Analiza numeryczna

Kwadratury wysokiego rzędu

Rafał Nowak

19 grudnia 2017 r.

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

Lemat

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

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*

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

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

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

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

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

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

Kwadratura Clenshawa-Curtisa

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

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

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

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

Interpolacyjna kwadratura Clenshawa-Curtisa

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

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

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

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

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

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 $\mathcal{O}(n^2)$.

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.

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

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

Zakładamy, że $N = 2M$. Mamy

$$\begin{aligned} y_k &= \sum_{j=0}^{N-1} x_j \theta_N^{jk} = \sum_{j=0}^{M-1} x_j \theta_N^{jk} + \sum_{j=M}^{N-1} x_j \theta_N^{jk} \\ &= \sum_{j=0}^{M-1} x_j \theta_N^{jk} + \sum_{j=0}^{M-1} x_{M+j} \theta_N^{(M+j)k} = \sum_{j=0}^{M-1} x_j \theta_N^{jk} + \sum_{j=0}^{M-1} (-1)^k x_{M+j} \theta_N^{jk} \\ &= \sum_{j=0}^{M-1} \left(x_j + (-1)^k x_{M+j} \right) \theta_N^{jk} \end{aligned} \tag{3}$$

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

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

Dla parzystych i nieparzystych wskaźników otrzymujemy wzory

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

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

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

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

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

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

```
# 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;
```