Analiza numeryczna

5. Kwadratury liniowe

Rafał Nowak

Rozważmy zbiór $\mathbb{F} \equiv \mathbb{F}[a, b]$ funkcji całkowalnych (ograniczonych i ciągłych prawie wszędzie w $[a, b]$). Funkcjonał liniowy I_p odwzorowujący $\mathbb{F}$ w zbiór liczb rzeczywistych $\mathbb{R}$ określamy następująco:

$$I_p(f) := \int_a^b p(x) f(x)\, dx \qquad (f \in \mathbb{F}), \tag{1}$$

gdzie *funkcja wagowa* $p \in \mathbb{F}$ jest nieujemna w $[a, b]$, znika w skończonej liczbie punktów tego przedziału.

Definicja

Kwadraturą liniową nazywamy funkcjonał Q_n określony następująco

$$Q_n(f) := \sum_{k=0}^{n} A_k f(x_k) \qquad (n > 0). \tag{2}$$

gdzie liczby $A_k \equiv A_k^{(n)}, (k = 0, 1, \ldots, n)$ – nazywamy *współczynnikami (wagami)*, a liczby $x_k \equiv x_k^{(n)}, (k = 0, 1, \ldots, n)$ – *węzłami kwadratury* Q_n. *Resztą kwadratury* Q_n nazywamy funkcjonał

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

Definicja

Mówimy, że kwadratura Q_n jest *rzędu* r, jeśli

(i) $R_n(f) = 0$ dla każdego wielomianu $f \in \Pi_{r-1}$,

(ii) istnieje taki wielomian $w \in \Pi_r \setminus \Pi_{r-1}$, że $R_n(w) \neq 0$.

Lemat

Jeśli kwadratura Q_n jest określona wzorem (2), to jej rząd nie przekracza $2n+2$.

Rozważamy kwadratury

$$Q_n(f) := I_p(L_n[f]), \tag{3}$$

gdzie $L_n[f]$ jest wielomianem interpolacyjnym dla funkcji f w punktach x_k. Wprowadźmy oznaczenie na *wielomian węzłowy*:

$$\omega(x) := (x - x_0)(x - x_1)\dots(x - x_n).$$

Współczynniki kwadratury interpolacyjnej wyrażają się wzorem

$$A_k := I_p(\lambda_k) := \int_a^b p(x)\lambda_k(x)\mathrm{d}x = \int_a^b p(x) \prod_{j=0,\, j\neq k}^{n} \frac{x - x_j}{x_k - x_j}\mathrm{d}x, \tag{4}$$

a reszta – wzorem

$$\begin{aligned} R_n(f) &= \int_a^b p(x)\omega(x) f[x, x_0, x_1, \dots, x_n]\mathrm{d}x \\ &= \frac{1}{(n+1)!}\int_a^b p(x)\omega(x) f^{(n+1)}(\xi_x)\mathrm{d}x \qquad (k = 0, 1, \dots, n). \end{aligned}$$

Ostatni wzór zachodzi przy założeniu, że $f \in C^{n+1}[a, b]$.

Kwadratury Newtona to kwadratury interpolacyjne z węzłami równoodległymi

$$x_k \equiv x_k^{(n)} := a + kh \qquad (k = 0, 1, \ldots, n;\ h := (b-a)/n), \tag{5}$$

stosowane do obliczenia całki (1) dla $p \equiv 1$, czyli całki

$$I(f) := \int_a^b f(x)\mathrm{d}x.$$

Zatem

$$Q_n(f) := \sum_{k=0}^{n} A_k f(a + kh),$$

gdzie zgodnie z wzorem (4)

$$A_k \equiv A_k^{(n)} = I(\lambda_k) = \int_a^b \lambda_k(x)\mathrm{d}x = \frac{h(-1)^{n-k}}{k!(n-k)!} \int_0^n \prod_{j=0,\, j\neq k}^{n} (t-j)\, dt \qquad (k = 0, 1, \ldots$$

Twierdzenie

Reszta R_n kwadratury Newtona-Cotesa wyraża się wzorem

$$R_n(f) = \begin{cases} \frac{f^{(n+1)}(\xi)}{(n+1)!} \displaystyle\int_a^b \omega(x)\mathrm{d}x & (n = 1, 3, \ldots), \\ \frac{f^{(n+2)}(\eta)}{(n+2)!} \displaystyle\int_a^b x\,\omega(x)\mathrm{d}x & (n = 2, 4, \ldots), \end{cases} \tag{6}$$

gdzie $\xi, \eta \in (a, b)$.

