Wybrane problemy analizy numerycznej i przykładowe sposoby ich rozwiązywania

Paweł Keller

Iwona Wróbel

Alicja Smoktunowicz

Wprowadzenie

Analiza numeryczna

Wprowadzenie

Analiza numeryczna

Metody numeryczne

Wprowadzenie

Analiza numeryczna

Metody numeryczne

Matematyka obliczeniowa

Wprowadzenie

Analiza numeryczna

Metody numeryczne

Matematyka obliczeniowa

Obliczenia techniczne

Wprowadzenie

Wprowadzenie

Wprowadzenie

Wprowadzenie

Wprowadzenie

Część 1

Obliczanie *trudnych* całek

Całkowanie numeryczne

$$\int_a^b g(x)\,dx$$

Całkowanie numeryczne

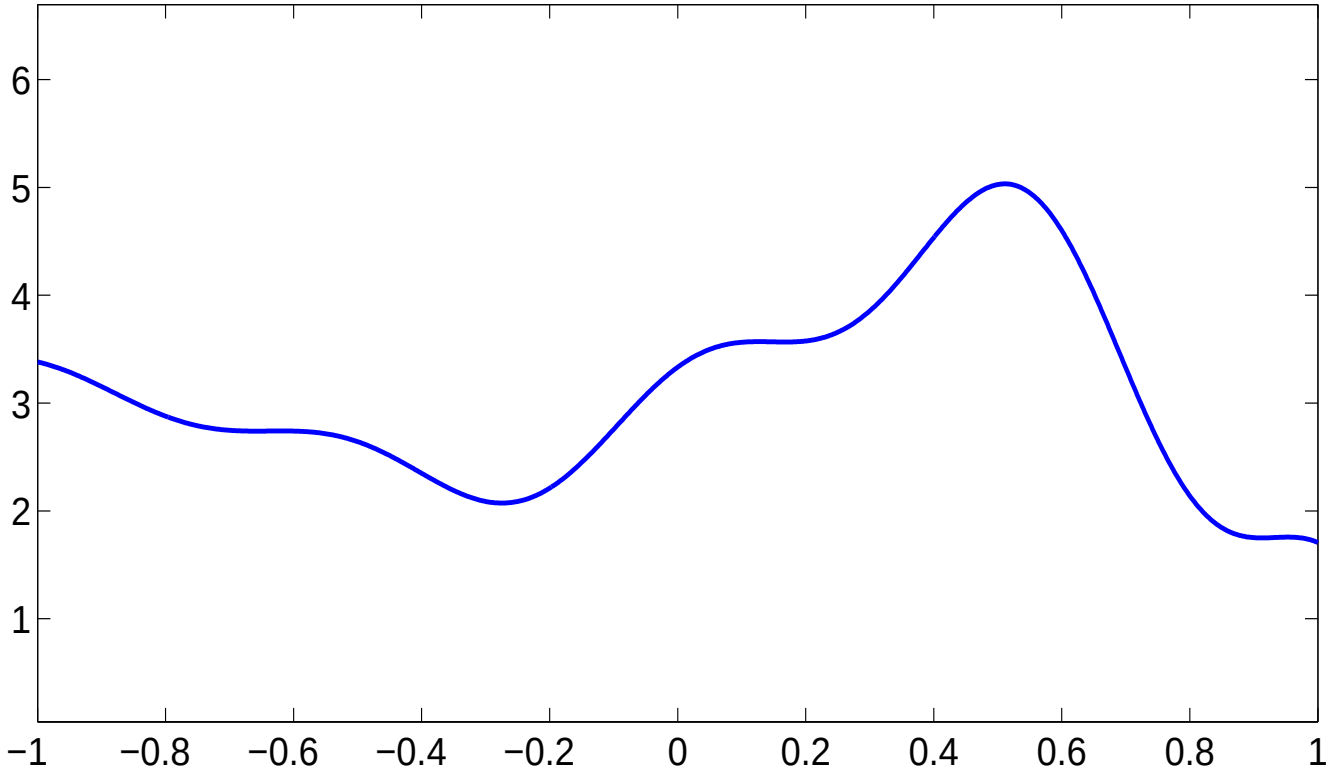
$$\int_a^b g(x)\,dx \simeq \int_a^b L_n(x)\,dx$$

