \approx \sum_{i=0}^n A_i f(t_i)$$

podstawiając $x = \lambda(t)$ do całki, gdzie

$$\lambda(t) = \frac{(b-a)t + ad - bc}{d - c}$$

otrzymamy po przekształceniach

$$\int_a^b f(x) dx \approx \frac{b-a}{d-c} \sum_{i=0}^n A_i f\left(\frac{(b-a)t_i + ad - bc}{d-c}\right)$$

Zauważmy, że jeśli $f(t)$ jest wielomianem, to $f(\lambda(t))$ również jest wielomianem, a ich stopnie są równe. Zatem nowy wzór będzie dokładny dla $f \in \Pi_m$

4. Kwadratury Newtona-Cotesa

Na początek zajmę się grupą metod całkowania numerycznego nazywanymi kwadraturami Newtona-Cotesa. Będziemy zakładać, że całkowana funkcja jest zdefiniowana na przedziale $[a, b]$ i znamy jej wartość w równoodległych węzłach x_i gdzie $i = 0, 1 \dots n$ oraz $x_0 = a, x_n = b$. Dodatkowo zakładamy $x_0 < x_1 < x_2 \dots < x_n$. Wtedy szukana całkę możemy aproksymować poprzez interpolację wielomianami Lagrange'a.

$$\int_a^b f(x) \approx \int_a^b L_n(x) dx = \int_a^b \left(\sum_{i=0}^n f(x_i) l_i(x) \right) = \sum_{i=0}^n f(x_i) \int_a^b l_i(x) dx$$

gdzie $L_n(x) \in \Pi_n$ jest wielomianem interpolacyjnym Lagrange'a, natomiast $l_i(x)$ jest postaci

$$\prod_{j=0, j \neq i}^n \frac{x - x_j}{x_i - x_j}$$

Oznaczmy teraz odległość między węzłami jako $h = (b - a)/n$. Wtedy możemy dobierać różne wartości n , czyli ilość podziałów na podprzedziały. Podstawiając $n = 1$ uzyskamy najprostszą kwadraturę Newtona-Cotesa - metodę trapezów.

$$l_0(x) = \frac{b-x}{b-a}, l_1(x) = \frac{x-a}{b-a}$$

wtedy

$$A_0 = \int_a^b l_0(x) dx = \frac{1}{2}(b-a), A_1 = \int_a^b l_1(x) dx = \frac{1}{2}(b-a)$$

W końcu wartość całki możemy zapisać zwartym wzorem.

$$I(f) = \frac{h}{2}(f_0 + f_1)$$

Analizując to, w jaki sposób przybliżamy całkowaną funkcję w metodzie trapezów, łatwo zauważyć, że tak naprawdę przybliżamy funkcję pewną funkcją liniową. Oczywiście, możemy spróbować osiągnąć lepsze przybliżenie poprzez użycie wyższego stopnia, np funkcji kwadratowej, która będzie mieć takie same wartości jak $f(x)$ w krańcach przedziału oraz w punkcie środkowym $m = (a+b)/2$. Wtedy po podstawieniu do wzoru otrzymamy

$$I(f) = \frac{h}{3}(f_0 + 4f_1 + f_2)$$

W taki poprzez sposób poprzez zwiększanie ilości węzłów możemy zwiększać stopień funkcji interpolującej w metodzie Newtona-Cotesa. Można pokazać, że zamknięte wzory Newtona-Cotesa, czyli takie, które interpolują w krańcach przedziału, mają rząd dokładności $d = n$ dla n nieparzystych oraz $d = n+1$ dla n parzystych. Większa dokładność dla parzystych wynika z symetrii funkcji parzystych. Poniżej podaję tabelę współczynników kolejnych stopni kwadratur Newtona-Cotesa na przedziale $[0, 1]$, gdzie $n = 1$ odpowiada metodzie trapezów, natomiast $n = 2$ metodzie Simpsona. Wyrazy $c_{n,d}$ odpowiadają współczynnikom przy błędzie postaci

$$\frac{h^{n+2} f^{(n+1)}(\xi)}{(n+1)!} \int_0^n t(t-1) \dots (t-n) dt$$

gdzie h to odległość między węzłami, a ξ to punkt pośredni.

n	d	A	c_0	c_1	c_2	c_3	c_4	c_5	c_6	$c_{n,d}$
1	1	1/2	1	1						-1/12
2	3	1/3	1	4	1					-1/90
3	3	3/78	1	3	3	1				-3/80
4	5	2/45	7	32	12	32	7			-8/945
5	5	5/288	19	75	50	50	75	19		-275/12096
6	7	1/140	41	236	27	272	27	236	41	-9/1400

Tabela 1: Tablica współczynników wagowych $w_i = Ac_i$ w n punktowej kwadraturze Newtona-Cotesa

W praktyce nie używa się wyższych stopni wielomianów, z powodu znanego problemu z samej interpolacji, czyli *efektu Rungego*. W literaturze pojawiają się również otwarte kwadratury Newtona-Cotesa, które nie korzystają we wzorach z krańców przedziałów, ale nimi już nie będą się zajmować, ponieważ jest to prawie ta sama metoda co kwadratury zamknięte.

