WYKŁAD 6

INTERPOLACJA FUNKCJAMI SKLEJANYMI (SPLAJNY)

W tym wykładzie omówimy problem interpolacji przy pomocy tzw. funkcji sklejanych, zwanych też (żargonowo) **splajnami**. W przeciwieństwie do metod interpolacyjnych opisanych w Wykładzie nr 1, gdzie stosowaliśmy jeden globalny wielomian dla całego przedziału interpolacji, w metodzie splajnów stosowane są funkcje zdefiniowane jako wielomiany niskiego stopnia osobno dla każdego odcinka pomiędzy sąsiednimi węzłami interpolacyjnymi. Te lokalne wielomiany są jednak tak dobrane, aby – oprócz warunków interpolacji – spełniać warunki sklejenia w taki sposób, aby cały splajn był funkcją o odpowiedniej regularności. Skoncentrujemy się przede wszystkim na zagadnieniu interpolacji za pomocą **splajnu kubicznego**, tj. funkcji ciągłej wraz z pochodnymi do rzędu drugiego włącznie i zbudowanej z wielomianów 3-ego stopnia.

Należy wspomnieć, że funkcje sklejane (i ich daleko idące uogólnienia), maja wiele zastosowań praktycznych, w szczególności stanowią podstawowe narzędzie współczesnego projektowanie geometrycznego (CAD).

Rozważmy układ n węzłów interpolacyjnych $\{(x_0, y_0), (x_1, y_1), ..., (x_{n-1}, y_{n-1})\}$, gdzie $a = x_0 < x_1 < .. < x_{n-1} = b$.

Splajnem kubicznym *C* = *C*(*x*) nazywamy funkcje określoną na przedziale [*a*,*b*] i taką, że:

1. $C(x) \in C^2([a,b])$, tj. jest ona ciągła wraz z pierwszą i drugą pochodną w [*a*,*b*].

2. $C_k(x) := C(x)\big|_{[x_k, x_{k+1}]} = a_{k,3}x^3 + a_{k,2}x^2 + a_{k,1}x + a_{k,0}, \;\; k = 0,..,n-2$, tj. wewnątrz każdego podprzedziału funkcja ta jest pewnym wielomianem 3-ego stopnia.