Całkowanie numeryczne

$$\int_a^b g(x)\,dx \simeq \int_a^b L_n(x)\,dx = \sum_{k=0}^{n} A_k g(x_k)$$

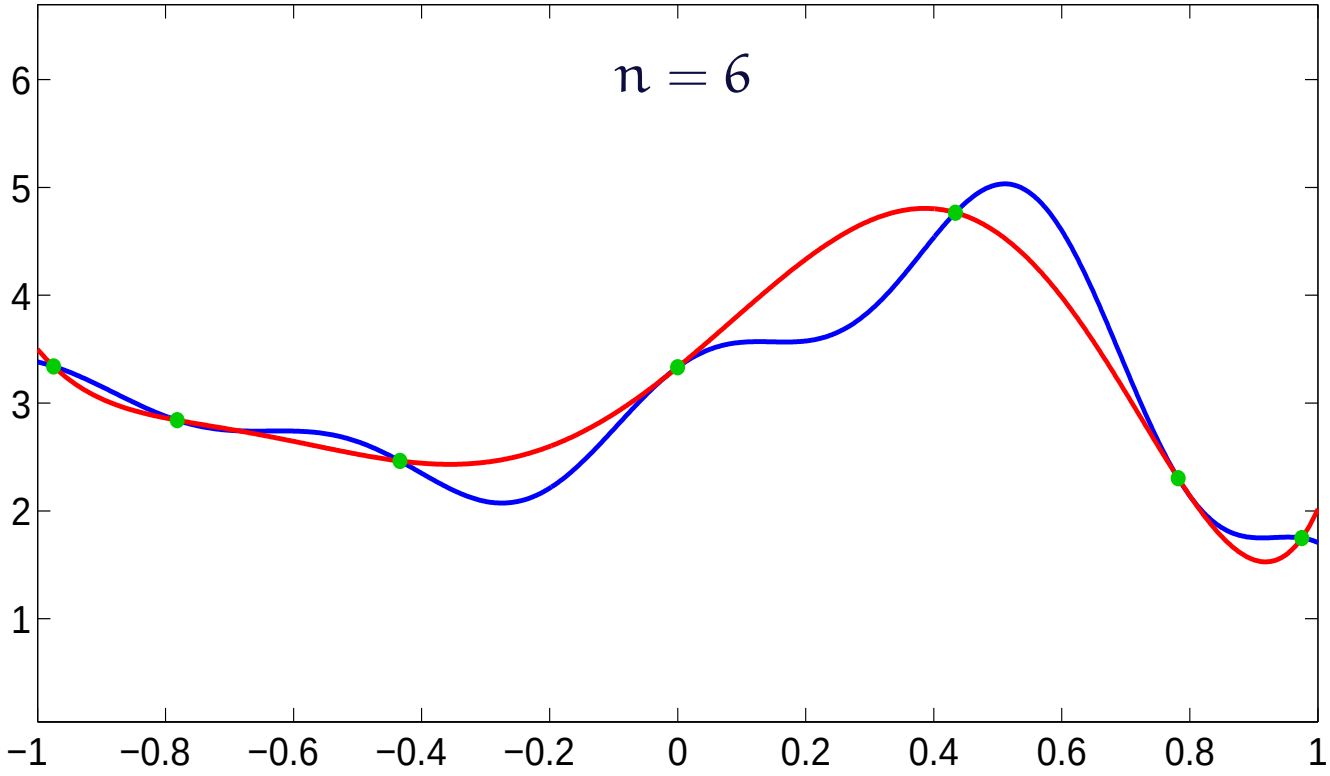
Całkowanie numeryczne

$$\int_a^b g(x)\,dx \simeq \int_a^b L_n(x)\,dx = \sum_{k=0}^{n} A_k g(x_k)$$



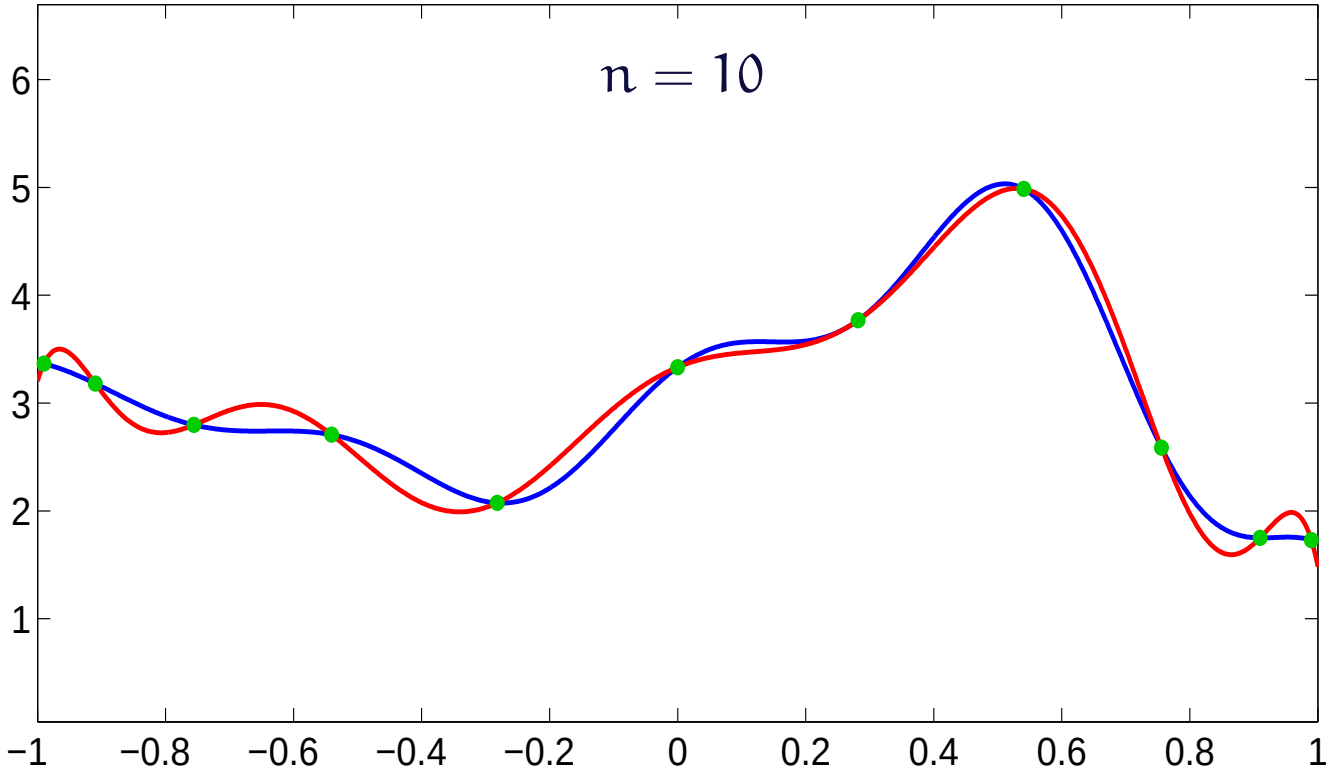
Całkowanie numeryczne

$$\int_a^b g(x)\,dx \simeq \int_a^b L_n(x)\,dx = \sum_{k=0}^{n} A_k g(x_k)$$



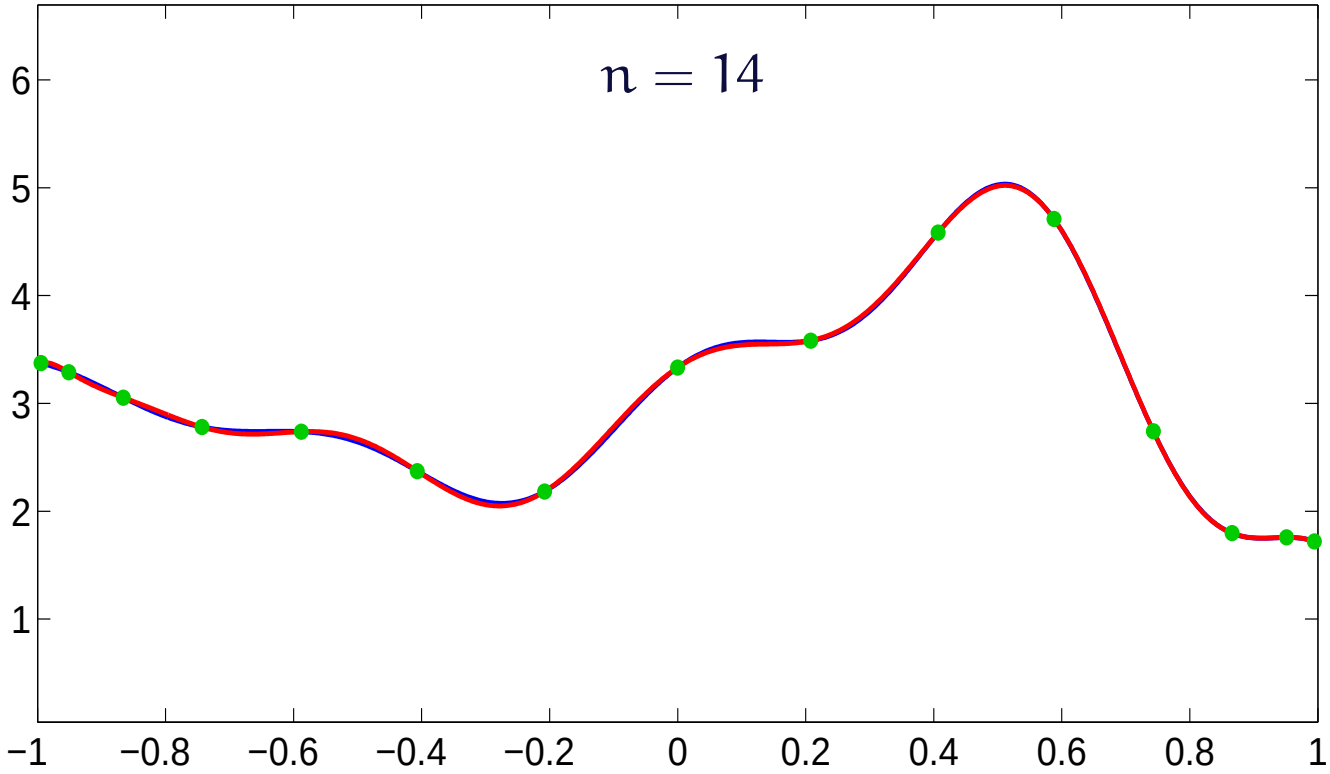
Całkowanie numeryczne

$$\int_a^b g(x)\,dx \simeq \int_a^b L_n(x)\,dx = \sum_{k=0}^{n} A_k g(x_k)$$



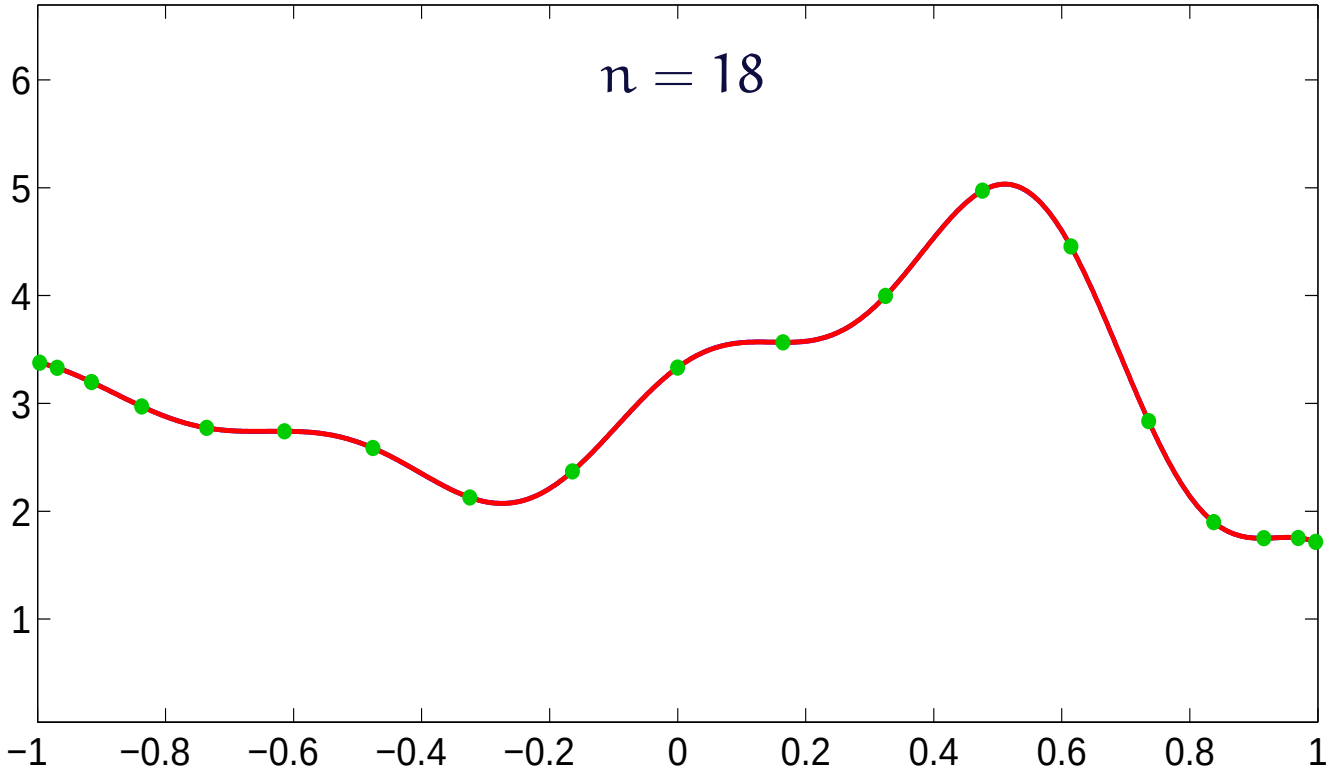
Całkowanie numeryczne

$$\int_a^b g(x)\,dx \simeq \int_a^b L_n(x)\,dx = \sum_{k=0}^{n} A_k g(x_k)$$



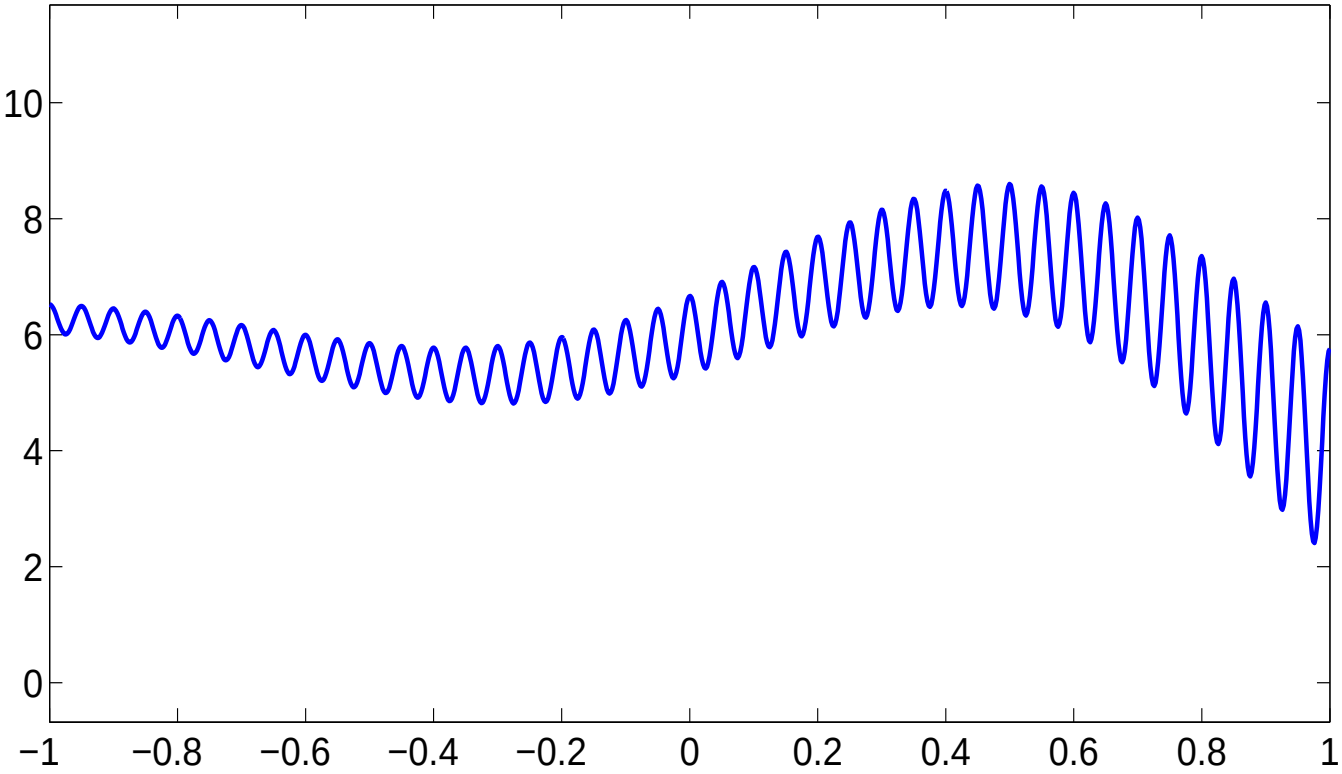
Całkowanie numeryczne

$$\int_a^b g(x)\,dx \simeq \int_a^b L_n(x)\,dx = \sum_{k=0}^{n} A_k g(x_k)$$



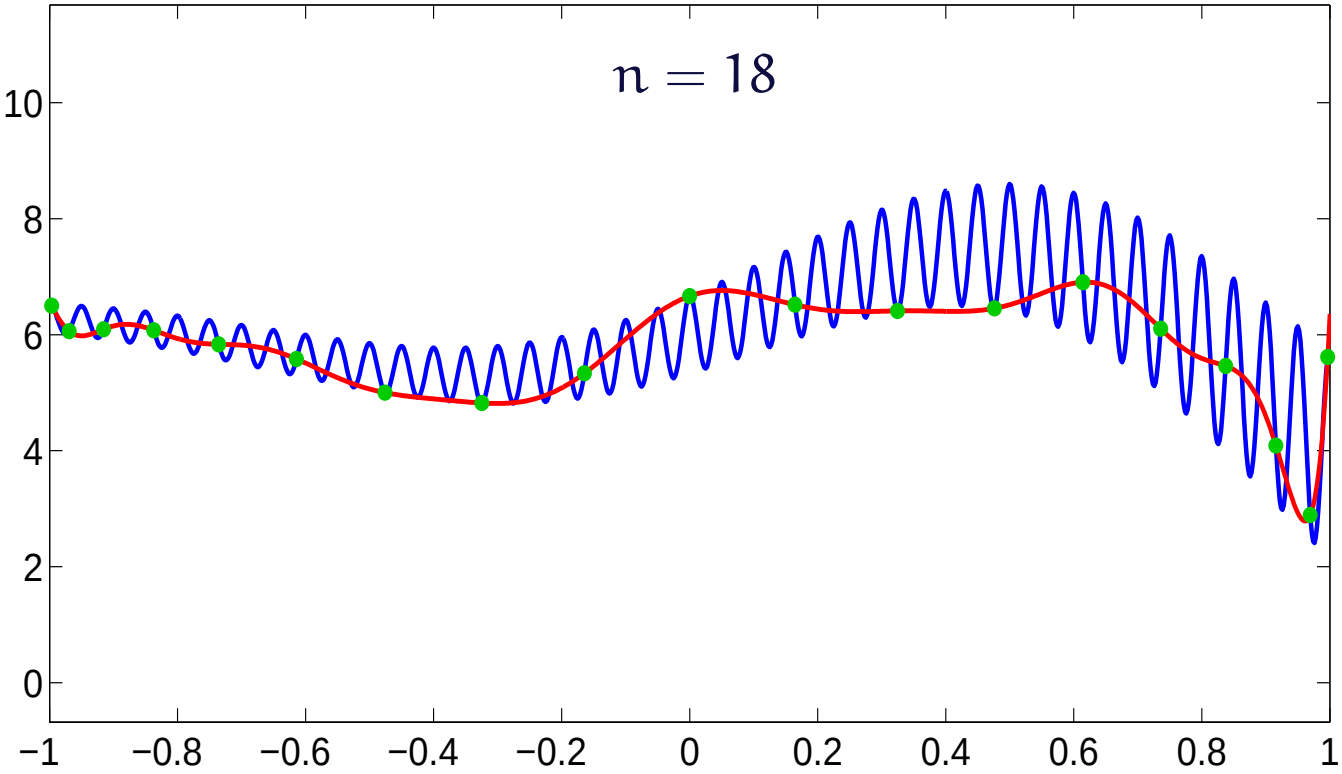
Całkowanie numeryczne

$$\int_a^b g(x)\,dx \simeq \int_a^b L_n(x)\,dx = \sum_{k=0}^{n} A_k g(x_k)$$