Wypadek $n = 1, 2$

W wypadku $n = 1$ kwadratura Newtona-Cotesa nosi nazwę *wzoru trapezów*. Mamy $h = b - a$, $x_0 = a$, $x_1 = b$, $A_0 = A_1 = h/2$,

$$Q_1(f) := \frac{b-a}{2}[f(a) + f(b)], \tag{7}$$

$$R_1(f) = \frac{f''(\xi)}{2!}\int_a^b (x-a)(x-b)\mathrm{d}x = -\frac{(b-a)^3}{12}f''(\xi) = -\frac{h^3}{12}f''(\xi). \tag{8}$$

Dla $n = 2$ otrzymujemy *wzór Simpsona*:

$$h = (b-a)/2, \quad x_0 = a, \quad x_1 = (a+b)/2, \quad x_1 = b,$$
$$A_0 = A_2 = h/3, \quad A_1 = 4h/3,$$

$$\begin{aligned} Q_2(f) &:= \frac{b-a}{6}[f(a) + 4f((a+b)/2) + f(b)], & (9)\\ R_2(f) &= \frac{f^{(4)}(\eta)}{4!}\int_a^b x(x-a)(x - \frac{a+b}{2})(x-b)\mathrm{d}x & (10)\\ &= -\frac{1}{90}\left(\frac{b-a}{2}\right)^5 f^{(4)}(\eta) = -\frac{h^5}{90}f^{(4)}(\eta). \end{aligned}$$

Złożony wzór trapezów

$$T_n(f) := h \sum_{k=0}^{n}{}'' f(t_k) \qquad \left(h := \frac{b-a}{n},\ t_k := a + kh\right), \tag{11}$$

Reszta R_n^T jest równa

$$R_n^T(f) = -n\frac{h^3}{12}f''(\xi) = -(b-a)\frac{h^2}{12}f''(\xi). \tag{12}$$

dla pewnego $\xi \in (a,b)$.

Złożony wzór trapezów

$$T_n(f) := h \sum_{k=0}^{n}{}'' f(t_k) \qquad \left(h := \frac{b-a}{n},\ t_k := a + kh\right), \tag{11}$$

Reszta R_n^T jest równa

$$R_n^T(f) = -n\frac{h^3}{12} f''(\xi) = -(b-a)\frac{h^2}{12} f''(\xi). \tag{12}$$

dla pewnego $\xi \in (a, b)$.

Twierdzenie (Euler-Maclaurin)

Jeśli funkcja f jest klasy $C^{2m+2}[a, b]$, to

$$R_n^T(f) = \frac{c_1}{n^2} + \frac{c_2}{n^4} + \ldots + \frac{c_m}{n^{2m}} + \frac{d(n)}{n^{2m+2}}, \tag{13}$$

gdzie

$$c_k := \frac{(b-a)^{2k} B_{2k}}{(2k)!} \left[f^{(2k-1)}(a) - f^{(2k-1)}(b)\right] \qquad (k = 1, 2, \ldots, m),$$

$d(n)$ jest ograniczoną funkcją zmiennej n: istnieje taka stała M, że dla każdego n zachodzi nierówność $|d(n)| \leqslant M$, a B_{2k} są tzw. liczbami Bernoulliego. (Np. $B_0 = 1$, $B_2 = 1/6$, $B_4 = -1/30$, $B_6 = 1/42$, $B_8 = -1/30$, $B_{10} = 5/66$).

Złożony wzór Simpsona

$$
\begin{aligned}
S_n(f) &:= \frac{h}{3}\{f(t_0)+4f(t_1)++2f(t_2)+4f(t_3)+2f(t_4)+\ldots \\
&\quad \ldots+2f(t_{2m-2})+4f(t_{2m-1})+f(t_{2m})\} \\
&= \frac{h}{3}\left\{2\sum_{k=0}^{m}{}'' f(t_{2k})+4\sum_{k=1}^{m} f(t_{2k-1})\right\} \\
&= \frac{1}{3}(4T_n-T_m) \qquad (n=2m,\ h=\frac{b-a}{n}).
\end{aligned} \tag{14}
$$

Rzeszta $R_n^S(f)$ jest równa

$$
R_n^S(f) = -m\frac{h^5}{90}f(\eta) = -(b-a)\frac{h^4}{180}f^{(4)}(\eta), \tag{15}
$$

gdzie $\eta \in (a,b)$.

$$h_k := (b-a)/2^k,$$
$$x_i^{(k)} := a + ih_k \qquad (i = 0, 1, .., 2^k),$$
$$T_{0k} := T_{2^k}(f) = h_k \sum_{i=0}^{2^k}{}'' f(x_i^{(k)}).$$

$$T_{mk} = \frac{4^m T_{m-1,k+1} - T_{m-1,k}}{4^m - 1} \qquad (k = 0, 1, \ldots;\ m = 1, 2, \ldots).$$

Tak więc, zaczynając od złożonych wzorów trapezów $T_{00}, T_{01}, T_{02}, \ldots$ budujemy trójkątną **tablicę Romberga** przybliżeń całki.