3. Dla każdego węzła funkcja *C(x)* spełnia warunki interpolacji tj.
   $C(x_k) = y_k \quad , \quad k = 0,..,n-1.$

Z powyższego wynika, że wielomiany lokalne muszą spełniać warunki interpolacyjne

$$C_k(x_k) = y_k \quad , \quad k = 0,..,n-1,$$

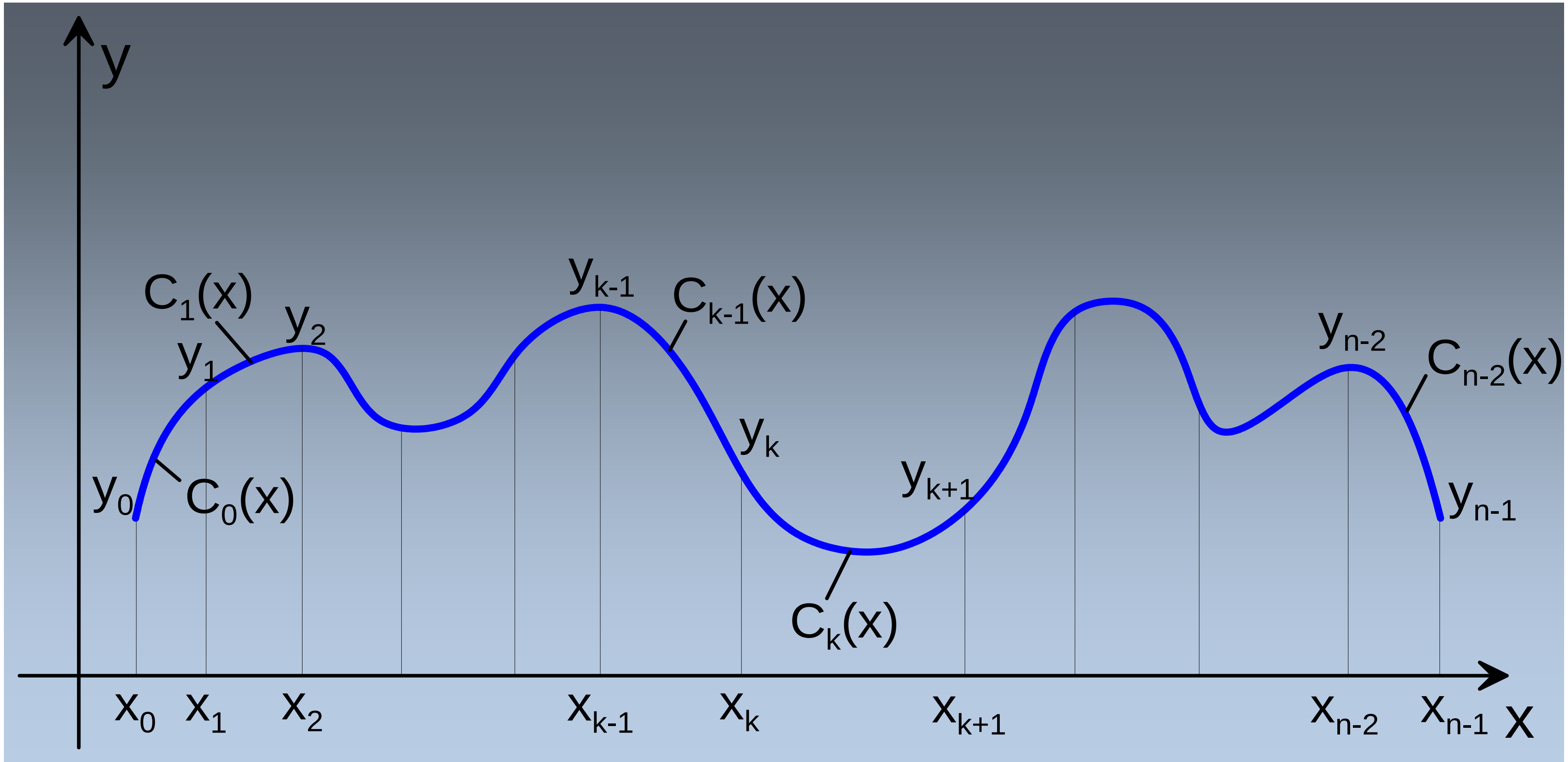
oraz warunki sklejenia zapewniające założoną regularność funkcji *C*, a mianowicie

$$\begin{cases} C_{k-1}(x_k) = C_k(x_k) \\ C'_{k-1}(x_k) = C'_k(x_k) \\ C''_{k-1}(x_k) = C''_k(x_k) \end{cases}$$

dla $k = 1,..,n-2$ (tj. w węzłach wewnętrznych).

Zauważmy, że liczba postawionych warunków jest równa 4*n* - 6. Całkowita liczba nieznanych współczynników lokalnych wielomianów jest natomiast równa 4(*n* - 1) = 4*n* - 4. Zatem, problem wyznaczenia splajna kubicznego jest niedookreślony.

Potrzebujemy nałożyć dwa dodatkowe warunki tak, aby zagadnienie wyznaczenia funkcji *C* miało jednoznaczne rozwiązanie.


y
x
$C_1(x)$
y_1
y_2
y_0
$C_0(x)$
y_{k-1}
$C_{k-1}(x)$
y_k
$C_k(x)$
y_{k+1}
y_{n-2}
$C_{n-2}(x)$
y_{n-1}
x_0
x_1
x_2
x_{k-1}
x_k
x_{k+1}
x_{n-2}
x_{n-1}

Splajn kubiczny

Zwykle (ale nie zawsze) warunki dodatkowe mają formę warunków „brzegowych” nałożonych na pierwszą lub drugą pochodną funkcji *C*, a mianowicie:

$$(C'(x_0)=\alpha \ \text{ lub } \ C''(x_0)=\beta) \ \text{ i } \ (C'(x_{n-1})=\gamma \ \text{ lub } \ C''(x_{n-1})=\delta)$$

W powyższych warunkach liczby α, β, γ, i δ są oczywiście zadane.