Całkowanie numeryczne

$$\int_a^b g(x)\,dx \simeq \int_a^b L_n(x)\,dx = \sum_{k=0}^{n} A_k g(x_k)$$



Transf. Hilberta funkcji oscylujących

Transf. Hilberta funkcji oscylujących

Problem: Wyznaczyć wartość transformaty Hilberta (na przedziale skończonym) funkcji silnie oscylującej:

$$\left.\begin{aligned} S(f,\omega,\tau) &:= \fint_{-1}^{1} \frac{f(x)\sin(\omega x)}{x-\tau}\,dx, \\ C(f,\omega,\tau) &:= \fint_{-1}^{1} \frac{f(x)\cos(\omega x)}{x-\tau}\,dx \end{aligned}\right\} \quad (f - \text{ciągła},\ -1<\tau<1)\,.$$

Transf. Hilberta funkcji oscylujących

Problem: Wyznaczyć wartość transformaty Hilberta (na przedziale skończonym) funkcji silnie oscylującej:

$$\left.\begin{aligned} S(f,\omega,\tau) &:= \fint_{-1}^{1} \frac{f(x)\sin(\omega x)}{x-\tau}\,dx, \\ C(f,\omega,\tau) &:= \fint_{-1}^{1} \frac{f(x)\cos(\omega x)}{x-\tau}\,dx \end{aligned}\right\} \quad (f - \text{ciągła},\ -1<\tau<1)\,.$$