4.1. Kwadratury złożone

Dużo lepszym sposobem, który zapobiega efektowi Rungego jest zastosowanie kwadratury złożonej dla małego n . Polega ona na podziale przedziału całkowania na podprzedziały i zsumowaniu całek. Wtedy dla stopnia $n_k = 1$ otrzymujemy złożony wzór trapezów

$$I(f) = \sum_{k=1}^n \int_{x_{k-1}}^{x_k} f(x) dx \approx \sum_{k=1}^n \frac{h}{2} (f(x_{k-1}) + f(x_k))$$

gdzie h to długość podprzedziału. Dla węzłów równoodległych otrzymamy następujący wzór:

$$T_n(f) = h \sum_{k=0}^n f(x_k)$$

Podobnie możemy pokazać wzór dla złożonego Simpsona dla $n_k = 2$. Oznaczmy $n = 2m$

$$I(f) = \sum_{k=1}^m \int_{x_{2k-2}}^{x_{2k}} f(x) dx \approx h \sum_{k=1}^m \left(\frac{1}{3} f(x_{2k-2}) + \frac{4}{3} f(x_{2k-1}) + \frac{1}{3} f(x_{2k}) \right)$$

W ten sam sposób możemy stosować kolejne kwadratury Newtona-Cotesa, jednak w praktyce lepszym pomysłem jest zastosowanie metod kwadratur adaptacyjnym, później przybliże.

5. Kwadratury Gaussa

W metodach Newtona-Cotesa braliśmy pod uwagę układ węzłów x_i równoodległych. Wzory te działają w praktyce całkiem dobrze dla niskich n , jednak mogą być rozbieżne nawet tak szybko jak 2^n przy $n \rightarrow \infty$. Warto zastanowić się, czy szczególny dobór węzłów pozwoli nam w lepszy sposób przybliżać szukaną całkę. Takie kwadratury nazywamy kwadraturami Gaussa, nazwane na imieniem Carla Friefricha Gaussa. Mają one za zadanie być dokładne dla $f \in \Pi_{2n+1}$ dla odpowiednich węzłów x_i i poszczególnych wag w_i dla $i = 1, 2, \dots, n$. Ważne jest to, że dla wybranej funkcji $p(x)$ wagowej szukamy współczynników A_i i węzłów x_i . Do wyznaczania współczynników kwadratur Gaussa używamy wielomianów ortogonalnych.

$$I_p(f) = \int_a^b p(x)f(x) \approx Q_n(f) = \sum_{k=0}^n A_k^{(n)} f(x_k^{(n)})$$

gdzie

$$A_k^{(n)} = \int_a^b p(x)\lambda_k(x)dx, \quad \lambda_k(x) = \frac{\omega(x)}{\omega'(x_k)(x - x_k)}$$

natomiast $\omega(x)$ to $n + 1$ wielomian ortogonalny, natomiast x_0, \dots, x_n to zera tego wielomianu.

W zależności od tego, jaką mamy funkcję wagową, to otrzymujemy różne wielomiany ortogonalne i różne kwadratury.

przedział	ω_x	wielomiany ortogonalne	nazwa kwadratury
$[-1, 1]$	1	Legendre'a	Gaussa-Legendre'a
$(-1, 1)$	$\frac{1}{\sqrt{1-x^2}}$	Czebyszewa	Gaussa-Czebyszewa
$(-1, 1)$	$\frac{1}{\sqrt{1-x^2}}$	Czebyszewa	Gaussa-Czebyszewa
$[0, \infty)$	e^{-x}	Laguerra	Gaussa-Laguerra
$(-\infty, \infty)$	e^{-x^2}	Hermite'a	Gaussa-Hermite'a

Tabela 2: Porównanie różnych kwadratur Gaussa

Odpowiedni dobór funkcji wagowej pozwala na nie tylko obliczanie całek na przedziale ograniczonym, ale także takich na przedziale jedno bądź obustronnie nieograniczonym. Ciekawym faktem jest to, że Kwadratury Gaussa są zbieżne dla każdej funkcji ciągłej, czyli

$$\lim_{n \rightarrow \infty} \sum_{i=0}^n A_{ni} x_{ni} = \int_a^b f(x) \omega(x)$$

