Metody numeryczne
Laboratorium 5
Kwadratury

1 Część teoretyczna

Całka jest jednym z najważniejszych pojęć współczesnej analizy matematycznej. Stosuje się ją w matematyce, fizyce, technice i wielu innych dziedzinach nauki. W matematyce badaniem własności i obliczaniem wartości całek zajmuje się dział zwany rachunkiem całkowym.

1.1 Całka nieoznaczona

Całką nieoznaczoną nazywamy odwrotność pochodnej danej funkcji.

$$F'(x) = F(x) \tag{1}$$

Funkcję $F(x)$ nazywamy **funkcja pierwotną** danej funkcji $f(x)$. Całkę nieoznaczoną opisujemy:

$$\int f(x)dx = F(x) + C \tag{2}$$

1.2 Całka oznaczona

Całka oznaczona to pole powierzchni między wykresem funkcji $f(x)$ w pewnym przedziale $< a, b >$.
Całka oznaczona funkcji $f(x)$ w przedziale $< a, b >$ oznaczana jest symbolem:

$$\int_a^b f(x)dx \tag{3}$$

Jeśli znamy całkę nieoznaczoną $F(x)$ funkcji podcałkowej $f(x)$, to całkę oznaczoną funkcji $f(x)$ w przedziale $< a, b >$ możemy obliczyć korzystając z twierdzenia Newtona-Leibniza:

$$\int_a^b f(x)dx = F(b) - F(a) \tag{4}$$

Nie stosujemy jednak tego sposobu w obliczeniach komputerowych, ponieważ znalezienie całki nieoznaczonej jest zadaniem numerycznie skomplikowanym.

W obliczeniach numerycznych stosuje się definicję Riemanna, w ktorej całka oznaczona jest przedstawiana jako suma pól obszarów ograniczonych wykresem funkcji $f(x)$ oraz osią OX. Elementy znajdujące się pod osią OX mają wartość ujemną.

Podzielmy przedział całkowania na bardzo dużo małych przedziałów $< x_{i-1}, x_i >$. Wewnątrz każdego z tych przedziałów wybierzmy dowolny punkt t_i taki, że $(x_{i-1} < t_i < x_i)$. Całka oznaczona Riemanna może być zinterpetowana jako suma wielu bardzo wąskich prostokątów o podstawie równej $(x_i - x_{i-1})$ i wysokości $f(t_i)$, czyli

$$\int_{x_a}^{x_b} f(x)dx = \lim_{n\to\infty} \sum_{i=1}^{n} f(t_i)(x_i - x_{i-1}) \tag{5}$$

Gdy odległości pomiędzy punktami podziałowymi zbliżają się do zera, suma pól prostokątów dąży do pola obszaru ograniczonego wykresem funkcji.

1.3 Kwadratury Newtona-Cotesa

Metoda prostokątów

W tej metodzie korzystamy z definicji uprzednio opisanej całki oznaczonej Riemanna.
Przedział całkowania $< x_a, x_b >$ dzielimy na n równo odległych punktów $x_1, x_2, ..., x_n$. Punkty te wyznaczamy w prosty sposob wg wzoru:

dla $i = 1, 2, ..., n$

$$x_i = x_a + \left(i - \frac{1}{2}\right) h, \tag{6}$$

gdzie $h = \dfrac{x_b - x_a}{n}$, co jest równe odległości pomiędzy sąsiednimi punktami x_{i-1}, x_i.

Dla każdego wyznaczonego w ten sposób punktu obliczamy wartość funkcji $f(x)$ w tym punkcie:

$$f_i = f(x_i), \text{ dla } i = 1, 2, ..., n \tag{7}$$

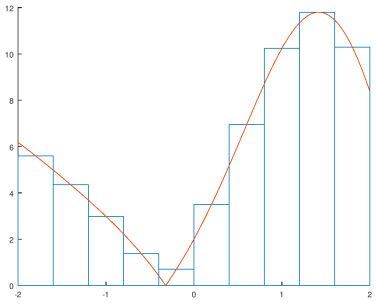
Suma pól prostokątów o boku h oraz $f(x_i)$ jest przybliżoną wartością całki oznaczonej funkcji $f(x)$ w przedziale $< x_a, x_b >$.

$$\int_{x_a}^{x_b} f(x)dx \approx h \sum_{i=1}^{n} f(x_i) \tag{8}$$

Metoda jest bardzo prosta w implementacji:

Dane: f, xa, xb, n
Wyniki: I
  funkcja f_rectI(f, xa, xb, n)
    $h \leftarrow (xb - ax)/n$
    $i \leftarrow 1, 2, ..., n$
    $X \leftarrow xa + (i - 1/2) * h$
    $Y \leftarrow f(X)$
    $I \leftarrow h * sum(Y)$