Zgodnie z definicją,

$$\fint_{-1}^{1} \frac{g(x)}{x-\tau}\,dx = \lim_{\varepsilon\to 0^+}\left(\int_{-1}^{\tau-\varepsilon} \frac{g(x)}{x-\tau}\,dx + \int_{\tau+\varepsilon}^{1} \frac{g(x)}{x-\tau}\,dx\right).$$

Transf. Hilberta funkcji oscylujących

Problem: Wyznaczyć wartość transformaty Hilberta (na przedziale skończonym) funkcji silnie oscylującej:

$$\left.\begin{aligned} S(f,\omega,\tau) &:= \fint_{-1}^{1} \frac{f(x)\sin(\omega x)}{x-\tau}\,dx, \\ C(f,\omega,\tau) &:= \fint_{-1}^{1} \frac{f(x)\cos(\omega x)}{x-\tau}\,dx \end{aligned}\right\} \quad (f - \text{ciągła},\ -1<\tau<1)\,.$$

Zgodnie z definicją,

$$\fint_{-1}^{1} \frac{g(x)}{x-\tau}\,dx = \lim_{\varepsilon\to 0^+}\left(\int_{-1}^{\tau-\varepsilon} \frac{g(x)}{x-\tau}\,dx + \int_{\tau+\varepsilon}^{1} \frac{g(x)}{x-\tau}\,dx\right).$$

Powyższa całka istnieje, gdy g spełnia warunek Höldera:

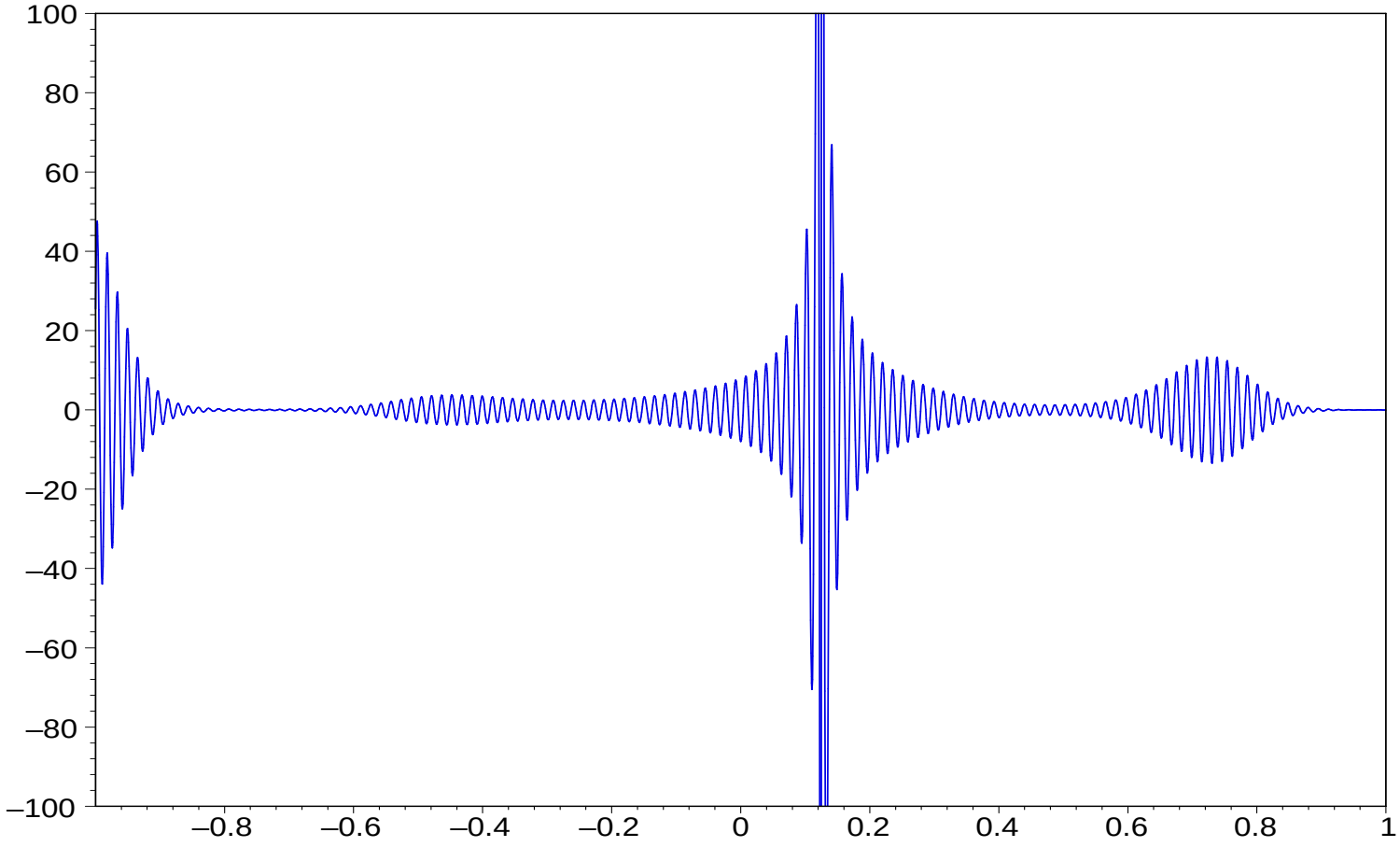
$\exists_{p,q}\forall_{x,y}\ |g(x)-g(y)| < p|x-y|^q$.

Transf. Hilberta funkcji oscylujących

Wykres funkcji $\dfrac{e^{4x^2}\sin(11x)\cos(400x)}{x-\frac{1}{8}}$:

Transf. Hilberta funkcji oscylujących

Wykres funkcji $\dfrac{e^{4x^2}\sin(11x)\cos(400x)}{x-\frac{1}{8}}$:



Transf. Hilberta funkcji oscylujących

Przykładowa metoda obliczania powyższej transformaty Hilberta:

Transf. Hilberta funkcji oscylujących

Przykładowa metoda obliczania powyższej transformaty Hilberta:

Ze znanej równości

$$e^{ix} = \cos(x) + i\sin(x) \qquad (i = \sqrt{-1})$$

Transf. Hilberta funkcji oscylujących

Przykładowa metoda obliczania powyższej transformaty Hilberta:

Ze znanej równości

$$e^{ix} = \cos(x) + i\sin(x) \qquad (i = \sqrt{-1})$$

widać, że

$$S(f,\omega,\tau) = \Im H(f,\omega,\tau)\,, \qquad C(f,\omega,\tau) = \Re H(f,\omega,\tau)\,,$$

gdzie

$$H(f,\omega,\tau) := \fint_{-1}^{1} \frac{f(x)\,e^{i\omega x}}{x-\tau}\,dx\,,$$

Transf. Hilberta funkcji oscylujących

$$H(f,\omega,\tau) := \int_{-1}^{1} \frac{f(x)\, e^{i\omega x}}{x-\tau}\, dx = \ ?$$

Transf. Hilberta funkcji oscylujących

$$H(f,\omega,\tau) := \int_{-1}^{1} \frac{f(x)\, e^{i\omega x}}{x-\tau}\, dx = \ ?$$

Niech $K(x) = e^{i\omega x}(x-\tau)^{-1}$.

Transf. Hilberta funkcji oscylujących

$$H(f,\omega,\tau) := \int_{-1}^{1} \frac{f(x)\, e^{i\omega x}}{x-\tau}\, dx = \ ?$$

Niech $K(x) = e^{i\omega x}(x-\tau)^{-1}$. Łatwo sprawdzić, że

$$(x-\tau)K'(x) + [1 - i\omega(x-\tau)]\, K(x) = 0\,.$$

Transf. Hilberta funkcji oscylujących

$$H(f,\omega,\tau) := \int_{-1}^{1} \frac{f(x)\, e^{i\omega x}}{x-\tau}\, dx = \ ?$$

Niech $K(x) = e^{i\omega x}(x-\tau)^{-1}$. Łatwo sprawdzić, że

$$(x-\tau)K'(x) + [1 - i\omega(x-\tau)]\, K(x) = 0\,.$$

Stąd ... otrzymujemy:

$$\left[e^{i\omega x} F(x)\right]' = f(x)\, e^{i\omega x}\, (x-\tau)^{-1},$$

dla *dowolnego* rozwiązania $F(x)$ równania różniczkowego

$$(x-\tau)\, F'(x) + i\omega(x-\tau)\, F(x) = f(x)\,.$$

Transf. Hilberta funkcji oscylujących

$$f = \sum_{k=0}^{\infty}{}' a_k T_k\,, \qquad F = \sum_{k=0}^{\infty}{}' d_k T_k$$