$$\begin{array}{cccccc}
T_{00} & & & & & \\
T_{01} & T_{10} & & & & \\
T_{02} & T_{11} & T_{20} & & & \\
T_{03} & T_{12} & T_{21} & T_{30} & & \\
\vdots & \vdots & \vdots & \vdots & \ddots & \\
T_{0m} & T_{1,m-1} & T_{2,m-2} & T_{3,m-3} & \ldots & T_{m0} \\
\vdots & \vdots & \vdots & \vdots & & \ddots
\end{array}$$

Można wykazać, że

1^o $T_{mk} = I - c_m^* h_k^{2m+2} - \ldots \qquad (k \geqslant 0;\ m \geqslant 1)$;

2^o $T_{mk} = \sum_{j=0}^{2^{m+k}} A_j^{(m)} f(x_j^{(m+k)}) \qquad (k \geqslant 0;\ m \geqslant 1)$ (elementy k-tego wiersza tablicy Romberga zawierają te same węzły, co T_{0k}), gdzie $A_j^{(m)} > 0$ $(j = 0, 1, \ldots, 2^{m+k})$;

3^o dla każdej pary k, m T_{mk} jest sumą Riemanna;

4^o każdy z wzorów T_{m0}, T_{m1}, ... jest kwadraturą rzędu $2m+2$;

5^o (wniosek z 2^o, 3^o, 4^o i z twierdzenia o zbieżności ciągu kwadratur o dodatnich współczynnikach) niech $I = I(f)$, gdzie f jest dowolną funkcją ciągłą w $[a, b]$; wówczas

$$\lim_{k\to\infty} T_{mk} = I \qquad (m = 1, 2, \ldots);$$
$$\lim_{m\to\infty} T_{mk} = I \qquad (k = 0, 1, \ldots).$$

Przykład

$$I = \int_1^3 \frac{dx}{x} = \ln 3 = 1.098612\ldots$$

$$\begin{array}{lllll}
T_{00} = 1.333333 & & & & \\
T_{01} = 1.166667 & T_{10} = 1.111111 & & & \\
T_{02} = 1.116667 & T_{11} = 1.100000 & T_{20} = 1.099259 & & \\
T_{03} = 1.103211 & T_{12} = 1.098726 & T_{21} = 1.098641 & T_{30} = 1.098631 & \\
T_{04} = 1.099768 & T_{13} = 1.098620 & T_{22} = 1.098613 & T_{31} = 1.098613 & T_{40} = 1.098613 \\
T_{05} = 1.098902 & T_{14} = 1.098613 & T_{23} = 1.098613 & T_{32} = 1.098613 & T_{41} = 1.098613 \\
T_{06} = 1.098685 & T_{15} = 1.098613 & & & \\
T_{07} = 1.098630 & T_{16} = 1.098612 & & &
\end{array}$$

Kwadratury Gaussa

$$I_p(f) := \int_a^b p(x) f(x)\,\mathrm{d}x \qquad (f \in \mathbb{F})$$

$$Q_n(f) := \sum_{k=0}^{n} A_k^{(n)} f(x_k^{(n)})$$

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

$$\omega(x) = \bar{P}_{n+1}(x) = (x-x_0)(x-x_1)\cdots(x-x_n)$$

Kwadratury Gaussa

$$I_p(f) := \int_a^b p(x) f(x)\,\mathrm{d}x \qquad (f \in \mathbb{F})$$

$$Q_n(f) := \sum_{k=0}^{n} A_k^{(n)} f(x_k^{(n)})$$

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

$$\omega(x) = \bar{P}_{n+1}(x) = (x - x_0)(x - x_1) \cdots (x - x_n)$$

Lemat

$$A_k^{(n)} = \frac{a_{n+1}}{a_n} \cdot \frac{\|P_n\|^2}{P'_{n+1}(x_k)\, P_n(x_k)}$$

$$P_j(x) = a_j x^k + \ldots$$

Kwadratury Gaussa

Lemat

Współczynniki kwadratury Gaussa są dodatnie, tzn.

$$A_k^{(n)} > 0, \qquad k = 0, 1, \ldots, n.$$

Kwadratury Gaussa

Lemat

Współczynniki kwadratury Gaussa są dodatnie, tzn.

$$A_k^{(n)} > 0, \qquad k = 0, 1, \ldots, n.$$

Lemat

Jeśli $f \in C^{2n+2}[a, b]$*. to reszta kwadratury Gaussa wyraża się wzorem*

$$R_n(f) := I_p(f) - Q_n(f) = \frac{f^{(2n+2)}(\xi)}{(2n+2)!\, a_{n+1}^2} \int_a^b p(x) [P_{n+1}(x)]^2 \,\mathrm{d}x.$$