6. Kwadratura Clenshawa-Curtisa

Kwadratury Clenshawa-Curtisa, czy też kwadratury Fejera, podobnie jak kwadratury Gaussa, w szczególny sposób wybierają węzły x_i . Korzystają z zer wielomianów Czebyszewa pierwszego lub drugiego rodzaju. Stosują one podstawienie $x = \cos(\theta)$ i używają dyskretnej transformacji cosinowej (DCT) do aproksymacji szeregu cosinusa. Dla przypomnienia, rozważamy:

$$\int_{-1}^1 f(x)dx = \sum_{k=0}^n \omega_k f(x_k)$$

Pierwsza zasada Fejera używa zer wielomianów Czebyszewa pierwszego rodzaju postaci $T_n(x) = \cos(n \arccos(x))$

$$x_k = \cos(\theta_k), \quad \theta_k = \frac{(2k-1)\pi}{2n}, \quad k = 1 : n$$

Może dla nich podać jawną postać na funkcję wagową

$$\omega_k^{f1} = \frac{2}{n} \left(1 - 2 \sum_{j=1}^{\lfloor n/2 \rfloor} \frac{\cos(2j\theta_k)}{4j^2 - 1} \right) \quad k = 1 : n$$

Druga zasada Fejera używa natomiast zer wielomianów Czebyszewa drugiego rodzaju, czyli ekstremów pierwszego rodzaju. Wtedy

$$x_k = \cos(\theta_k), \quad \theta_k = \frac{2k\pi}{2n}, \quad k = 1 : n$$

i otrzymujemy wzory na wagi

$$\omega_k^{f_2} = \frac{4 \sin \theta_k}{n} \sum_{j=1}^{\lfloor n/2 \rfloor} \frac{\sin((2j-1)\theta_k)}{2j-1} \quad k = 1 : n$$

Warto zwrócić, że obydwie zasady Fejera należą do kwadratur otwartych, tzn nie korzystają one z krańców przedziałów. Kwadratura Clenshawa-Curtisa to modyfikacja Drugiej zasady Fejera w celu osiągnięcia kwadratury zamkniętej. Mamy wtedy jawną postać na wagi

$$\omega_k^{cc} = \frac{4}{n} \sum_{j=0}^{n/2} \frac{\cos(2kj\pi/n)}{1-4j^2} \quad k = 1 : n$$

Ważną zaletą tych metody jest możliwość obliczenia potrzebnych współczynników wagowych w czasie $O(n \log n)$, poprzez zastosowanie szybkiej transformaty Fouriera. Co ciekawe, sama FFT powstała później, niż te kwadratury.

6.1. Szybka Transformata Fouriera (FFT)

Szybka Transformata Fouriera to algorytm wyznaczania dyskretnej transformaty Fouriera (DFT), danej wzorem

$$f(x) = \sum_{j=0}^{N-1} c_j e^{ijx}$$

gdzie wartości $f(x_k)$ są znane w $x_k = \pi k/N, k = 0 : N-1$ natomiast współczynniki są dane wzorem

$$c_j = \frac{1}{N} \sum_{k=0}^{N-1} f(x_k) e^{-ijx_k}, \quad j = 0 : N-1$$

Postawiając $\omega_N = e^{-2\pi i/N}$ otrzymujemy

$$c_j = \frac{1}{N} \sum_{k=0}^{N-1} f(x_k) \omega_N^{jk}, \quad j = 0 : N-1$$

gdzie ω_N to pierwiastki n -tego stopnia z 1. Idea szybkiej transformaty Fouriera polega na zastosowaniu strategii dziel i zwyciężaj. Ze względu na złożoność tematu przedstawię tylko główne założenia algorytmu, po więcej informacji odsyłam do G. Dahlquist, A. Björck *Numerical Methods in Scientific Computing*, rozdział 4.7 - The Fast Fourier Transform. Zakładamy, że $N = 2^p$. W naszym przypadku będziemy potrzebować dyskretnej transformacji cosinusowej (DCT) do obliczenia

$$y_k = \sum_{j=0}^{N-1} \cos\left(kj \frac{\pi}{N}\right) f_j \quad j = 1 : N-1$$

gdzie f_j to wartość dla punktu x_j

Rozważmy teraz

$$y_k = \sum_{j=0}^{N-1} \theta_N^{jk} f_j \quad (\theta_N = e^{2\pi i/N}, \quad i^2 = -1)$$