$(T_0(x) \equiv 1,\ T_1(x) = x,\ T_k(x) = 2xT_{k-1}(x) - T_{k-2}(x)$ dla $k > 1)$

Transf. Hilberta funkcji oscylujących

Jeżeli

$$f = \sum_{k=0}^{\infty}{}' a_k T_k\,, \qquad F = \sum_{k=0}^{\infty}{}' d_k T_k$$

$(T_0(x) \equiv 1,\ T_1(x) = x,\ T_k(x) = 2xT_{k-1}(x) - T_{k-2}(x)$ dla $k > 1)$ oraz

$$(x - \tau)\, F'(x) \ + \ i\omega(x - \tau)\, F(x) \ = \ f(x)$$

Transf. Hilberta funkcji oscylujących

Jeżeli

$$f = \sum_{k=0}^{\infty}{}' a_k T_k\,, \qquad F = \sum_{k=0}^{\infty}{}' d_k T_k$$

$(T_0(x) \equiv 1,\ T_1(x) = x,\ T_k(x) = 2xT_{k-1}(x) - T_{k-2}(x)$ dla $k > 1)$ oraz

$$(x-\tau)\,F'(x) \;+\; i\omega(x-\tau)\,F(x) \;=\; f(x)$$

to

$$\begin{aligned} i\omega\,d_{k-2} \;&+\; 2[k-1-i\omega\tau]\,d_{k-1} \;-\; 4k\tau\,d_k \;+ \\ &+\; 2[k+1+i\omega\tau]\,d_{k+1} \;-\; i\omega\,d_{k+2} \;=\; 2(a_{k-1}-a_{k+1}) \qquad (k>0) \end{aligned}$$

$(a_{-k} = a_k,\ d_{-k} = d_k)$.

Transf. Hilberta funkcji oscylujących

Jeżeli

$$f = \sum_{k=0}^{\infty}{}' a_k T_k\,, \qquad F = \sum_{k=0}^{\infty}{}' d_k T_k$$

$(T_0(x) \equiv 1,\ T_1(x) = x,\ T_k(x) = 2xT_{k-1}(x) - T_{k-2}(x)$ dla $k > 1)$ oraz

$$(x-\tau)\,F'(x) \;+\; i\omega(x-\tau)\,F(x) \;=\; f(x)$$

to

$$\begin{aligned} i\omega\,d_{k-2} \;&+\; 2[k-1-i\omega\tau]\,d_{k-1} \;-\; 4k\tau\,d_k \;+ \\ &+\; 2[k+1+i\omega\tau]\,d_{k+1} \;-\; i\omega\,d_{k+2} \;=\; 2(a_{k-1}-a_{k+1}) \qquad (k>0) \end{aligned}$$

$(a_{-k} = a_k,\ d_{-k} = d_k)$.

Niestety, powyższe równanie rekurencyjne nie ma rozwiązania.

Transf. Hilberta funkcji oscylujących

Równanie rekurencyjne

$$i\omega d_{k-2} + 2[k-1-i\omega\tau]d_{k-1} - 4k\tau d_k + \\ + 2[k+1+i\omega\tau]d_{k+1} - i\omega d_{k+2} = 2(a_{k-1} - a_{k+1}) \quad (k>0)$$

nie ma rozwiązania

Transf. Hilberta funkcji oscylujących

Równanie rekurencyjne

$$i\omega d_{k-2} + 2[k-1-i\omega\tau]d_{k-1} - 4k\tau d_k + \\ + 2[k+1+i\omega\tau]d_{k+1} - i\omega d_{k+2} = 2(a_{k-1}-a_{k+1}) \quad (k>0)$$

nie ma rozwiązania, ale ma je równanie

$$i\omega d_{k-2} + 2[k-1-i\omega\tau]d_{k-1} - 4k\tau d_k + \\ + 2[k+1+i\omega\tau]d_{k+1} - i\omega d_{k+2} = 2(a_{k-1}-a_{k+1}) \quad (k>1)\,.$$

Transf. Hilberta funkcji oscylujących

Równanie rekurencyjne

$$i\omega d_{k-2} + 2[k-1-i\omega\tau]d_{k-1} - 4k\tau d_k + \\ + 2[k+1+i\omega\tau]d_{k+1} - i\omega d_{k+2} = 2(a_{k-1}-a_{k+1}) \quad (k>0)$$

nie ma rozwiązania, ale ma je równanie

$$i\omega d_{k-2} + 2[k-1-i\omega\tau]d_{k-1} - 4k\tau d_k + \\ + 2[k+1+i\omega\tau]d_{k+1} - i\omega d_{k+2} = 2(a_{k-1}-a_{k+1}) \quad (k>1)\,.$$

W takiej sytuacji funkcja $F = \sum_{k=0}^{\infty}{}' d_k T_k$ spełnia równanie różniczkowe

$$(x-\tau)F'(x) + [i\omega(x-\tau)]F(x) = f(x) + \frac{R}{4}\,,$$

gdzie $R = 2i\omega\tau(d_2-d_0) + i\omega(d_1-d_3) + 4(d_2-\tau d_1) - 2(a_0-a_2)\,.$

Transf. Hilberta funkcji oscylujących

$$\int_{-1}^{1} \frac{\left(f(x) + \frac{R}{4}\right) e^{i\omega x}}{x - \tau}\, dx$$

Transf. Hilberta funkcji oscylujących

Oczywiata jest równość

$$\int_{-1}^{1} \frac{f(x)\, e^{i\omega x}}{x-\tau}\, dx \;=\; \int_{-1}^{1} \frac{\left(f(x)+\frac{R}{4}\right) e^{i\omega x}}{x-\tau}\, dx \;-\; \frac{R}{4}\int_{-1}^{1} \frac{e^{i\omega x}}{x-\tau}\, dx\,.$$

Transf. Hilberta funkcji oscylujących

Oczywiata jest równość

$$\int_{-1}^{1} \frac{f(x)\, e^{i\omega x}}{x-\tau}\, dx \;=\; \int_{-1}^{1} \frac{\left(f(x)+\frac{R}{4}\right) e^{i\omega x}}{x-\tau}\, dx \;-\; \frac{R}{4}\int_{-1}^{1} \frac{e^{i\omega x}}{x-\tau}\, dx\,.$$

Dodatkowo, można udowodnić, że

$$\int_{-1}^{1} \frac{e^{i\omega x}}{x-\tau}\, dx \;=\; e^{i\omega\tau}\left[\pi i + E_1(i\omega(1+\tau)) - E_1(i\omega(-1+\tau))\right],$$

gdzie

$$E_1(z) = \int_1^{\infty} t^{-1} e^{-tz}\, dt$$

to tak zwana całka wykładnicza (znane są szybkie i proste algorytmy jej przybliżania, za pomocą szeregu albo ułamka łańcuchowego).

Transf. Hilberta funkcji oscylujących

Podsumowując:

Transf. Hilberta funkcji oscylujących

Podsumowując: jeśli $f = \sum_{k=0}^{\infty}{}' a_k T_k$.

Transf. Hilberta funkcji oscylujących

Podsumowując: jeśli $f = {\sum_{k=0}^{\infty}}' a_k T_k$. Wtedy

$$\fint_{-1}^{1} \frac{f(x)\, e^{i\omega x}}{x-\tau}\, dx \;=\; e^{i\omega} {\sum_{k=0}^{\infty}}' d_k \;-\; e^{-i\omega} {\sum_{k=0}^{\infty}}' (-1)^k d_k \;-$$

$$-\; \frac{R}{4}\, e^{i\omega\tau} \left[\pi i + E_1(i\omega(1+\tau)) - E_1(i\omega(-1+\tau)) \right],$$

$$i\omega\, d_{k-2} \;+\; 2[k-1-i\omega\tau]\, d_{k-1} \;-\; 4k\tau\, d_k \;+$$

$$+\; 2[k+1+i\omega\tau]\, d_{k+1} \;-\; i\omega\, d_{k+2} \;=\; 2(a_{k-1} - a_{k+1}) \qquad (k>1),$$

$$R \;=\; 2i\omega\tau(d_2 - d_0) \;+\; i\omega(d_1 - d_3) \;+\; 4(d_2 - \tau d_1) - 2(a_0 - a_2),$$

$$E_1(z) = \int_1^{\infty} t^{-1} e^{-tz}\, dt\,.$$

Transf. Hilberta funkcji oscylujących

W praktyce: $f \approx \sum_{k=0}^{N}{}' a_k T_k$.