Ważnym przypadkiem szczególnym jest tzw. **splajn naturalny**. Jest to taki splajn kubicznym który spełnia warunki postaci

$$C''(x_0)=0 \ , \ C''(x_{n-1})=0$$

Splajn naturalny posiada interesującą własność. Okazuje się, że spośród wszystkich funkcji ciągłych wraz z dwiema pierwszymi pochodnymi i interpolujących zadany układ węzłów **splajn naturalny ma najmniejszą wartość całki z kwadratu drugiej pochodnej na przedziale interpolacji [*a*,*b*]**, tj.

$$\int_a^b \left[C''(x)\right]^2 dx = \min$$

Dokładniej, mamy następujące

TWIERDZENIE: Niech $f \in C^2([x_0, x_{n-1}])$ będzie dowolna. Załóżmy, że $C''(a) = 0$ i $C''(b) = 0$ lub $C'(a) = f'(a)$ i $C'(b) = f'(b)$. Wówczas

$$\int_a^b [C''(x)]^2\,dx \le \int_a^b [f''(x)]^2\,dx$$

Dowód:

$$\int_a^b C''(x)[f''(x) - C''(x)]dx \underset{\substack{\text{przez}\\ \text{czesci}}}{=} C''(x)[f'(x) - C'(x)]\Big|_{x=a}^{x=b} - \int_a^b C'''(x)[f'(x) - C'(x)]dx =$$

$$= \underbrace{C''(b)[f'(b) - C'(b)]}_{=0 \text{ z zalozenia}} - \underbrace{C''(a)[f'(a) - C'(a)]}_{=0 \text{ z zalozenia}} - \int_a^b C'''(x)[f'(x) - C'(x)]dx =$$

$$= -\sum_{k=0}^{n-1} \int_{x_k}^{x_{k+1}} \underbrace{C_k'''(x)\ [f'}_{6a_{k,3} = const}(x) - C'(x)]dx = -6\sum_{k=0}^{n-1} a_{k,3} \int_{x_k}^{x_{k+1}} [f'(x) - C'(x)]dx =$$

$$= -6\sum_{k=0}^{n-1} a_{k,3} \underbrace{[f(x) - C(x)]\Big|_{x=x_k}^{x=x_{k+1}}}_{\substack{=0\\ \text{bo } C(x_k) = f(x_k),\ k=0,1,..,n}} = 0$$

Otrzymaliśmy równość $\int_a^b C''(x) f''(x) dx = \int_a^b [C''(x)]^2 dx.$

Dalej mamy

$$0 \le \int_a^b [f''(x) - C''(x)]^2 dx = \int_a^b [f''(x)]^2 dx - 2 \underbrace{\int_a^b f''(x) C''(x) dx}_{= \int_a^b [C''(x)]^2 dx} + \int_a^b [C''(x)]^2 dx =$$

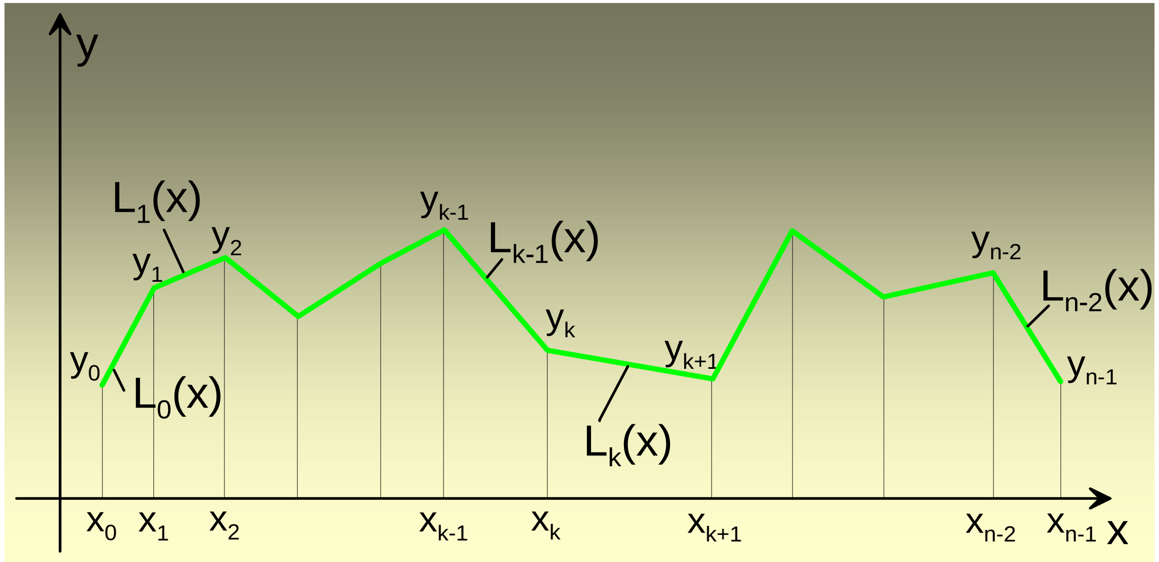
$$= \int_a^b [f''(x)]^2 dx - \int_a^b [C''(x)]^2 dx$$