Zakładamy, że $N = 2M$. Po przekształceniach otrzymujemy wzory dla parzystych i nieparzystych wskaźników

$$y_{2k} = \sum_{j=0}^{M-1} (x_j + x_{M+j}) \theta_M^{jk}, \quad k = 0 : M-1$$

$$y_{2k+1} = \sum_{j=0}^{M-1} (x_j x_{M+j}) \theta_N^j \theta_M^{jk}, \quad k = 0 : M-1$$

Zatem widzimy, że możemy rozbić problem rekurencyjnie na dwa o dwukrotnej krótszej długości i osiągniemy złożoność $O(n \log n)$. Warto zaznaczyć, że jest także możliwe obliczanie FFT dla $N \neq 2^p$, jednak wymaga ona generowania macierzy Fouriera. Dla późniejszych testów sprawdzających efektywność FFT zaimplementowałem metodę uproszczoną dla $N = 2^p$.

6.2. Porównanie z kwadraturami gaussowskimi

Kwadratury gaussowskie są rzędu $2n+1$, natomiast rozważana kwadratura Clenshawa-Curtisa jest rzędu tylko n . Wydawałoby się, że będzie w takim razie szybsza, ale nie tak dokładna. Praktyka pokazuje, że obydwie metody mają podobną dokładność dla określonej liczby węzłów. Co ciekawe, nie jest to powszechnie znane zjawisko, dlatego warto o nim wiedzieć i spróbuję to pokazać w późniejszych testach. Znaczną zaletą Clenshawa-Curtisa jest możliwość liczenia funkcji wagowej w czasie $O(n \log n)$ przy pomocy FFT, gdzie dla kwadratur Gaussa potrzebujemy złożoności $O(n^2)$ (problem obliczania wartości własnych macierzy). Dodatkowo, ważną zaletą jest możliwość wykorzystania poprzednia liczonych węzłów przy kwadraturach adaptacyjnych tzn Clenshaw-Curtis używa podzbioru węzłów przy n węzłach ze zbioru $2n$. Standardowo kwadratury Gaussa tego nie umożliwiają, potrzebna jest modyfikacja, np metoda Gaussa-Kronroda.

7. Metoda Romberga

Metoda Romberga jest kolejną metodą całkowania numerycznego. Polega ona na zastosowaniu ekstrapolacji Richardsona na metodzie trapezów. Powstaje wtedy trójkątna macierz kolejnych przybliżeń. Podobnie jak metody Newtona-Cotesa, jest ona stosowna głównie do węzłów równoodległych, jednak tym razem dzielimy przedział na dokładnie 2^n podprzedziałów

$$R(0,0) = h_1(f(a) + f(b)) \quad h_i = \frac{b-a}{2^i}$$

$$R(n,0) = \frac{1}{2}R(n-1,0) + h_n \sum_{k=1}^{2^{n-1}} f(a + a(2k-1)h_n)$$

$$R(n,m) = \frac{4^m R(n,m-1) - R(n-1,m-1)}{4^m - 1} \quad (2)$$

Zerowa ekstrapolacja, $R(n,0)$, jest równoważna metodzie trapezów z $2^n + 1$ węzłami, pierwsza ekstrapolacja to $R(n,1)$, to metoda Simpsona, natomiast $R(n,3)$ to wzór Boole'a. Kolejne kolumny powstałej macierzy różnią się od metod Newtona-Cotesa, są one bardziej rozszerzeniem wzoru Boole'a.

Pokażę teraz dowód, że

$$\lim_{n \rightarrow \infty} R(n, m) = \int_a^b f(x) = I$$

Pierwsza kolumna tablicy zawiera przybliżenia, które można wyrazić jako

$$h \sum_{i=0}^k f(a + ih) = \frac{1}{2}h \sum_{i=0}^{k-1} f(a + ih) + \frac{1}{2}h \sum_{i=1}^k f(a + ih), \quad h = \frac{b-a}{k}$$

po prawej stronie mamy średnia arytmetyczną dwóch sum riemmanowskich dla całki I. Wiemy, z właściwości całki Riemanna, że te sumy dążą do I, kiedy $k \rightarrow \infty$. W drugiej kolumnie mamy wtedy $R(n, 1) = \frac{1}{4}R(n, 0) - \frac{1}{3}R(n-1, 0)$. Ponownie ta suma przy $n \rightarrow \infty$ jest równa całce I. Podobnie rozumiemy dla pozostałych kolumn.