Transf. Hilberta funkcji oscylujących

W praktyce: $f \approx \sum_{k=0}^{N}{}' a_k T_k$. Wtedy

$$\int_{-1}^{1} \frac{f(x)\, e^{i\omega x}}{x-\tau}\, dx = e^{i\omega} \sum_{k=0}^{M}{}' d_k - e^{-i\omega} \sum_{k=0}^{M}{}' (-1)^k d_k -$$

$$- \frac{R}{4}\, e^{i\omega\tau} \left[\pi i + E_1(i\omega(1+\tau)) - E_1(i\omega(-1+\tau)) \right].$$

Transf. Hilberta funkcji oscylujących

W praktyce: $f \approx \sum_{k=0}^{N}{}' a_k T_k$. Wtedy

$$\int_{-1}^{1} \frac{f(x)\, e^{i\omega x}}{x-\tau}\, dx = e^{i\omega} \sum_{k=0}^{M}{}' d_k - e^{-i\omega} \sum_{k=0}^{M}{}' (-1)^k d_k - \\ - \frac{R}{4}\, e^{i\omega\tau} \left[\pi i + E_1(i\omega(1+\tau)) - E_1(i\omega(-1+\tau)) \right].$$

$M = O(N)$.

Transf. Hilberta funkcji oscylujących

W praktyce: $f \approx \sum_{k=0}^{N}{}' a_k T_k$. Wtedy

$$\int_{-1}^{1} \frac{f(x)\, e^{i\omega x}}{x-\tau}\, dx \;=\; e^{i\omega} \sum_{k=0}^{M}{}' d_k \;-\; e^{-i\omega} \sum_{k=0}^{M}{}' (-1)^k d_k \;-$$

$$-\; \frac{R}{4}\, e^{i\omega\tau} \left[\pi i + E_1(i\omega(1+\tau)) - E_1(i\omega(-1+\tau)) \right].$$

$M = O(N)$.

$N \Longleftarrow$ żądana dokładność przybliżenia całki $H(f, \omega, \tau)$.

Transf. Hilberta funkcji oscylujących

Przedstawiona metoda, przy $N \to \infty$, jest zbieżna jednostajnie ze względu na ω i τ do wartości dokładnej.

Transf. Hilberta funkcji oscylujących

Przedstawiona metoda, przy $N \to \infty$, jest zbieżna jednostajnie ze względu na ω i τ do wartości dokładnej.

Błąd $\sim N^{-s} \log N$,
gdy f ma ciągłą jedynie $(s-1)$-szą pochodną.

Transf. Hilberta funkcji oscylujących

Przedstawiona metoda, przy $N \to \infty$, jest zbieżna jednostajnie ze względu na ω i τ do wartości dokładnej.

Błąd $\sim N^{-s} \log N$,
gdy f ma ciągłą jedynie $(s-1)$-szą pochodną.

Błąd $\sim r^{-N} \log N$ $(r > 1)$,
gdy f jest analityczna w elipsie zawierającej $[-1, 1]$.

Testy

Testy

Wyniki dla całki $\int_{-1}^{1} \frac{e^{4x^2 \sin(11x)} \cos(400x)}{x - \frac{1}{8}} = 0.98097539735692699 33$

Testy

Wyniki dla całki $\int_{-1}^{1} \frac{e^{4x^2} \sin(11x) \cos(400x)}{x - \frac{1}{8}} = 0.98097539735692699 33$

zadana tolerancja	wartość N	błąd bezwzględny
$1.0 \cdot 10^{-02}$	47	$1.3 \cdot 10^{-03}$
$1.0 \cdot 10^{-04}$	64	$4.0 \cdot 10^{-05}$
$1.0 \cdot 10^{-06}$	84	$2.0 \cdot 10^{-07}$
$1.0 \cdot 10^{-08}$	100	$2.4 \cdot 10^{-09}$
$1.0 \cdot 10^{-10}$	113	$1.2 \cdot 10^{-11}$
$1.0 \cdot 10^{-12}$	128	$3.1 \cdot 10^{-13}$
$1.0 \cdot 10^{-14}$	146	$6.7 \cdot 10^{-17}$

Testy

Wyniki dla całki $\int_{-1}^{1} \frac{\sqrt{(1-x^2)^3}\cos(400x)}{x-\frac{1}{8}} = 0.8050319228314049224$

Testy

Wyniki dla całki $\int_{-1}^{1} \frac{\sqrt{(1-x^2)^3}\cos(400x)}{x-\frac{1}{8}} = 0.8050319228314049224$

zadana tolerancja	wartość N	błąd bezwzględny
$1.0 \cdot 10^{-02}$	8	$8.4 \cdot 10^{-04}$
$1.0 \cdot 10^{-04}$	30	$1.6 \cdot 10^{-05}$
$1.0 \cdot 10^{-06}$	90	$8.7 \cdot 10^{-08}$
$1.0 \cdot 10^{-08}$	280	$1.6 \cdot 10^{-09}$
$1.0 \cdot 10^{-10}$	1022	$1.1 \cdot 10^{-12}$
$1.0 \cdot 10^{-12}$	2856	$2.2 \cdot 10^{-14}$
$1.0 \cdot 10^{-14}$	8890	$2.4 \cdot 10^{-16}$

Testy

Wyniki dla całki $\int_{-1}^{1} \frac{\sqrt{(1-x^2)^3}\cos(400x)}{x-\frac{1}{8}} = 0.8050319228314049224$

zadana tolerancja	wartość N	błąd bezwzględny
$1.0 \cdot 10^{-02}$	8	$8.4 \cdot 10^{-04}$
$1.0 \cdot 10^{-04}$	30	$1.6 \cdot 10^{-05}$
$1.0 \cdot 10^{-06}$	90	$8.7 \cdot 10^{-08}$
$1.0 \cdot 10^{-08}$	280	$1.6 \cdot 10^{-09}$
$1.0 \cdot 10^{-10}$	1022	$1.1 \cdot 10^{-12}$
$1.0 \cdot 10^{-12}$	2856	$2.2 \cdot 10^{-14}$
$1.0 \cdot 10^{-14}$	8890	$2.4 \cdot 10^{-16}$

Czasy ($N = 8890$). W&X(2009): 0.3s., W&X(2010): N/A, K: 0.012s.

Część 2

Własności numeryczne algorytmów wyznaczania uogólnionej odwrotności Moore’a-Penrose’a

Odwracanie macierzy

Macierz odwrotna

Macierz odwrotna

Niech $A \in \mathbb{C}^{n \times n}$. Macierz $X \in \mathbb{C}^{n \times n}$ jest macierzą odwrotną do macierzy A wtedy i tylko wtedy, gdy

$$AX = XA = I.$$

Oznaczenie: A^{-1}.

Macierz odwrotna

Niech $A \in \mathbb{C}^{n \times n}$. Macierz $X \in \mathbb{C}^{n \times n}$ jest macierzą odwrotną do macierzy A wtedy i tylko wtedy, gdy

$$AX = XA = I.$$

Oznaczenie: A^{-1}.

”To most numerical analysts, matrix inversion is a sin.”

Nicholas Higham

”Accuracy and Stability of Numerical Algorithms”, 1996.

Macierz odwrotna

Niech $A \in \mathbb{C}^{n \times n}$. Macierz $X \in \mathbb{C}^{n \times n}$ jest macierzą odwrotną do macierzy A wtedy i tylko wtedy, gdy

$$AX = XA = I.$$

Oznaczenie: A^{-1}.

"To most numerical analysts, matrix inversion is a sin."

Nicholas Higham

"Accuracy and Stability of Numerical Algorithms", 1996.

"Almost anything you can do with A^{-1} can be done without it."

George E. Forsythe and Cleve B. Moler

"Computer Solution of Linear Algebraic Systems", 1967.

Kiedy nie używać A^{-1}?

- Aby rozwiązać równanie $Ax = b$, teoretycznie można wyznaczyć A^{-1}, a potem obliczyć $x = A^{-1}b$.

Kiedy nie używać A^{-1}?

- Aby rozwiązać równanie $Ax = b$, teoretycznie można wyznaczyć A^{-1}, a potem obliczyć $x = A^{-1}b$.

  Jednak nie powinno się tak robić, ze względu na koszt obliczeń oraz dokładność wyniku.

Kiedy nie używać A^{-1}?

- Aby rozwiązać równanie $Ax = b$, teoretycznie można wyznaczyć A^{-1}, a potem obliczyć $x = A^{-1}b$.

  Jednak nie powinno się tak robić, ze względu na koszt obliczeń oraz dokładność wyniku.

Test: 100000 macierzy o elementach losowych (z rozkładu $U(0,1)$). Tabela zawiera wartości błędu $\|b - A\tilde{x}\|_\infty$, gdzie $\tilde{x}$ jest rozwiązaniem obliczonym.

Kiedy nie używać A^{-1}?

- Aby rozwiązać równanie $Ax = b$, teoretycznie można wyznaczyć A^{-1}, a potem obliczyć $x = A^{-1}b$.

  Jednak nie powinno się tak robić, ze względu na koszt obliczeń oraz dokładność wyniku.