Algorytm 1: Złożony wzór prostokątów



Metoda trapezów

Popdobnym do uprzednio zaprozentowanego rozwiązania sposobem jest zastosowanie zamiast prostokątów - trapezów o wysokości h i podstawach równych odpowiednio wartości funkcji w punktach krańcowych. Dalsza kolejność działania nie odbiega od metody prostokątów.
Przedział całkowania $< x_a, x_b >$ dzielimy na n równo odległych punktów $x_1, x_2, ..., x_n$. Punkty te wyznaczamy w prosty sposób wg wzoru: dla $i = 0, 1, 2, ..., n$

$$x_i = x_a + ih, \tag{9}$$

gdzie $h = \frac{x_b - x_a}{n}$, co jest równe odległości pomiędzy sąsiednimi punktami x_{i-1}, x_i. Dla każdego wyznaczonego w ten sposób punktu obliczamy wartość funkcji $f(x)$ w tym punkcie:

$$f_i = f(x_i), \text{ dla } i = 0, 1, 2, ..., n \tag{10}$$

Wartocią przybliżenia jest suma n trapezów o podstawach $f(x_{i-1})$ oraz $f(x_i)$ i o wysokości h. Uproszczony i uporządkowany wzór prezentuje się następująco:

$$\int_{x_a}^{x_b} f(x)dx = h\left(\sum_{i=1}^{n-1} f(x_i) + \frac{f(x_a) + f(x_b)}{2}\right) \tag{11}$$

Dane: f, xa, xb, n
Wyniki: I
funkcja f_trapI(f, xa, xb, n)
$h \leftarrow (xb - ax)/n$
$i \leftarrow 0, 1, 2, ..., n$
$X \leftarrow xa + i * h$
$Y \leftarrow f(X)$
$I \leftarrow h * (sum(Y(1 : n - 1)) + (Y(0) + Y(n))/2)$

Algorytm 2: Złożony wzór trapezów

Metoda Simpsona

Metoda Simpsona w odróżnieniu do poprzednich algorytmów wykorzystuje do przybliżeń parabole. Zamiast pól prostokątów czy trapezów będziemy obliczań pola wycinków pod parabolą.
Przedział całkowania $< x_a, x_b >$ dzielimy na $n+1$ równo odległych punktów $x_0, x_1, x_2, ..., x_n$:

dla $i = 0, 1, 2, ..., n$

$$x_i = x_a + ih, \tag{12}$$

gdzie $h = \dfrac{x_b - x_a}{n}$, co jest równe odległości pomiędzy sąsiednimi punktami x_{i-1}, x_i. Dla każdych dwóch sąsiednich punktów wyznaczamy punkt środkowy t_i wg wzoru:

dla $i = 1, 2, ..., n$

$$t_i = \frac{x_{i-1} + x_i}{2} \tag{13}$$

Dla każdego wyznaczonego w ten sposób punktu obliczamy wartość funkcji f(x) w tym punkcie:

$$\begin{array}{cc} \text{dla } i = 0, 1, 2, ..., n & \text{dla } i = 1, 2, ..., n \\ f_i = f(x_i) & f_{t_i} = f(t_i) \end{array} \tag{14}$$

W każdym podprzedziale $< x_{i-1}, x_i >$ przybliżamy funkcję za pomocą paraboli. Ostatecznie uporządkowany wzór pozwala na obliczenie sum pól wycinków pod parabolami przybliżającymi funkcję f(x) w następujący sposób:

$$\int_{x_a}^{x_b} f(x)dx = \frac{h}{6}\left(2\sum_{i=1}^{n-1} f(x_i) + 4\sum_{i=1}^{n} f(t_i) + f(x_a) + f(x_b)\right) \tag{15}$$

Dane: f, xa, xb, n
Wyniki: I
funkcja f_SimpI(f, xa, xb, n)
$h \leftarrow (xb - xa)/n$
$i \leftarrow 0, 1, 2, ..., n$
$X \leftarrow xa + i * h$
$j \leftarrow 1, 2, ..., n$
$T \leftarrow (X(j-1) + X(j))/2$
$Y_x \leftarrow f(X)$
$Y_t \leftarrow f(T)$
$I \leftarrow h/6 * (2 * sum(Y_x(1 : n-1)) + 4 * sum(Y_t) + Y_x(0) + Y_x(n))$

Algorytm 3: Złożony wzór parabol

1.4 Całkowanie metodą Monte Carlo

Metoda Monte Carlo jest popularną metodą symulacyjną stworzoną przez polsko-amerykańskiego matematyka – Stanisława Ulama. Jedną z najczęstszych aplikacji tej metody jest numeryczne całkowanie.