Warto wspomnieć, o tym dlaczego wzór 2 jest używany do konstruowania kolejnych wierszy w metodzie Romberga. Za pomocą wzoru Eulera-Maclaurina możemy zapisać formułę na błąd na 2^{n-1} przedziałach

$$\int_a^b f(x)dx = R(n-1, 0) + a_2h^2 + a_4h^4 + a_6h^6 + \dots \quad (3)$$

$R(n-1, 0)$ reprezentuje wyraz pierwszej kolumny tablicy Romberga. Mamy błąd rzędu $O(h^2)$. Zastosujmy teraz ekstrapolację Richardsona, Wtedy zastępujemy n przez $n+1$ oraz h przez $h/2$. Otrzymujemy

$$\int_a^b f(x)dx = R(n, 0) + \frac{1}{4}a_2h^2 + \frac{1}{16}a_4h^4 + \frac{1}{64}a_6h^6 + \dots \quad (4)$$

Teraz odejmijmy równanie 3 od 4 razy równania 4 celem zniesienia błędu $O(h^2)$.

$$\int_a^b f(x)dx = R(n, 1) - \frac{1}{4}a_4h^4 - \frac{5}{16}a_6h^6 - \dots$$

gdzie

$$R(n, 1) = R(n, 0) + \frac{1}{3}[R(n, 0) - R(n-1, 0)], \quad n > 0$$

Zatem błąd w drugiej kolumnie tablicy jest rzędu $O(h^4)$. Powtarzając ekstrapolację oraz rozumowanie dla 3 kolumny otrzymamy

$$\int_a^b f(x)dx = R(n, 2) + \frac{1}{4^3}a_6h^6 + \frac{21}{4^5}a_8h^8 + \dots$$

gdzie

$$R(n, 2) = R(n, 1) + \frac{1}{15}[R(n, 1) - R(n-1, 1)], \quad n > 1$$

Zredukowaliśmy błąd do rzędu $O(h^6)$. W ten sposób możemy wyprowadzić wzór 2 na kolejne wyrazy w wierszu tablicy Romberga.

8. Metody adaptacyjne

Kolejnym pomysłem na całkowanie numeryczne są metody adaptacyjne. Poprzednie metody miały sztywno zdefiniowane węzły. W tej metodzie chcemy podać algorytmowi tylko przedział całkowania, funkcję oraz dopuszczalny błąd. Będziemy rozmieszczać węzły mniej gęsto tam, gdzie funkcja nie wykazuje dużej zmienności, natomiast gdy będzie ona bardzo zmienna, to wtedy umieścimy węzły gęściej, aby osiągnąć zadaną dokładność. Obecnie są to bardzo powszechnie stosowane metody,

ponieważ pozwalają próbkować tylko tam, gdzie jest to wymagane, co ogranicza złożoność. Jako metodę adaptacyjną można zaimplementować wiele metod pierwotnie ze statycznymi węzłami, np. metody Newtona-Cotesa. Przykładem jest także wariant kwadratury Gaussa, kwadratura Gaussa-Kronroda.

9. Inne bardziej specjalistyczne kwadratury

9.1. Monte Carlo

Monte Carlo jest metodą całkowania numerycznego używającą liczb pseudolosowych. W porównaniu do poprzednich metod, gdzie możemy przewidzieć gdzie będą węzły znając metodę i właściwości funkcji, tutaj są one rozmieszczone losowo. Jest to metoda szczególnie ważna przy całkach wielowymiarowych.

9.2. Kwadratura Gaussa-Laguerra

Kwadratura Gaussa-Laguerra jest rozszerzeniem kwadratury Gaussa, w celu badania całek oznaczonych następującej postaci.

$$\int_0^\infty e^{-x} f(x) dx \approx \sum_{i=1}^n w_i f(x_i)$$

gdzie x_i to i -ty pierwiastek wielomianu Laguerre'a $L_n(x)$, a wagi są dane wzorem

$$w_i = \frac{x_i}{(n+1)^2 [L_{n+1}(x_i)]^2}$$

9.3. Kwadratura Gaussa-Hermite'a

Kwadratura Gaussa-Hermite'a również rozwija pomysł kwadratury Gaussa na przedział $[-\infty, \infty]$. Aproksymuje ona całkę postaci

$$\int_{-\infty}^\infty e^{-x^2} f(x) dx \approx \sum_{i=1}^n w_i f(x_i)$$

Tym razem węzły x_i to zera wielomianu Hermite'a $h_n(x)$, a wagi są dane wzorem

$$w_i = \frac{2^{n-1} n! \sqrt{\pi}}{n^2 [H_{n-1}(x_i)]^2}$$

9.4. Funkcje oscylacyjne

Kolejnym problemem są całki postaci

$$I(f) = \int_a^b f(x) e^{i\omega g(x)} dx$$

gdzie $f(x)$ jest funkcją o małej zmienności, natomiast $e^{i\omega g(x)}$ jest funkcją oscylacyjną. Takimi całkami zajmują się kwadratury Filona czy Levina. Nie będę już rozwijał ich tematu, gdyż ponownie byłby to temat na kolejną pracownię.