Lemat

Jeśli $f \in C[a, b]$*, to* $\lim_{n \to \infty} Q_n(f) = I_p(f)$*.*

Kwadratury Gaussa

Związek rekurencyjny dla wielomianów ortogonalnych P_k

$$P_k(x) = (b_k x + c_k)P_{k-1}(x) - d_k P_{k-2}(x),$$

można zapisać macierzowo

$$x\boldsymbol{p}(x) = A\boldsymbol{p}(x) + \frac{1}{b_{n+1}} P_{n+1}(x)\boldsymbol{e}_{n+1}$$

$$\boldsymbol{p}(x) = [P_0(x), P_1(x), \ldots, P_n(x)]^T, \quad \boldsymbol{e}_{n+1} = [0, 0, \ldots, 0, 1]^T \in \mathbb{R}^{n+1}$$

$$A = \{a_{ij}\}, \quad a_{ij} = \begin{cases} -c_i/b_i, & i = j, \\ d_i/b_i, & i = j+1, \\ 1/b_i, & i = j-1, \\ 0, & \text{w p.p.} \end{cases}$$

Kwadratury Gaussa

Lemat

Węzły $x_k^{(n)}$ *kwadratury Czebyszewa są wartościami własnymi macierzy* A*, a współczynniki wyrażają się wzorami*

$$A_k^{(n)} = [\boldsymbol{v}_1^{(k)}]^2 \int_a^b p(x)\,\mathrm{d}x,$$

gdzie $\boldsymbol{v}^{(k)}$ *jest wektorem własnym odpowiadającym wartości* $x_k^{(n)}$.

Kwadratury Gaussa

Lemat

Węzły $x_k^{(n)}$ *kwadratury Czebyszewa są wartościami własnymi macierzy* A*, a współczynniki wyrażają się wzorami*

$$A_k^{(n)} = [\boldsymbol{v}_1^{(k)}]^2 \int_a^b p(x)\,\mathrm{d}x,$$

gdzie $\boldsymbol{v}^{(k)}$ *jest wektorem własnym odpowiadającym wartości* $x_k^{(n)}$*.*

Lemat

Macierz A *jest podobna do macierzy symetrycznej trójprzekątniowej* $T = \{t_{ij}\}$*, gdzie*

$$t_{ii} = -\frac{c_i}{b_i}, \qquad t_{i+1,i} = t_{i,i+1} = \left(\frac{d_{i+1}}{b_i b_{i+1}}\right)^{1/2}.$$

Kwadratury Gaussa-Legendre'a

$$P_k(x) = \frac{2k-1}{k} x P_{k-1}(x) - \frac{k-1}{k} P_{k-2}(x)$$

Kwadratury Gaussa-Legendre'a

$$P_k(x) = \frac{2k-1}{k} x P_{k-1}(x) - \frac{k-1}{k} P_{k-2}(x)$$

$$t_{ii} = 0, \qquad t_{i,i+1} = t_{i+1,i} = (4 - 1/i^2)^{-1/2}$$

Kwadratury Gaussa-Legendre'a

$$P_k(x) = \frac{2k-1}{k} x P_{k-1}(x) - \frac{k-1}{k} P_{k-2}(x)$$

$$t_{ii} = 0, \qquad t_{i,i+1} = t_{i+1,i} = (4 - 1/i^2)^{-1/2}$$

Implementacja kwadratury GL w Juli

```
# Funkcja oblicza całkę \int_{-1}^{1} f(x) dx
# za pomocą (n+1)-punktowej  kwadratury Gaussa-Legendre'a
function GaussLegendre(f,n)
  β = 1 ./ sqrt(4.0 -(1:n).^(-2.0));
  T = SymTridiagonal(zeros(n+1),β);    # Macierz podobna do macierzy
  x,V = eig(T);                        # x - wartości własne (pierwia
  w = 2.0*vec(V[1,:]).^2;              # w - współczynniki kwadratury
  return dot(w,f(x));
end;

# Przykładowe użycie
GaussLegendre(x -> sqrt(1-x.^2), 100);
```

Kwadratura Gaussa-Czebyszewa

$$I_p(f) = \int_{-1}^{1} p(x) f(x)\,\mathrm{d}x, \qquad p(x) = \frac{1}{\sqrt{1-x^2}}$$

$$Q_n^{GC}(f) := \int_{-1}^{1} p(x) I_n(x)\,\mathrm{d}x, \quad I_n(t_k) = f(t_k), \quad t_k = \cos\left(\frac{2k+1}{2n+2}\pi\right)$$

Lemat

$$I_n(x) = {\sum_{i=0}^{n}}{}' \alpha_i T_i(x), \qquad \alpha_i = \frac{2}{n+1} \sum_{k=0}^{n} f(t_k) T_i(t_k)$$

Kwadratura Gaussa-Czebyszewa

$$I_p(f) = \int_{-1}^{1} p(x) f(x)\,\mathrm{d}x, \qquad p(x) = \frac{1}{\sqrt{1-x^2}}$$

$$Q_n^{GC}(f) := \int_{-1}^{1} p(x) I_n(x)\,\mathrm{d}x, \quad I_n(t_k) = f(t_k), \quad t_k = \cos\left(\frac{2k+1}{2n+2}\pi\right)$$