czyli $\int_a^b [f''(x)]^2 dx \ge \int_a^b [C''(x)]^2 dx.$ **Koniec dowodu.**

Pozostaje kwestia jak w efektywny sposób wyznaczyć (skonstruować) splajn kubiczny dla zadanego układu węzłów. W teorii, moglibyśmy zbudować i rozwiązać układ równań (liniowych) dla nieznanych współczynników lokalnych wielomianów C_k ($k = 0,1,..,n-2$). Układ taki zawierałby $4n$-4 równań, a macierz współczynników miałaby dość „paskudną" strukturę.

Okazuje się (jak zwykle?), że istnieje alternatywna metoda inteligentna!

Zacznijmy od spostrzeżenia, że druga pochodna poszukiwanej funkcji sklejanej C jest funkcją kawałkami liniową (mówimy też – jest splajnem liniowym). Można ją zapisać następująco:



$$C''(x)\Big|_{[x_k,x_{k+1}]} \equiv C_k''(x) =$$

$$= C_x''(x_k)\,\frac{x_{k+1}-x}{x_{k+1}-x_k} + C_x''(x_{k+1})\,\frac{x-x_k}{x_{k+1}-x_k}$$

lub

$$C_k''(x) = m_k\,\frac{x_{k+1}-x}{h_k} + m_{k+1}\,\frac{x-x_k}{h_k}$$

gdie $\quad m_k \equiv C''(x_k)\ ,\ h_k = x_{k+1}-x_k$

Całkując dwukrotnie powyższą formułę otrzymamy postać lokalnego wielomianu C_k

$$C_k(x) = \frac{m_k}{6h_k}(x_{k+1}-x)^3 + \frac{m_{k+1}}{6h_k}(x-x_k)^3 + p_k(x_{k+1}-x) + q_k(x-x_k)$$

przy czym symbole p_k i q_k oznaczają stałe całkowania. Czytelnik będzie uprzejmy upewnić się, że druga pochodna funkcji C_k ma istotnie odpowiednią postać.

Póki co, stałe całkowania były dowolne. **Teraz dobierzemy je tak, aby spełnić warunki interpolacji**

$$C_k(x_k) = y_k \quad , \ C_k(x_{k+1}) = y_{k+1}$$

Otrzymujemy wartości stałych p_k i q_k

$$y_k = \tfrac{1}{6} m_k h_k^2 + p_k h_k \quad \Rightarrow \quad p_k = \frac{y_k}{h_k} - \tfrac{1}{6} m_k h_k$$

$$y_{k+1} = \tfrac{1}{6} m_{k+1} h_k^2 + q_k h_k \quad \Rightarrow \quad q_k = \frac{y_{k+1}}{h_k} - \tfrac{1}{6} m_{k+1} h_k$$

i w konsekwencji **ostateczna postać lokalnego wielomianu** wyraża się formułą

$$C_k(x) = \frac{m_k}{6h_k}(x_{k+1} - x)^3 + \frac{m_{k+1}}{6h_k}(x - x_k)^3 + \left(\frac{y_k}{h_k} - \tfrac{1}{6} m_k h_k\right)(x_{k+1} - x) + \left(\frac{y_{k+1}}{h_k} - \tfrac{1}{6} m_{k+1} h_k\right)(x - x_k)$$

przy czym indeks k przyjmuje wartości od 0 do n-2.