10. Implementacja i testy

Do implementacji wybrałem wiele metod z różnych grup, aby sprawdzić jak zachowują się w przypadku różnych funkcji. Do testowania z metod Newtona-Cotes zaimplementowałem metodę trapezów, Simpsona oraz regułę 3/8. Z rodziny kwadratur Gaussowskich uwzględniłem Gaussa-Czebyszewa, Lobatto oraz Gaussa-Legendre'a. Z użyciem wielomianów Czebyszewa wybrałem kwadraturę Clenshawa-Curtisa. Za ciekawą uznałem również metodę korzystającą z ekstrapolacji Richardsona - metodę Romberga oraz implementację metody adaptacyjnej - adaptacyjna metoda Simpsona. Testy wykonywałem w podwójnej precyzji 64-bitowej według standardu IEEE 754, a otrzymane wyniki porównywałem z biblioteką QuadGK w precyzji 128-bitowej, która jest implementacją metody Gaussa-Kronroda. Testy wykonywałem na przedziale $[-1, 1]$.

10.1. Wielomiany

Na początku wykonałem testy na wielomianach 10 stopnia ze współczynnikami z rozkładu jednostajnego z przedziału $[-100, 100]$ (Tabela 3). Już na tym stosunkowo prostym przykładzie widać różnice między metodami. Metody Newtona-Cotesa mają bardzo małą dokładność, nawet przy większej ilości węzłów. Kwadratura Gaussa-Czebyszewa i kwadratura Lobatto radzą sobie od nich sporo lepiej, jednak wciąż dużo brakuje to dokładność bliskiej maszynowej w 64 bitach, czyli rzędu 10^{16} . natomiast kwadratura Clenshawa-Curtisa, Gaussa-Legendre'a, Romberga dla tablicy rozmiaru 6x6 oraz adaptacyjnego Simpsona z tolerancją tylko 10^{10} dają dokładność bliską maszynowej już dla najmniejszej liczby węzłów. Sprawdźmy teraz przypadki dla innych klas funkcji oraz przykładów danych w zadaniu.

metoda\węzły	10	100	1000	10 000
Trapez	3.702705e-01	1.412863e-03	2.533202e-04	2.453698e-04
Simpson	4.510322e-01	3.844638e-03	4.946672e-04	4.904801e-04
3/8	1.745032e-01	5.935442e-03	3.330790e-02	3.791611e-04
Gauss-Czebyszew	1.396928e-02	6.331863e-06	1.035600e-07	1.008909e-08
Lobatto	2.746119e-02	1.267547e-05	2.071185e-07	2.017819e-08
Clenshaw-Curtis	3.136850e-17	2.536929e-17	6.453085e-17	2.096803e-16
Gauss-Legendre	1.812661e-15	1.204292e-15	1.029117e-15	7.994914e-16
Romberg	9.415041e-19	4.606058e-17	4.108264e-17	5.028346e-17
adaptSimpson	3.306264e-17	4.124802e-16	5.853890e-17	7.163252e-17

Tabela 3: średnie błędy dla wielomianów 10 st. ze współczynnikami z rozkładu jednostajnego z przedziału $[-100, 100]$

10.2. Przykłady z zadania

W treści zadania dostaliśmy 3 różne funkcje.

$$f_1(x) = \frac{1}{x^4 + x^2 + 0.9}, \quad f_2(x) = \frac{2}{\pi(1 + x^2)}, \quad f_3(x) = \frac{2\pi(1 + x)}{(1 - x)(3 + x)} \sin(\pi(1 + x))$$

Dla funkcji $f_1(x)$ i $f_2(x)$ (Tabela 4 i 5) otrzymujemy podobne wyniki jak dla wielomianów, jedynie możemy zauważyć dla metody Romberga potrzebowalibyśmy większej tablicy dla dokładności maszynowej.

Problem pojawia dla funkcji $f_3(x)$. Obliczając analitycznie wartość funkcji w $x = 1$ poprzez skorzystanie z granicy $\lim_{x \rightarrow 0} \sin(x)/x = 1$ otrzymamy wartość $-\pi^3$. Niestety, obliczając numerycznie nie znamy tej granicy, a wartość funkcji w tym punkcie wyniesie ∞ . Zatem przesunę funkcję o epsilon, aby uniknąć nieskończoności.