Test: 100000 macierzy o elementach losowych (z rozkładu $U(0,1)$). Tabela zawiera wartości błędu $\|b - A\tilde{x}\|_\infty$, gdzie $\tilde{x}$ jest rozwiązaniem obliczonym.

	min	max
$x = A^{-1}b$	$2.8422e{-}14$	$1.0397e{-}8$
GEPP	$7.1054e{-}15$	$4.2633e{-}14$

Liniowe zadanie najmniejszych kwadratów

- To samo odnosi się do liniowego zadania najmniejszych kwadratów

$$\min_x \|b - Ax\|_2, \quad A \in \mathbb{R}^{m \times n}, \quad m \geq n = rank(A).$$

Liniowe zadanie najmniejszych kwadratów

- To samo odnosi się do liniowego zadania najmniejszych kwadratów

$$\min_x \|b - Ax\|_2, \quad A \in \mathbb{R}^{m\times n}, \quad m \geq n = rank(A).$$

Rozwiązanie normalne to $x = (A^T A)^{-1} A^T b$, jednak nie powinno się korzystać z tego wzoru ze względu na szybkość i dokładność.

Liniowe zadanie najmniejszych kwadratów

- To samo odnosi się do liniowego zadania najmniejszych kwadratów

$$\min_x \|b - Ax\|_2, \quad A \in \mathbb{R}^{m \times n}, \quad m \geq n = rank(A).$$

  Rozwiązanie normalne to $x = (A^T A)^{-1} A^T b$, jednak nie powinno się korzystać z tego wzoru ze względu na szybkość i dokładność.

W pewnych sytuacjach odwrotność jednak trzeba wyznaczyć.

Odwrotność Moore'a-Penrose'a

Definicja. (Warunki Moore'a-Penrose'a)
Macierz $X \in \mathbb{C}^{n \times m}$ jest uogólnioną odwrotnością macierzy $A \in \mathbb{C}^{m \times n}$ wtedy i tylko wtedy, gdy

(1) $AXA = A$,

(2) $XAX = X$,

(3) $AX = (AX)^*$,

(4) $XA = (XA)^*$.

Oznaczenie: $X = A^\dagger$ (odwrotność Moore'a-Penrose'a)

Odwrotność Moore'a-Penrose'a

Definicja. (Warunki Moore'a-Penrose'a)
Macierz $X \in \mathbb{C}^{n\times m}$ jest uogólnioną odwrotnością macierzy $A \in \mathbb{C}^{m\times n}$ wtedy i tylko wtedy, gdy

(1) $AXA = A$,

(2) $XAX = X$,

(3) $AX = (AX)^*$,

(4) $XA = (XA)^*$.

Oznaczenie: $X = A^\dagger$ (odwrotność Moore'a-Penrose'a)

- ♡ Każda macierz posiada jednoznaczną odwrotność Moore'a-Penrose'a.
- ♡ Jeśli $A \in \mathbb{C}^{n\times n}$ jest odwracalna, to $A^\dagger = A^{-1}$.
- ♡ Jeśli $A \in \mathbb{C}^{m\times n}$ oraz $m \geq n = \mathrm{rank}(A)$, to $A^\dagger = (A^*A)^{-1}A^*$.

Wpływ zaburzeń

Twierdzenie. Załóżmy, że $A \in \mathbb{R}^{m \times n}$, $m \geq n = \mathrm{rank}(A)$.

Niech ponadto $\|E\|_2 \leq \varepsilon \|A\|_2$, $\varepsilon\, \kappa(A) < 1$ oraz
$\mathrm{rank}(A + E) = \mathrm{rank}(A) = n$.

Wtedy

$$\|(A + E)^\dagger - A^\dagger\|_2 \leq \sqrt{2}\, \varepsilon\, \kappa(A)\, \|(A + E)^\dagger\|_2,$$

gdzie $\kappa(A) = \|A^\dagger\|_2 \|A\|_2$ jest wskaźnikiem uwarunkowania A.

Przykład

Niech

$$A = \begin{pmatrix} 1 & 0 \\ 0 & 0 \end{pmatrix}, \qquad E = \begin{pmatrix} 0 & 0 \\ 0 & \alpha \end{pmatrix}.$$

Przykład

Niech

$$A = \begin{pmatrix} 1 & 0 \\ 0 & 0 \end{pmatrix}, \qquad E = \begin{pmatrix} 0 & 0 \\ 0 & \alpha \end{pmatrix}.$$

Mamy $A^\dagger = A$, zatem A jest dobrze uwarunkowana ($\kappa(A) = \|A\|_2 \|A^\dagger\|_2 = 1$). Jednak mamy również $\mathrm{rank}(A + E) > \mathrm{rank}(A)$.

Przykład

Niech

$$A = \begin{pmatrix} 1 & 0 \\ 0 & 0 \end{pmatrix}, \qquad E = \begin{pmatrix} 0 & 0 \\ 0 & \alpha \end{pmatrix}.$$

Mamy $A^\dagger = A$, zatem A jest dobrze uwarunkowana ($\kappa(A) = \|A\|_2 \|A^\dagger\|_2 = 1$). Jednak mamy również $\mathrm{rank}(A + E) > \mathrm{rank}(A)$.

Dla $\varepsilon > 0$

$$(A + E)^\dagger - A^\dagger = \begin{pmatrix} 1 & 0 \\ 0 & \frac{1}{\alpha} \end{pmatrix} - \begin{pmatrix} 1 & 0 \\ 0 & 0 \end{pmatrix} = \begin{pmatrix} 0 & 0 \\ 0 & \frac{1}{\alpha} \end{pmatrix}.$$

Przykład

Niech

$$A = \begin{pmatrix} 1 & 0 \\ 0 & 0 \end{pmatrix}, \qquad E = \begin{pmatrix} 0 & 0 \\ 0 & \alpha \end{pmatrix}.$$

Mamy $A^\dagger = A$, zatem A jest dobrze uwarunkowana ($\kappa(A) = \|A\|_2 \|A^\dagger\|_2 = 1$). Jednak mamy również $\mathrm{rank}(A + E) > \mathrm{rank}(A)$.

Dla $\varepsilon > 0$

$$(A + E)^\dagger - A^\dagger = \begin{pmatrix} 1 & 0 \\ 0 & \frac{1}{\alpha} \end{pmatrix} - \begin{pmatrix} 1 & 0 \\ 0 & 0 \end{pmatrix} = \begin{pmatrix} 0 & 0 \\ 0 & \frac{1}{\alpha} \end{pmatrix}.$$

Zmniejszanie α powoduje wzrost błędu!

Stabilność

Algorytm wyznaczania $A^\dagger$, gdzie $A \in \mathbb{R}^{m\times n}$, jest stabilny, jeśli obliczona nim odwrotność $\tilde{X} \in \mathbb{R}^{n\times m}$ spełnia

$$\frac{\|\tilde{X} - A^\dagger\|_2}{\|A^\dagger\|_2} \leq L_1\, \varepsilon_M\, \kappa(A),$$

gdzie

$\kappa(A) = \|A^\dagger\|_2 \|A\|_2$ – wskaźnik uwarunkowania A,

$L_1 = L_1(m,n)$ – wolno rosnąca funkcja m i n,

ε_M – precyzja obliczeń.

Numeryczna poprawność

Algorytm obliczania $A^\dagger$, gdzie $A \in \mathbb{R}^{m\times n}$, jest numerycznie poprawny, jeśli obliczona nim odwrotność $\tilde{X} \in \mathbb{R}^{n\times m}$ jest prawie dokładną odwrotnością macierzy $A + E$, tj.:

$$\tilde{X} + F = (A + E)^\dagger,$$

gdzie

$$\|E\|_2 \leq L_2 \varepsilon_M \|A\|_2, \qquad \|F\|_2 \leq L_3 \varepsilon_M \|\tilde{X}\|_2,$$

a $L_2 = L_2(m, n)$ oraz $L_3 = L_3(m, n)$
są wolno rosnącymi funkcjami m i n.

Fakt. Algorytm numerycznie poprawny jest stabilny.

Algorytmy

Załóżmy, że $A \in \mathbb{R}^{m \times n}$ jest pełnego rzędu: $rank(A) = n \leq m$.

Jak obliczyć $X = A^{\dagger} = (A^T A)^{-1} A^T$?

Metody bezpośrednie: równania normalne

Algorytm 1 – Układ równań normalnych

- Stwórz $M = A^T A$.
- Rozwiąż równanie $MX = A^T$ wyznaczając X.

Metody bezpośrednie: równania normalne

Algorytm 1 – Układ równań normalnych

- Stwórz $M = A^T A$.
- Rozwiąż równanie $MX = A^T$ wyznaczając X.

Fakt:

$$\kappa(M) = (\kappa(A))^2.$$

Metoda Cholesky'ego

Algorytm 2 – Rozkład Cholesky'ego

- Wyznacz rozkład Cholesky'ego macierzy $M = A^T A$, tj. $M = R^T R$.
- Wyznacz X z równania

$$R^T R X = A^T$$

rozwiązując dwa równania z macierzami trójkątnymi:

- $R^T Y = A^T$,
- $RX = Y$.