Pozostało obliczyć wartości drugiej pochodnej splajnu w węzłach, czyli wielkości $m_0, m_1, \ldots, m_{n-1}$.

Zauważmy, że nie wykorzystaliśmy jeszcze warunku „dopasowania” pierwszej pochodnej sąsiadujących wielomianów lokalnych. Różniczkując otrzymaną wyżej formułę dla C_k mamy

$$C_k'(x) = -\frac{m_k}{2h_k}(x_{k+1} - x)^2 + \frac{m_{k+1}}{2h_k}(x - x_k)^2 + \frac{y_{k+1} - y_k}{h_k} - \tfrac{1}{6}(m_{k+1} - m_k)h_k$$

Rozważmy węzeł x_k. Z powyższego wzoru wynika dla $x = x_k$, że

$$C_k'(x_k) = -\tfrac{1}{3}m_k h_k - \tfrac{1}{6}m_{k+1}h_k + \underbrace{\frac{y_{k+1} - y_k}{h_k}}_{d_k} = -\tfrac{1}{3}m_k h_k - \tfrac{1}{6}m_{k+1}h_k + d_k$$

Wartość pierwszej pochodnej wielomianu C_{k-1} w węźle x_k otrzymamy następująco: najpierw w ogólnej formule dla 1-szej pochodnej wielomianu C_k podmienimy formalnie k na k-1, a następnie podstawimy $x = x_k$. Oto rezultat (sprawdzić!)

$$C_{k-1}'(x_k) = \tfrac{1}{3}m_k h_{k-1} + \tfrac{1}{6}m_{k-1}h_{k-1} + \underbrace{\frac{y_k - y_{k-1}}{h_{k-1}}}_{d_{k-1}} = \tfrac{1}{3}m_k h_{k-1} + \tfrac{1}{6}m_{k-1}h_{k-1} + d_{k-1}$$

Z warunków ciągłości 1-szej pochodnej w węzłach mamy $C'_{k-1}(x_k) = C'_k(x_k)$, co po podstawieniu otrzymanych wzorów i prostych przekształceniach prowadzi do następującego układu równań dla wielkości $m_0, m_1, ..., m_{n-1}$

$$h_{k-1}m_{k-1} + 2(h_{k-1} + h_k)m_k + h_k m_{k+1} = u_k \ , \ k = 1, 2, .., n-2$$

gdzie oznaczyliśmy $u_k = 6(d_k - d_{k-1}) = 6\left(\frac{y_{k+1} - y_k}{h_k} - \frac{y_k - y_{k-1}}{h_{k-1}}\right)$.

Wiemy już, że do wyznaczenia funkcji sklejanej C potrzebujemy dwóch dodatkowych warunków. Jeśli zdecydujemy się na **określenie wartości pierwszej pochodnej w węzłach skrajnych** to warunki te przyjmą postać

$$C'(x_0) = -\tfrac{1}{3}h_0 m_0 - \tfrac{1}{6}h_0 m_1 + d_0 = \alpha \quad \Rightarrow \quad 2h_0 m_0 + h_0 m_1 = 6(d_0 - \alpha)$$

$$C'(x_{n-1}) = \tfrac{1}{3}h_{n-2}m_{n-1} + \tfrac{1}{6}h_{n-2}m_{n-2} + d_{n-2} = \gamma \quad \Rightarrow \quad h_{n-2}m_{n-2} + 2h_{n-2}m_{n-1} = 6(\gamma - d_{n-2})$$

Jeśli natomiast określimy wartości brzegowe drugiej pochodnej to dodatkowe równania są bardzo proste, a mianowicie $m_0 = \beta$ i $m_{n-1} = \delta$. **W szczególności, jeśli $\beta = \delta = 0$ to otrzymamy splajn kubiczny naturalny**.