W przypadku tej funkcji, dokładność jest dużo niższa niż w poprzednich przykładach (Tabela 6). Okazuje się, że istnieją funkcje, które pomimo, że są ciągłe, to obliczenie ich całek z dużą dokładnością jest trudniejszym problemem. Znacznie gorsze wyniki osiągnęła kwadratura Clenshawa-Curtisa, bo tylko około 10^{-8} dla 10 000 węzłów. Tutaj poradziła sobie metoda adaptacyjna oparta na kwadraturze Simpsona, funkcja biblioteczna używająca kwadratury Gaussa-Kronroda oraz kwadratura Gaussa-Legendre'a. Wszystkie te 3 metody pomimo doboru funkcji osiągnęły dokładność bliską maszynowej.

Tabela 4: błędy dla $f_1(x)$

metoda\węzły	100	10 000
Trapez	4.328686e-03	3.584618e-05
Simpson	8.900028e-03	8.719296e-05
3/8	5.030358e-03	4.365063e-05
Gauss-Czebyszew	1.792194e-05	1.792463e-09
Lobatto	3.584618e-05	3.584925e-09
Clenshaw-Curtis	8.464237e-17	1.960301e-16
Gauss-Legendre	1.459056e-15	6.459873e-16
Romberg	2.675358e-08	2.675358e-08
adaptSimpson	2.751031e-15	2.751031e-15

Tabela 5: błędy dla $f_2(x)$

100	10 000
6.344977e-03	6.365986e-05
1.286057e-02	1.273367e-04
6.827492e-03	6.370656e-05
2.617843e-05	2.617994e-09
5.235816e-05	5.235988e-09
4.626746e-17	4.626746e-17
1.045468e-15	8.419110e-16
6.938956e-09	6.938956e-09
5.726892e-15	5.726892e-15

Tabela 6: błędy dla $f_3(x)$

metoda\węzły	100	10 000
Trapez	3.582121e-02	3.583858e-04
Simpson	4.199033e-02	4.210690e-04
3/8	1.662193e-01	1.684149e-03
Gauss-Czebyszew	8.658536e-05	8.658103e-09
Lobatto	1.731670e-04	1.731620e-08
Clenshaw-Curtis	2.844921e-04	2.844637e-08
Gauss-Legendre	5.692125e-15	2.184109e-16
Romberg	5.416793e-02	5.416793e-02
adaptSimpson	5.404797e-15	5.405016e-15

10.3. Funkcje trygonometryczne

Kolejną rodziną funkcji są funkcje trygonometryczne. Wybrałem tutaj dwie funkcje złożone jako reprezentację tej klasy funkcji.

$$g_1(x) = a_1 \sin(a_2 \sin(a_3 \cos(\frac{x + a_4}{a_5})))$$

$$g_2(x) = b_1 \cos(b_2)^2 + \sin(b_3 * \tan(x))$$

gdzie a_i i b_i , $i = 1 : 5$ należą do rozkładu normalnego z przedziału $[-100, 100]$

Tabela 7: błędy dla $g_1(x)$

metoda\węzły	100	10 000
Trapez	7.117419e-01	4.240346e-03
Simpson	5.740519e-01	8.050357e-03
3/8	7.978338e-01	1.959444e-03
Gauss-Czebyszew	2.588979e-01	1.729585e-07
Lobatto	3.204052e-01	3.459100e-07
Clenshaw-Curtis	3.200404e-01	1.139351e-12
Gauss-Legendre	3.257299e-01	4.091469e-13
Romberg	2.269444e-15	9.182785e-14
adaptSimps	2.033180e-12	1.343188e-13

Tabela 8: błędy dla $g_2(x)$

100	10 000
1.000000e-02	1.000000e-04
2.000000e-02	2.000000e-04
5.076974e-03	3.807762e-04
4.112454e-05	4.112335e-09
8.224806e-05	8.224670e-09
8.812342e-16	1.824613e-16
1.049250e-15	5.776260e-16
4.292083e-16	2.220348e-15
9.711752e-17	3.009943e-17

Tutaj osiągamy podobne wyniki jak przy wielomianach, przy czym dla funkcji $g_1(x)$ nie są one tak zadowalające jakbyśmy chcieli dla 10 000 węzłów, dla funkcji $g_2(x)$ 100 węzłów jest już wystarczające. Funkcja $g_1(x)$ okazuje się trudna do efektywnego całkowania, gdyż 100 węzłów dla Gaussa-Legendre'a zapewnia dokładność tylko rzędu 10^{-1} . Tutaj trzeba byłoby użyć większej ilości węzłów.