Lemat

$$I_n(x) = \sum_{i=0}^{n}{}' \alpha_i T_i(x), \qquad \alpha_i = \frac{2}{n+1} \sum_{k=0}^{n} f(t_k) T_i(t_k)$$

Wniosek

$$Q_n^{GC}(f) = \sum_{k=0}^{n} A_k f(t_k), \qquad A_k = \frac{\pi}{n+1}.$$

Kwadratura Lobatto

$$I_p(f) = \int_{-1}^{1} p(x) f(x)\,\mathrm{d}x, \qquad p(x) = \frac{1}{\sqrt{1-x^2}}$$

$$Q_n^L(f) := \int_{-1}^{1} p(x) J_n(x)\,\mathrm{d}x, \quad J_n(u_k) = f(u_k), \quad u_k = \cos\left(\frac{k}{n}\pi\right)$$

Kwadratura Lobatto

$$I_p(f) = \int_{-1}^{1} p(x) f(x)\,\mathrm{d}x, \qquad p(x) = \frac{1}{\sqrt{1-x^2}}$$

$$Q_n^L(f) \coloneqq \int_{-1}^{1} p(x) J_n(x)\,\mathrm{d}x, \quad J_n(u_k) = f(u_k), \quad u_k = \cos\left(\frac{k}{n}\pi\right)$$

Lemat

$$J_n(x) = {\sum_{j=0}^{n}}'' \beta_j T_j(x), \qquad \beta_j = \frac{2}{n} {\sum_{k=0}^{n}}'' f(u_k) T_j(u_k)$$

Kwadratura Lobatto

$$I_p(f) = \int_{-1}^{1} p(x) f(x)\,\mathrm{d}x, \qquad p(x) = \frac{1}{\sqrt{1-x^2}}$$

$$Q_n^L(f) := \int_{-1}^{1} p(x) J_n(x)\,\mathrm{d}x, \quad J_n(u_k) = f(u_k), \quad u_k = \cos\left(\frac{k}{n}\pi\right)$$

Lemat

$$J_n(x) = {\sum_{j=0}^{n}}'' \beta_j T_j(x), \qquad \beta_j = \frac{2}{n} {\sum_{k=0}^{n}}'' f(u_k) T_j(u_k)$$

Wniosek

$$Q_n^L(f) = {\sum_{k=0}^{n}}'' A_k f(u_k), \qquad A_k = \frac{\pi}{n}.$$

Kwadratura Lobatto

Lemat

Wzór

$$\int_{-1}^{1} \frac{1}{\sqrt{1-x^2}} f(x)\,\mathrm{d}x \approx \frac{\pi}{n} \sum_{k=0}^{n}{}'' f(u_k)$$

jest dokładny dla $f \in \Pi_{2n-1}$.

Implementacja kwadratury Gaussa-Czebyszewa i Lobatto w Juli

```
function GaussChebyshev(f,n)
  x = cos(collect(1:2:2*n+1)*π/(2*n+2)); # węzły kwadratury Gaussa-(
  return π/(n+1)*sum(f(x));
end;

function Lobatto(f,n)
  x = cos(collect(0:n)*π/n); # węzły kwadratury Lobatto (punkty eks
  y = f(x); y[1] *= 0.5; y[n+1] *= 0.5;
  return π/n*sum(y);
end;


# Przykładowe użycie
GaussChebyshev(x -> (1-x.^2), 1000);
Lobatto(x -> (1-x.^2), 1000);
```

Szereg Czebyszewa

Niech $f \in C^1[-1,1]$. Wówczas funkcję f można rozwinąć w szereg Czebyszewa

$$f(x) = {\sum_{k=0}^{\infty}}' a_k T_k(x) \qquad (-1 \leqslant x \leqslant 1),$$

$$a_k \equiv a_k[f] := \frac{2}{\pi} \int_{-1}^{1} \frac{1}{\sqrt{1-x^2}} f(x) T_k(x)\, \mathrm{d}x.$$

Szereg Czebyszewa

Niech $f \in C^1[-1,1]$. Wówczas funkcję f można rozwinąć w szereg Czebyszewa

$$f(x) = \sum_{k=0}^{\infty}{}' a_k T_k(x) \qquad (-1 \leqslant x \leqslant 1),$$

$$a_k \equiv a_k[f] := \frac{2}{\pi} \int_{-1}^{1} \frac{1}{\sqrt{1-x^2}} f(x) T_k(x) \, \mathrm{d}x.$$