Podsumowując, kompletny układ równań liniowych dla wartości drugiej pochodnej splajna kubicznego w węzłach może być zapisany następująco:

$$\begin{cases} m_0 = \beta \quad \text{lub} \quad 2h_0 m_0 + h_0 m_1 = 6(d_0 - \alpha) \quad , \quad k = 0 \\ h_{k-1} m_{k-1} + 2(h_{k-1} + h_k) m_k + h_k m_{k+1} = u_k \quad , \quad k = 1,..,n-2 \\ m_{n-1} = \delta \quad \text{lub} \quad h_{n-2} m_{n-2} + 2h_{n-2} m_{n-1} = 6(\gamma - d_{n-2}) \quad , \quad k = n-1 \end{cases}$$

W notacji macierzowo-wektorowej: $\mathbf{Tm} = \mathbf{r}$. Zauważmy, że macierz $\mathbf{T}$ jest **trójdiagonalna**. Niezerowe elementy tej macierzy można zapisać jako elementy trzech wektorów **a**,**b** and **c**.

$$\mathbf{T} = \begin{bmatrix} c_0 & -b_0 & 0 & 0 & 0 & 0 & 0 \\ -a_1 & c_1 & -b_1 & 0 & 0 & 0 & 0 \\ 0 & \ddots & \ddots & \ddots & 0 & 0 & 0 \\ 0 & 0 & -a_k & c_k & -b_k & 0 & 0 \\ 0 & 0 & 0 & \ddots & \ddots & \ddots & 0 \\ 0 & 0 & 0 & 0 & -a_{n-2} & c_{n-2} & -b_{n-2} \\ 0 & 0 & 0 & 0 & 0 & -a_{n-1} & c_{n-1} \end{bmatrix}$$

Wartości zapisane w tych wektorach zależą od przyjętego wariantu „warunków brzegowych" i przedstawiają się następująco:

$$\begin{cases} a_0 = 0 \quad (\textit{nie używany}) \\ a_k = -h_{k-1} \quad , \; k = 1,..,n-2 \\ a_{n-1} = 0 \;\; \textit{lub} \;\; a_{n-1} = -h_{n-2} \end{cases} \qquad \begin{cases} b_0 = 0 \;\; \textit{lub} \;\; b_0 = -h_0 \\ b_k = -h_k \quad , \; k = 1,..,n-2 \\ b_{n-1} = 0 \;\; (\textit{nie używany}) \end{cases} \qquad \begin{cases} c_0 = 1 \;\; \textit{lub} \;\; c_0 = 2h_0 \\ c_k = 2(h_{k-1} + h_k) \; , \; k = 1,..,n-2 \\ c_{n-1} = 1 \;\; \textit{lub} \;\; c_{n-1} = 2h_{n-2} \end{cases}$$

Do efektywnego rozwiązania układu liniowego z macierzą trójdiagonalną stosujemy specjalny wariant metody eliminacji Gaussa zwany **algorytmem przeganiania (Thomasa)**. Przedstawimy ten algorytm zakładając, że rozwiązywany układ równań ma postać

$$\begin{cases} c_0 m_0 - b_0 m_1 = r_0 \\ -a_k m_{k-1} + c_k m_k - b_k m_{k+1} = r_k \quad , \quad k = 1,..,n-2 \\ -a_{n-1} m_{n-2} + c_{n-1} m_{n-1} = r_{n-1} \end{cases}$$

Metoda przeganiania (algorytm Thomasa)

Rozważmy dwa pierwsze równania układu

$$\begin{cases} c_0 m_0 - b_0 m_1 = r_0 \\ -a_1 m_0 + c_1 m_1 - b_1 m_2 = r_1 \end{cases}$$

i załóżmy, że $c_0 \neq 0$. Najpierw „normalizujemy" pierwsze równanie dzieląc je przez c_0, następnie eliminujemy m_0 metodą przeciwnych współczynników

$$\begin{cases} m_0 - \alpha_0 m_1 = \beta_0 \quad , \quad \alpha_0 = b_0 / c_0 \quad , \beta_0 = r_0 / c_0 \\ -a_1 m_0 + c_1 m_1 - b_1 m_2 = r_1 \end{cases}$$

i w końcu sprowadzamy zmodyfikowane drugie równanie do postaci „znormalizowanej"

$$(c_1 - \alpha_0 a_1) m_1 - b_1 m_2 = r_1 + a_1 \beta_0 \quad / : (c_1 - \alpha_0 a_1)$$

$$\Downarrow$$

$$m_1 - \alpha_1 m_2 = \beta_1 \quad , \quad \alpha_1 = \frac{b_1}{c_1 - \alpha_0 a_1} \quad , \quad \beta_1 = \frac{r_1 + a_1 \beta_0}{c_1 - \alpha_0 a_1}$$

W trakcie obliczeń pojawiają się dwie pomocnicze wielkości α_1 and β_1.