Jeśli chcemy policzyć całkę oznaczoną (3) możemy posłużyć się następującą metodą: Losujemy w przedziale całkowania n punktów z przedziału $[x_a, x_b]$, a następnie przybliżamy

$$\int_{x_a}^{x_b} f(x)dx \approx \frac{x_b - x_a}{n} \sum_{i=1}^{n} f(x_i)dx \tag{16}$$

Algorytm tej metody zapisany w pseudokodzie ma następującą postać:

Dane: f, xa, xb, n
Wyniki: I
  funkcja f_crudeMonteC(f, xa, xb, n)
    $h \leftarrow (xb - xa)/n$;
    wylosuj n dowolnych punktów X z $< xa, xb >$
    $Y \leftarrow f(X)$
    $I \leftarrow Y \cdot h$

Algorytm 4: Metoda Crude Monte Carlo

Istnieje również algorytm Monte Carlo z zastosowaniem metody akceptacji i odrzuceń. Tutaj wartość całki obliczana jest jako ułamek pola powierzchni prostokąta o bokach $max(f(x))$ oraz $(xb - xa)$. Ułamek ten jest jednocześnie prawdopodobieństwem, że losowy punkt na tym obszarze znajdzie się pod wykresem funkcji $f(x)$:

Dane: f, xa, xb, n
Wyniki: I
  funkcja f_monteC(f, xa, xb, n)
    wylosuj n dowolnych punktów X z $< xa, xb >$
    $F_X \leftarrow f(X)$
    $F_{max} \leftarrow max(F_X)$
    wylosuj n dowolnych punktów Y z $< 0, F_{max} >$
    $k \leftarrow sum(Y < F_X)$
    $I \leftarrow k/n * (xb - xa) * F_{max}$

Algorytm 5: Metoda "typowe"Monte Carlo (akceptacji i odrzuceń)

1.5 Kwadratury Gaussa

Kwadratura Gaussa zastała stworzona w celu uzyskania dokładnego wyniku dla wielomianów stopnia $2n-1$. Polega ona na odpowiednim doborze wag $w_1, w_2, \ldots, w_n$ i punktów x_i.
Jeśli za dziedzinę przyjmiemy zakres $< -1; 1 >$ to będziemy mogli zapisać ją jako:

$$\int_{-1}^{1} f(x)dx = \sum_{i=1}^{n} w_i f(x_i) \tag{17}$$

Stosując podstawianie, wzór (17) można przekształcić do postaci:

$$\int_{a}^{b} f(x)dx \approx \sum_{i=1}^{n} w_i \frac{b-a}{2} f\left(\frac{b+a}{2} + \frac{b-a}{2} t_i\right), \tag{18}$$

gdzie t_i są pierwiastkami wielomianu Legendre'a, a w_i są wagami danych pierwiastków i wymagają do obliczenia m.in. znajomości pochodnej wielomianu Legendre'a danego stopnia. Powyższa metoda nosi nazwę **Kwadratury Gaussa-Legendre'a**

1.6 Całkowanie w matlab / octave

Środowisko Matlab/Octave oferuje kilka różnych metod numerycznego obliczania całek. Najczęściej używane to: `quad` (wykorzystujące kwadratury Gaussa) i `quadl` (adaptacyjna kwadratura Gaussa-Lobatto). Ich wywołanie może wyglądać następująco:
`Q=quad(f,a,b,tol,trace)`;
`Q=quadl(f,a,b,tol,trace)`;
`f` – łańcuch zawierający nazwę funkcji, funkcja musi być umieszczona w odpowiednim m-pliku, musi zwracać wektor wartości, a jej argumentem jest wektor elementów.
`a,b` – przedział całkowania,
`tol` – wymagana fokładność oszacowania, domyślnie $tol \approx 1.5 * 10^-8 \cdot 10^{-3}$
`trace` – parametr opcjonalny, pozwala na rysowanie wykresu z węzłami kwadratury.