Współczynniki Czebyszewa a_k obliczamy w sposób przybliżony

$$a_k \approx \alpha_k^n := \frac{2}{\pi} Q_n^{GC}(f \cdot T_k) = \frac{2}{n+1} \sum_{j=0}^{n} f(t_j) T_k(t_j),$$

$$a_k \approx \beta_k^n := \frac{2}{\pi} Q_n^{L}(f \cdot T_k) = \frac{2}{n} \sum_{j=0}^{n}{}'' f(u_j) T_k(u_j),$$

Szereg Czebyszewa

Lemat

Jeśli $f = \sum_{k=0}^{\infty}{}' a_k T_k$, *to zachodzą wzory*

$$\beta_k^n = a_k + \sum_{i=1}^{\infty} (a_{2in-k} + a_{2in+k}) \qquad (k = 0, 1, \ldots, n-1), \tag{16}$$

$$\beta_n^n = a_n + \sum_{i=1}^{\infty} a_{(2i+1)n}. \tag{17}$$

Wzory te mówią, że jeśli ciąg $\{a_k\}$ dąży dostatecznie regularnie do zera, to równość przybliżona $\beta_k^n \approx a_k$ jest obarczona niewielkim błędem, wyrażającym się przez współczynniki $a_m[f]$ dla $m > n$:

$$\begin{aligned}
\beta_0^n &= a_0 + 2a_{2n} + 2a_{4n} + \ldots &&\approx a_0 + 2a_{2n},\\
\beta_{n-1}^n &= a_{n-1} + a_{n+1} + a_{3n-1} + \ldots &&\approx a_{n-1} + a_{n+1},\\
&\ldots\ldots\ldots\ldots\ldots\ldots\\
\beta_{n-2}^n &= a_{n-2} + a_{n+2} + a_{3n-2} + \ldots &&\approx a_{n-2} + a_{n+2},\\
\beta_n^n &= a_n + a_{3n} + a_{5n} + \ldots &&\approx a_n + a_{3n}.
\end{aligned}$$

Kwadratura Clenshawa-Curtisa

$$I(f) = \int_{-1}^{1} f(x)\,\mathrm{d}x, \qquad f(x) = {\sum_{k=0}^{\infty}}' a_k T_k(x)$$

$$a_k = \frac{2}{\pi}\int_{-1}^{1} \frac{1}{\sqrt{1-x^2}} f(x) T_k(x)\,\mathrm{d}x, \qquad k \geqslant 0$$

Kwadratura Clenshawa-Curtisa

$$I(f) = \int_{-1}^{1} f(x)\,\mathrm{d}x, \qquad f(x) = {\sum_{k=0}^{\infty}}' a_k T_k(x)$$

$$a_k = \frac{2}{\pi}\int_{-1}^{1} \frac{1}{\sqrt{1-x^2}} f(x) T_k(x)\,\mathrm{d}x, \qquad k \geqslant 0$$

$$Q_n(f) := \int_{-1}^{1} \left({\sum_{k=0}^{n}}' a_k T_k(x) \right) \mathrm{d}x$$

Kwadratura Clenshawa-Curtisa

$$I(f) = \int_{-1}^{1} f(x)\,\mathrm{d}x, \qquad f(x) = {\sum_{k=0}^{\infty}}' a_k T_k(x)$$

$$a_k = \frac{2}{\pi} \int_{-1}^{1} \frac{1}{\sqrt{1-x^2}} f(x) T_k(x)\,\mathrm{d}x, \qquad k \geqslant 0$$

$$Q_n(f) := \int_{-1}^{1} \left({\sum_{k=0}^{n}}' a_k T_k(x) \right) \mathrm{d}x = 2 {\sum_{k=0}^{\lfloor n/2 \rfloor}}' \frac{a_{2k}}{1-4k^2}$$

Interpolacyjna kwadratura Clenshawa-Curtisa

$$I(f) = \int_{-1}^{1} f(x)\,\mathrm{d}x$$

Interpolacyjna kwadratura Clenshawa-Curtisa

$$I(f) = \int_{-1}^{1} f(x)\,\mathrm{d}x$$

$$Q_n^{CC}(f) := \int_{-1}^{1} J_n(x)\,\mathrm{d}x, \qquad J_n = {\sum_{j=0}^{n}}'' \beta_j T_j$$

Interpolacyjna kwadratura Clenshawa-Curtisa

$$I(f) = \int_{-1}^{1} f(x)\,\mathrm{d}x$$

$$Q_n^{CC}(f) := \int_{-1}^{1} J_n(x)\,\mathrm{d}x, \qquad J_n = \sum_{j=0}^{n}{}'' \beta_j T_j$$

$$Q_n^{CC}(f) = \sum_{k=0}^{n}{}'' A_k^{(n)} f(u_k), \qquad A_k^{(n)} := \frac{4}{n} \sum_{j=0}^{n/2}{}'' \frac{T_{2j}(u_k)}{1-4j^2}$$

Uwaga: w powyższym wzorze symbol $\sum_{j=0}^{n/2}{}''$ oznacza sumę, w której pierwszy składnik jest pomnożony przez $1/2$, zaś ostatni jest mnożony przez $1/2$ tylko, gdy n jest parzyste.

