

KSN — III FK — teoria 10

Całkowanie numeryczne

Zajmujemy się kwadraturami — tj. znajdowaniem numerycznie wartości całki oznaczonej I z funkcji $f(x)$ (bądź $q(x)$ z wagą $p(x)$) na przedziale $[a, b]$

$$I = \int_a^b f(x)dx = \int_a^b p(x)q(x)dx. \quad (1)$$

Często metody numerycznego całkowania funkcji bazują wprost na definicji całki Riemanna (czy jak kto woli na jej geometrycznej interpretacji) nie wykonując żądanego w definicji przejścia granicznego:

$$I = \lim_{\Delta_i \rightarrow 0} \sum_{i=1}^N f(\xi_i)\Delta_i, \quad \xi_i \in [x_i, x_{i+1}], \quad \Delta_i = x_{i+1} - x_i, \quad a \leq x_i \leq b. \quad (2)$$

Do przybliżonego znajdowania wartości całki (1) z (2) mamy:

$$I = \sum_{i=1}^N w_i f(x_i) \quad x_i \in [a, b], \quad (3)$$

przy czym współczynniki w_i nie zależą od $f(x)$ zaś x_i nazywamy węzłami kwadratury.

W najprostszym przypadku pole pod krzywą $y = f(x)$ przybliżamy sumą pól prostokątów (stąd nazwa metody) o długości „podstawy” Δ_i i „wysokości” $f(x_i)$, gdzie przedział całkowania $(b - a)$ podzieliliśmy na N (niekoniecznie równych) odcinków Δ_i . Wzór ten będzie dokładny dla funkcji stałej. Przybliżenie kolejnych kawałków funkcji podcałkowej trapezami (*metoda trapezów*) czy parabolami (*metoda Simpsona*) przy zachowaniu kroku tego samego kroku całkowania oczywiście poprawi dokładność obliczeń. Stopień wielomianu dla którego kwadratura jest dokładna nazywamy jej rzędem.

W przypadku, gdy węzły kwadratur są równoodległe $\Delta_i = \text{const} = (b - a)/N$ a funkcję podcałkową $f(x)$ zastępujemy wielomianem interpolacyjnym Lagrange’a z węzłami $a = x_0 < x_1 < \dots < x_N = b$ wzór (3) nazywamy zamkniętymi kwadraturami Newtona–Cotesa. W praktyce przedział całkowania rzadko przybliża się wielomianem wysokiego stopnia. Za zwyczaj przedział $[a, b]$ dzieli się na kilka podprzedziałów i w każdym z nich stosuje się kwadraturę niskiego rzędu (mówimy wówczas o kwadraturach złożonych).

Jeśli przy zadanej liczbie węzłów N będziemy poszukiwać węzłów x_i i współczynników w_i tak, by rząd kwadratury był jak największy — mówimy o konstrukcji kwadratur Gaussa. Gdy w równaniu (1) za przedział całkowania przyjmiemy $(-\infty, +\infty)$ a wagę $p(x) = \exp(-x^2)$ to mamy do czynienia z kwadraturami Gaussa–Hermite’a. Węzły kwadratur są wówczas zerami ortogonalnych wielomianów Hermite’a.

Geometryczną interpretację całki riemannowskiej można wykorzystać jeszcze na jeden sposób. Szukanie całki (1) jest równoważne poszukiwaniu pola powierzchni figury ograniczonej wykresem funkcji $y = f(x)$, osią OX oraz prostymi $x = a$ i $x = b$.

Jeśli ograniczymy naszą figurę prostokątem $y = \min f(x)$, $y = \max f(x)$, $x = a$ i $x = b$ to prawdopodobieństwo trafienia w nią przy losowym wyborze pewnego punktu wewnątrz naszego prostokąta dane jest stosunkiem jej pola powierzchni do pola powierzchni tego właśnie prostokąta. Liczba „trafień” do całkowitej liczby „strzałów” będzie więc przybliżać rzeczywistą wartość pola figury (a więc i całki (1)) tym lepiej, im więcej takich strzałów „oddamy”. Tak przedstawia się idea całkowania Monte Carlo w najprostszej postaci (metodą chybił-trafił, orzeł-reszka).

W bibliotece *Numerical Recipes* metody trapezów, Simpsona i Romberga implementowane są w procedurach: `qtrap`, `qsimp` i `qromb`.

Procedura

```
SUBROUTINE qsimp(func,a,b,s)
```

```
REAL a,b,func,s,EPS
```

```
...
```

wyznacza wartości całki `s` z funkcji `func` na przedziale `[a,b]`. Dokładność ich wyznaczenia reguluje parametr `EPS`.

Pseudoprzykładowe liczby o rozkładzie jednorodnym w przedziale $[0,1]$ generują procedury `ran0`, `ran1`, `ran2` i `ran3`.

Jedną z funkcji biblioteki *GSL* umożliwiających numeryczne całkowanie jest

```
int gsl_integration_qng (const gsl_function *f, double a, double b, double epsabs,  
                        double epsrel, double *result, double *abserr, size_t *neval)
```

Wykorzystuje ona schemat Gaussa-Kronroda do obliczenia całki funkcji f (przekazanej jako wskaźnik do obiektu typu `gsl_function`) na przedziale $[a; b]$. Parametry `epsabs` i `epsrel` oznaczają odpowiednio bezwzględną i względną wymaganą wartość dokładności (procedura zakończy się w momencie spełnienia jednego z tych warunków). Wynik umieszczany jest w `*result`, `*neval` zawierać będzie ilość wykonanych obliczeń wartości funkcji, a `abserr` oszacowanie błędu bezwzględnego.

Krzysztof Malarz & Maciej Wołoszyn, Kraków, 17 grudnia 2003