10.4. Funkcje wykładnicze

Warto również poruszyć temat funkcji wykładniczych.

$$e_1(x) = c_1 e^x + c_2 2^x + \left| \frac{c_3}{10} \right|^x$$

$$e_2(x) = \pi^x + c_1 + |c_2| x^5 + \left| \frac{c_3}{10} \right|^x$$

Tabela 9: błędy dla $e_1(x)$

metoda\węzły	100	10 000
Trapez	1.316369e-02	1.313069e-04
Simpsons	2.606245e-02	2.625871e-04
3/8	7.923879e-02	8.309054e-04
Gauss-Czebyszew	5.400152e-05	5.399641e-09
Lobatto	1.079987e-04	1.079928e-08
Clenshaw-Curtis	3.032431e-17	2.320020e-16
Gauss-Legendre	6.473173e-19	9.607864e-16
Romberg	1.227842e-15	2.300136e-15
adaptSimpson	3.505329e-17	6.987941e-18

Tabela 10: błędy dla $e_2(x)$

100	10 000
1.457350e-02	1.471936e-04
2.874861e-02	2.943447e-04
3.648772e-01	2.490942e-03
5.973184e-05	6.052875e-09
1.194576e-04	1.210575e-08
1.914495e-17	1.481593e-16
1.011759e-15	7.904739e-16
2.618418e-15	2.131718e-15
1.072257e-16	1.383822e-16

Tutaj wybrane funkcje również mają losowane współczynniki z rozkładu normalnego, a ich wyniki dają podobne wnioski jak wcześniejsze testy.

10.5. Testy efektywności FFT

Na koniec porównam skuteczność zastosowania szybkiej transformaty Fouriera dla kwadratury Clenshawa-Curtisa. Do testów wziąłem funkcję $f(x) = e^{-x^2}$, ostatnią przykładową funkcję z zadania $f_3(x)$ oraz jedną z funkcji trygonometrycznych, również wcześniej testowaną.

Tabela 11: czas działania dla $f(x)$, w sek.

węzły	bez FFT	z FFT
512	0.002991	0.001016
2048	0.037903	0.003316
8192	0.569408	0.022323
32768	10.145807	0.162157
65536	41.925297	0.261693
131072	326.514293	0.566005

Tabela 12: czas działania dla $f_3(x)$, w sek.

węzły	bez FFT	z FFT
512	0.004998	0.000841
2048	0.077873	0.005200
8192	1.199181	0.020526
32768	20.311407	0.167805
65536	84.607258	0.305183
131072	345.382358	0.671152

Tabela 13: czas działania dla $g_1(x)$, w sek.

węzły	bez FFT	z FFT
512	0.004679	0.000922
2048	0.075688	0.005372
8192	1.180448	0.026525
32768	19.951047	0.174046
65536	83.672176	0.441867
131072	344.547595	0.758619

Testy pokazują ogromną przewagę FFT nad zwykłym mnożeniem wyrazów ze złożonością kwadratową. Już dla ilości węzłów do kilku tysięcy otrzymujemy jeden rząd wielkości na korzyść FFT. Dla ponad 100 000 węzłów kwadratura Clenshawa-Curtisa bez FFT liczy się już ponad 5 minut, natomiast wyniki z FFT widzimy jeszcze przed upływem sekundy. Łatwo zauważyć, że zmiana czasu wykonywania obliczeń pokrywa się ze wcześniej podaną złożonością czasową, odpowiednio $O(n^2)$ oraz $O(n \log n)$.

Literatura

- [1] *QuadGK.jl*, biblioteka pochodząca z JuliaMath
- [2] David Kincaid, Ward Cheney. *Analiza numeryczna*. Wydawnictwo Naukowo-Techniczne, Warszawa, 2002.
- [3] G. Dahlquist, A. Björck *Numerical Methods in Scientific Computing volume 1*, Society for Industrial and Applied Mathematics, Philadelphia 2008
- [4] Rafał Nowak, *wykłady z Analizy Numerycznej (M)*, semestr zimowy roku akademickiego 2019/2020
- [5] Nick Trefethen. *Is Gauss quadrature better than Clenshaw-Curtis?*. Oxford University, paper submitted to SIAM Review).
- [6] Omar A. AL-Sammarraie, Mohammed Ali Bashir *Error Analysis of the High Order Newton Cotes Formulas*. Omdurman Islamic University, Neelain University, 2015.