Interpolacyjna kwadratura Clenshawa-Curtisa

Z własności wielomianów Czebyszewa

$$T_j(u_k) = T_k(u_j),$$

mamy

$$\beta_j = \frac{2}{n} \sum_{k=0}^{n}{}'' f(u_k) T_j(u_k) = \frac{2}{n} \sum_{k=0}^{n}{}'' f(u_k) T_k(u_j),$$

a więc przybliżenia β_j współczynników Czebyszyszewa $a_j[f]$ można obliczać za pomocą algorytmu Clenshawa.

Interpolacyjna kwadratura Clenshawa-Curtisa

Z własności wielomianów Czebyszewa

$$T_j(u_k) = T_k(u_j),$$

mamy

$$\beta_j = \frac{2}{n} \sum_{k=0}^{n}{}'' f(u_k) T_j(u_k) = \frac{2}{n} \sum_{k=0}^{n}{}'' f(u_k) T_k(u_j),$$

a więc przybliżenia β_j współczynników Czebyszyszewa $a_j[f]$ można obliczać za pomocą algorytmu Clenshawa. Obliczając raz wektor współczynników $[f(u_0), f(u_1), \ldots, f(u_n)]^T$ wywołujemy $n+1$ razy algorytm Clenshawa, mianowicie z parametrem $t = u_0, u_1, \ldots, u_n$, w celu obliczenia współczynników $\beta_0, \beta_1, \ldots, \beta_n$.

Interpolacyjna kwadratura Clenshawa-Curtisa

Z własności wielomianów Czebyszewa

$$T_j(u_k) = T_k(u_j),$$

mamy

$$\beta_j = \frac{2}{n} \sum_{k=0}^{n}{}'' f(u_k) T_j(u_k) = \frac{2}{n} \sum_{k=0}^{n}{}'' f(u_k) T_k(u_j),$$

a więc przybliżenia β_j współczynników Czebyszyszewa $a_j[f]$ można obliczać za pomocą algorytmu Clenshawa. Obliczając raz wektor współczynników $[f(u_0), f(u_1), \ldots, f(u_n)]^T$ wywołujemy $n+1$ razy algorytm Clenshawa, mianowicie z parametrem $t = u_0, u_1, \ldots, u_n$, w celu obliczenia współczynników $\beta_0, \beta_1, \ldots, \beta_n$.Ostatecznie otrzymujemy metodę o złożoności $\mathcal{O}(n^2)$.

Interpolacyjna kwadratura Clenshawa-Curtisa

Wszystkie współczynniki

$$\beta_j = \frac{2}{n} \sum_{k=0}^{n}{}'' f(u_k) \cos(kj\pi/n), \qquad j = 0, 1, \ldots, n$$

można wyznaczyć w czasie $\mathcal{O}(n \log n)$ za pomocą szybkiej transformaty Fouriera (FFT).

Interpolacyjna kwadratura Clenshawa-Curtisa

Wszystkie współczynniki

$$\beta_j = \frac{2}{n} \sum_{k=0}^{n}{}'' f(u_k) \cos(kj\pi/n), \qquad j = 0, 1, \ldots, n$$

można wyznaczyć w czasie $\mathcal{O}(n \log n)$ za pomocą szybkiej transformaty Fouriera (FFT).

- algorytm FFT powstał w 1965 roku.

Interpolacyjna kwadratura Clenshawa-Curtisa

Wszystkie współczynniki

$$\beta_j = \frac{2}{n} \sum_{k=0}^{n}{}'' f(u_k) \cos(kj\pi/n), \qquad j = 0, 1, \ldots, n$$

można wyznaczyć w czasie $\mathcal{O}(n \log n)$ za pomocą szybkiej transformaty Fouriera (FFT).

- algorytm FFT powstał w 1965 roku.
- Clenshaw i Curtis opublikowali swoją metodę w 1960 roku.

Interpolacyjna kwadratura Clenshawa-Curtisa

Wszystkie współczynniki

$$\beta_j = \frac{2}{n} \sum_{k=0}^{n}{}'' f(u_k) \cos(kj\pi/n), \qquad j = 0, 1, \ldots, n$$

można wyznaczyć w czasie $\mathcal{O}(n \log n)$ za pomocą szybkiej transformaty Fouriera (FFT).

- algorytm FFT powstał w 1965 roku.
- Clenshaw i Curtis opublikowali swoją metodę w 1960 roku.
- Uwaga: Jeśli $n = 2^j$, to wszystkie węzły $u_k = \cos(k\pi/n)$ można wyznaczyć obliczając tylko $\mathcal{O}(\log n)$ wywołań funkcji cosinus.