1.7 Metody adaptacyjne

Złożone wzory prostokątów, trapezów i parabol nie są optymalnymi algorytmami. Głównie ma to przyczynę w tym, że przedział całkowania dzielimy na równe części, nie niwelując przy tym błędów powstających w konkretnych przedziałach. Jak wiemy, błąd metody trapezów wyraża się wzorem:

$$I(f) - P_n(f) = -\frac{(b-a)^3}{12n^2} f^{(4)}(\xi_n), \tag{19}$$

W związku z tym prawdziwe jest, że

$$I(f) - P_2(f) = -\frac{(b-a)^3}{12} \frac{1}{4} f^{(4)}(\xi_2) \tag{20}$$

oraz

$$P_1(f) - P_2(f) = -\frac{(b-a)^3}{12} \left(f^{(4)}(\xi_1) - \frac{1}{4} f^{(4)}(\xi_2) \right) \tag{21}$$

Zakładając, że dla odpowiedznio małych zakresów $f^{(4)}(\xi_1) \approx f^{(4)}(\xi_2)$, można zapisać:

$$f^{(4)}(\xi_1) - \frac{1}{4} f^{(4)}(\xi_2) \approx \frac{3}{4} f^{(4)}(\xi_1) \tag{22}$$

zatem

$$I(f) - P_2(f) \approx \frac{1}{3}(P_1(f) - P_2(f)) \tag{23}$$

Oznacza to, że na danym przedziale można założyć, że różnica pomiędzy przybliżeniem na jednym oraz na dwóch przedziałach jest równa trzykrotności błędu na dwóch przedziałach. Jeśli ten błąd jest większy niż zakładana dokładność, przedział należy podzielić i ponownie obliczyć przybliżenie na nowo powstałych przedziałach, aż do uzyskania zakładanej dokładności.

Dane: f, xa, xb, tol
Wyniki: I
  funkcja f_adapt(f, xa, xb, tol)
    $P_1 \leftarrow f\_trapI(f, xa, xb, 1)$
    $P_2 \leftarrow f\_trapI(f, xa, xb, 2)$
    jeżeli $|P_1 - P2| < 3 * tol$
      $I \leftarrow P2$
    w przeciwnym razie
      $I \leftarrow f\_adapt(f, xa, (xa + xb)/2, tol/2) + f\_adapt(f, (xa + xb)/2, xb, tol/2)$

Algorytm 6: Metoda adaptacyjna

2 Ćwiczenia

UWAGA! PODCZAS WYKONYWANIA ĆWICZEŃ NIE ZAPOMNIJ, ŻE W PSEUDOKODZIE INDEKSY TABLIC ZACZYNAJĄ SIĘ OD 0, A W OCTAVE OD 1!

1. Zdefiniuj funkcję $y = x^3 + 2x^2 + 1$ w pliku **f1.m**
2. Utwórz skrypt **main5.m**, w którym:
   - określ początek i koniec przedziału całkowania $[0, 2]$ pod zmiennymi **a**, **b**,
   - określ ilość podprzedziałów całkowania jako 5 pod zmienną **n**,
   - narysuj funkcję podcałkową w przedziale całkowania – kolor czerwony.
3. Wykorzystaj funkcje wbudowane Octave'a (**quad**, **quadl**) opisane w punkcie (1.6) do obliczenia całki funkcji **f1** na przedziale **[a,b]**.
4. Korzystając z algorytmów 1 - 3 dokonaj implementacji kwadratur Newtona pod nazwami kolejno: **f_rectI.m**, **f_trapI.m** oraz **f_SimpI.m**.
5. W skrypcie **main5.m** wywołaj wymienione funkcje z parametrami w kolejności: **(funkcja całkowana, a, b, n)**
6. Porównaj wyniki wszystkich metod dla różnych **n**: np. 3, 7, 10.
7. Dokonaj implementacji prostej metody Crude Monte Carlo opisanej w algorytmie 4. Wykonaj metodę kilkukrotnie dla różnych n (np. 10, 100, 1000).
8. Dokonaj implementacji metody Monte Carlo opisanej w algorytmie 5:
   - zdefiniuj funkcję, która na przedziale **[a,b]** jest nieujemna, np. $y = x^2 + 2x + 1$ w pliku **f2.m**,
   - wykonaj metodę kilkukrotnie dla różnych n (np. 10, 100, 1000),
   - porównaj wyniki z metodą z punktu poprzedniego.
9. Porównaj wyniki, sprawdź wpływ parametru **tol** na wyniki działania metodach wbudowanych.
10. Korzystając ze wzoru (18) dokonaj implementacji metody Gaussa-Legendre'a. Dołączona funkcja **[t, w] = gauleg(n)** zwraca wartosci wezlow t_i oraz wag w_i dla podanego stopnia wielomianu **n** (wzór 30).
11. Porównaj wyniki zwracane przez powyższą funkcję z wbudowaną metodą **quad** dla różnych stopni **n** oraz tolerancji **tol**.
12. Korzystając z algorytmu 6 dokonaj implementacji metody adaptacyjnej przy użyciu wzoru trapezów.