Rozkład QR (Householdera)

Algorytm 3 (QR) – Rozkład QR (Householdera)

- Znajdź rozkład QR macierzy A, i.e.

$$A = QR,$$

gdzie $Q \in \mathbb{R}^{m \times n}$ jest kolumnowo ortogonalna, a $R \in \mathbb{R}^{n \times n}$ jest trójkątna górna.

- Założenie $\mathrm{rank}(A) = n$ gwarantuje, że R jest nieosobliwa.
- Macierz $X = A^{\dagger} = R^{-1}Q^{T}$ obliczamy rozwiązując równanie

$$RX = Q^{T}.$$

Bidiagonalizacja Householdera

Algorytm 4 (Bidiag) – Bidiagonalizacja Householdera
(algorytm Goluba-Kahana)

- Za pomocą transformacji Householdera sprowadź A do postaci dwudiagonalnej, tj. znajdź rozkład $A = UBV^T$, gdzie $U \in \mathbb{R}^{m \times n}$ ma ortogonalne kolumny, $V \in \mathbb{R}^{n \times n}$ jest ortogonalna, a $B \in \mathbb{R}^{n \times n}$ jest dwudiagonalna

  Założenie $rank(A) = n$ gwarantuje, że B jest nieosobliwa.

- Macierz $X = A^\dagger = VB^{-1}U^T$ oblicza się w dwóch krokach:
  - wyznacz Y z równania $BY = U^T$,
  - oblicz $X = VY$.

Singular Value Decomposition (SVD)

Twierdzenie. Dla każdej macierzy $A \in \mathbb{C}^{m \times n}$ istnieje rozkład

$$A = U \Sigma V^*,$$

gdzie

- $U \in \mathbb{C}^{m \times m}$, $V \in \mathbb{C}^{n \times n}$ są unitarne,
- $\Sigma \in \mathbb{R}^{m \times n}$, $\Sigma = \mathrm{diag}(\sigma_1, \ldots, \sigma_r, 0, \ldots, 0)$,
  $\underbrace{\sigma_1 \geq \sigma_2 \geq \ldots \geq \sigma_r}_{\text{(wartości szczególne)}} > 0$, $r = \mathrm{rank}(A)$.

Ponadto

$$A^\dagger = V \Sigma^\dagger U^*,$$

gdzie

$$\Sigma^\dagger = \mathrm{diag}(1/\sigma_1, \ldots, 1/\sigma_r, 0, \ldots, 0).$$

SVD

Algorytm 5 – SVD

- Znajdź rozkład SVD macierzy A, tj.

$$A = U\Sigma V^T,$$

gdzie $U \in \mathbb{R}^{m\times n}$ ma ortogonalne kolumny ($U^TU = I_n$), $V \in \mathbb{R}^{n\times n}$ jest ortogonalna, a $\Sigma \in \mathbb{R}^{n\times n}$ jest diagonalna.

- Mamy $X = A^\dagger = V\Sigma^{-1}U^T$.
- Matlab: $X = pinv(A)$.

Złożoność obliczeniowa

Zakładamy, że macierze $Q \in \mathbb{R}^{m \times n}$, $U \in \mathbb{R}^{m \times n}$ oraz $V \in \mathbb{R}^{n \times n}$ są wyznaczane explicite.

Złożoność obliczeniowa algorytmów wyznaczania $X = A^{\dagger}$ przy założeniu, że $A \in \mathbb{R}^{m \times n}$ jest pełnego rzędu, liczona jako liczba elementarnych operacji arytmetycznych: $+, -, *, /$.

Algorytm	Liczba operacji arytm.
Cholesky	$3mn^2 + n^3/3$
QR (Householder)	$5mn^2 - 4/3n^3$
Bidiag (Golub-Kahan)	$8mn^2 - 2/3n^3$
SVD	$16mn^2 + 8n^3$

Własności numeryczne algorytmów 2–5

Stabilność i poprawność numeryczna rozważanych algorytmów obliczania $X = A^{\dagger}$.

Algorytm	Stabilność	Poprawność num.
Cholesky	NIE	NIE
QR (Householder)	TAK	NIE
Bidiag (Golub-Kahan)	TAK	TAK
SVD	TAK	TAK

Testy numeryczne

Macierze Pascala:

$$A = \begin{pmatrix} 1 & 1 & 1 & \dots & 1 \\ 1 & 2 & 3 & \dots & n \\ 1 & 3 & 6 & \dots & a_{3,n-1} + a_{2,n} \\ \vdots & \vdots & \vdots & \ddots & \vdots \\ 1 & n & a_{n-1,3} + a_{n,2} & \dots & a_{n,n-1} + a_{n-1,n} \end{pmatrix}.$$

Współczynnik stabilności:

$$e_{algorithm} = \frac{\|\tilde{X}_{algorithm} - A^{\dagger}\|_2}{\varepsilon_M \|A^{\dagger}\|_2 \kappa_2(A)}.$$

Testy numeryczne

Wartości współczynnika stabilności algorytmów 2-5 obliczania $X = A^{\dagger}$, gdzie A jest macierzą Pascala.

n	$\kappa_2(A)$	e_{Chol}	e_{QR}	e_{bidiag}	e_{pinv}
4	6.91e+2	8.77e+0	2.60e–2	5.92e–2	1.14e–1
6	1.10e+5	9.89e+2	1.80e–2	1.04e–2	2.12e–3
8	2.06e+7	1.64e+4	4.53e–3	6.24e–3	2.05e–2
10	4.15e+9	porażka	7.35e–4	3.87e–3	3.16e–3

Testy numeryczne (metoda QR)

Macierz:

```
randn('state',0)
m = 24;   n = 2;
B = hilb(m);   A1 = ones(m,n) - B(:,1:n)*c;
B = pascal(m);   A2 = B(:,1:n);
A3 = randn(m,n) - A1;
A4 = A1 + 1.1e-7*randn(m,n);   A5 = A2 - 1.1e-7*randn(m,n);
B = magic(m);   A6 = B(:,1:n);
A = [A1   A2   A6+A2   A3   A4   A5-A4];
```

Testy numeryczne (metoda QR)

Macierz:

```
randn('state',0)
m = 24;   n = 2;
B = hilb(m);   A1 = ones(m,n) - B(:,1:n)*c;
B = pascal(m);   A2 = B(:,1:n);
A3 = randn(m,n) - A1;
A4 = A1 + 1.1e-7*randn(m,n);   A5 = A2 - 1.1e-7*randn(m,n);
B = magic(m);   A6 = B(:,1:n);
A = [A1   A2   A6+A2   A3   A4   A5-A4];
```

$$error_{QR} = \frac{\| |\tilde{R}^{-1}| \, |\tilde{R}| \, |\tilde{X}| \|_2}{\|\tilde{X}\|_2}$$

Twierdzenie. Jeśli powyższy współczynnik jest rzędu jedności, algorytmy QR są numerycznie poprawne.

Testy numeryczne (metoda QR)

c	$\kappa_2(A)$	$error_{QR}$	$error_{QR_{pivot}}$
1	1.44e+10	3.92e+2	5.21e+0
10^{-1}	1.37e+10	6.97e+3	5.66e+0
10^{-2}	1.25e+10	4.30e+4	5.66e+0
10^{-3}	1.29e+10	4.76e+5	5.79e+0
10^{-4}	1.36e+10	3.58e+6	5.81e+0
10^{-5}	1.20e+10	4.23e+7	5.25e+0
10^{-6}	5.58e+10	1.01e+8	1.20e+0
10^{-8}	6.62e+12	1.26e+8	1.55e+0

Podsumowanie

- Metody oparte na równaniach normalnych ($A^\dagger = (A^T A)^{-1} A^T$), (np. metoda wykorzystująca rozkład Cholesky'ego) są najszybsze, jednak mogą być stosowane tylko dla zadań dobrze uwarunkowanych.

  Ich dokładność zależy od $(\kappa(A))^2$, a nie od $\kappa(A)$.

Podsumowanie

- Metody oparte na równaniach normalnych ($A^{\dagger} = (A^T A)^{-1} A^T$), (np. metoda wykorzystująca rozkład Cholesky'ego) są najszybsze, jednak mogą być stosowane tylko dla zadań dobrze uwarunkowanych.

  Ich dokładność zależy od $(\kappa(A))^2$, a nie od $\kappa(A)$.

- W wielu przypadkach algorytm 3 (QR) daje zadowalające wyniki.

Podsumowanie

- Metody oparte na równaniach normalnych ($A^\dagger = (A^T A)^{-1} A^T$), (np. metoda wykorzystująca rozkład Cholesky'ego) są najszybsze, jednak mogą być stosowane tylko dla zadań dobrze uwarunkowanych.

  Ich dokładność zależy od $(\kappa(A))^2$, a nie od $\kappa(A)$.

- W wielu przypadkach algorytm 3 (QR) daje zadowalające wyniki.

- Algorytmy 4 (z wykorzystaniem bidiagonalizacji macierzami Householdera) oraz 5 (SVD) są numerycznie poprawne.