Szybka transformata Fouriera

Problem $\boldsymbol{y} = DCT(N, \boldsymbol{x})$

Dane: $\boldsymbol{x} = [x_0, x_1, \ldots, x_{N-1}]$
Wynik: $\boldsymbol{y} = [y_0, y_1, \ldots, y_{N-1}]$, gdzie

$$y_k = \sum_{j=0}^{N-1} x_j \theta_N^{jk} \qquad (\theta_N = \exp(2\pi \mathrm{i}/N), \quad \mathrm{i} = \sqrt{-1}).$$

Szybka transformata Fouriera

Problem $\boldsymbol{y} = DCT(N, \boldsymbol{x})$

Dane: $\boldsymbol{x} = [x_0, x_1, \dots, x_{N-1}]$
Wynik: $\boldsymbol{y} = [y_0, y_1, \dots, y_{N-1}]$, gdzie

$$y_k = \sum_{j=0}^{N-1} x_j \theta_N^{jk} \qquad (\theta_N = \exp(2\pi \mathrm{i}/N), \quad \mathrm{i} = \sqrt{-1}).$$

Zakładamy, że $N = 2M$. Mamy

$$\begin{aligned} y_k &= \sum_{j=0}^{N-1} x_j \theta_N^{jk} = \sum_{j=0}^{M-1} x_j \theta_N^{jk} + \sum_{j=M}^{N-1} x_j \theta_N^{jk} \\ &= \sum_{j=0}^{M-1} x_j \theta_N^{jk} + \sum_{j=0}^{M-1} x_{M+j} \theta_N^{(M+j)k} = \sum_{j=0}^{M-1} x_j \theta_N^{jk} + \sum_{j=0}^{M-1} (-1)^k x_{M+j} \theta_N^{jk} \\ &= \sum_{j=0}^{M-1} \left(x_j + (-1)^k x_{M+j} \right) \theta_N^{jk} \qquad (18) \end{aligned}$$

$$y_k = \sum_{j=0}^{M-1} \left(x_j + (-1)^k x_{M+j} \right) \theta_N^{jk}, \qquad k = 0, 1, \ldots, N-1$$

$$y_k = \sum_{j=0}^{M-1} \left(x_j + (-1)^k x_{M+j}\right) \theta_N^{jk}, \qquad k = 0, 1, \ldots, N-1$$

Dla parzystych i nieparzystych wskaźników otrzymujemy wzory

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

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

Stąd widzimy, że wektory $[y_0, y_2, \ldots, y_{N-2}] = DCT(M, \bar{\boldsymbol{x}})$, $[y_1, y_3, \ldots, y_{N-1}] = DCT(M, \tilde{\boldsymbol{x}})$ możemy obliczyć rekurencyjnie rozwiązując dwa podproblemy DCT o rozmiarze $M = N/2$.

$$y_k = \sum_{j=0}^{M-1} \left(x_j + (-1)^k x_{M+j} \right) \theta_N^{jk}, \qquad k = 0, 1, \ldots, N-1$$

Dla parzystych i nieparzystych wskaźników otrzymujemy wzory

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

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

Stąd widzimy, że wektory $[y_0, y_2, \ldots, y_{N-2}] = DCT(M, \bar{\boldsymbol{x}})$, $[y_1, y_3, \ldots, y_{N-1}] = DCT(M, \tilde{\boldsymbol{x}})$ możemy obliczyć rekurencyjnie rozwiązując dwa podproblemy DCT o rozmiarze $M = N/2$.
Dla uzasadnienia złożoności obliczeniowej algorytmu FFT, wystarczy skorzystać z tego, że rozwiązaniem związku rekurencyjnego

$$T(N) = 2T(N/2) + \mathcal{O}(N)$$

jest $T(N) = \mathcal{O}(N \log N)$.

Szybka transformata Fouriera
Implementacja w Juli

```
# Dyskretna transformacja cosinusowa
# Implementacja naiwna, wprost ze wzroru.
# Złożoność: O(n^2)
function slowFFT(x)
  N = length(x);
  θ = [exp(Complex(0,2π*j/N)) for j=0:N-1];
  y = Complex(0,0)*zeros(N);
  for k=0:N-1
    y[k+1] = dot(x,θ.^k);
  end;
  return y;
end;

# Dyskretna transformacja cosinusowa
# Implementacja za pomocą "dziel i zwyciężaj"
# Złożoność: O(n log n)
function myFFT(x) # N = 2^k
  N = length(x);
  if (N==1) return x; end;
  M = Int( floor(N/2) );
  xL = x[1:M];
  xR = x[M+1:N];
  ye = myFFT(xL+xR);
  yo = myFFT((xL-xR).*[exp(Complex(0,2π*j/N)) for j=0:M-1]);
  y = Complex(0,0)*zeros(N);
  y[1:2:N-1] = ye;
  y[2:2:N] = yo;
  return y;
end;
```