Zauważmy, że nowy zredukowany układ równań z niewiadomymi $m_1,\dots,m_{n-1}$ wygląda „tak samo” jak oryginalny, tj. ma strukturę 3-diagonalną i pierwsze z równań ma postać 3-diagonalną i pierwsze równanie ma postać znormalizowaną. Układ ten jest zatem gotowy do kontunuowania procedury eliminacji kolejnych niewiadomych. Po k krokach procedury eliminacyjnej dochodzimy do etapu, w którym układ zwiera niewiadome o numerach od k do n-1. Następny krok polega na eliminacji niewiadomej m_k metodą przeciwnych współczynników. W szczegółach wygląda to następująco

$$
\begin{array}{l}
\begin{cases}
m_k - \alpha_k m_{k+1} = \beta_k \qquad\qquad\qquad / \cdot a_{k+1} \\
-a_{k+1} m_k + c_{k+1} m_{k+1} - b_{k+1} m_{k+2} = r_{k+1}
\end{cases} \\
\hline
(c_{k+1} - \alpha_k a_{k+1}) m_{k+1} - b_{k+1} m_{k+2} = r_{k+1} + \beta_k b_{k+1}
\end{array}
$$

$$\Downarrow$$

$$m_{k+1} - \alpha_{k+1} m_{k+2} = \beta_{k+1} \;, \quad \alpha_{k+1} = \frac{b_{k+1}}{c_{k+1} - \alpha_k a_{k+1}} \;, \quad \beta_{k+1} = \frac{r_{k+1} + a_{k+1}\beta_k}{c_{k+1} - \alpha_k a_{k+1}}$$

Podczas obliczeń pojawia się kolejna para wielkości pomocniczych α_{k+1} and β_{k+1}.

Ostatecznie, po n-1 krokach eliminacji proces osiąga ostatnie równanie układu. Ostatni krok eliminacji przebiega następująco

$$\begin{cases} m_{n-2} - \alpha_{n-2} m_{n-1} = \beta_{n-2} & / \cdot a_{n-1} \\ -a_{n-1} m_{n-2} + c_{n-1} m_{n-1} = r_{n-1} & \end{cases}$$

$$\overline{(c_{n-1} - \alpha_{n-2} a_{n-1}) m_{n-1} = r_{n-1} + \beta_{n-2} b_{n-1}}$$

$$\Downarrow$$

$$m_{n-1} = \frac{r_{n-1} + \beta_{n-2} b_{n-1}}{c_{n-1} - \alpha_{n-2} a_{n-1}}$$

Zauważmy, że ostatnie równanie układu zawiera jedynie dwie niewiadome (przedostatnią i ostatnią), przez co wyniku eliminacji m_{n-2} otrzymujemy równanie z tylko jedną niewiadomą m_{n-1}. Zauważmy również, że podczas procesu eliminacji otrzymaliśmy rekursywną formułę wiążącą dwie niewiadome o kolejnych numerach. Ogólna postać tej formuły wynika z pierwszego równania (znormalizowanego) zredukowanego układu po k krokach eliminacji, a mianowicie

$$m_k = \beta_k + \alpha_k m_{k+1} \ , \ k = n-2, n-3, \ldots, 1, 0$$

W ten sposób wszystkie wyeliminowane wcześniej niewiadome mogą być wyznaczone w pętli chodzącej wstecz.

Metodę przeganiania (algorytm Thomasa) można podsumować następująco:

$\underline{ETAP\ 1\ (sweep-up)}$

$\alpha_0 = b_0 / c_0 \quad ; \quad \beta_0 = r_0 / c_0\ ;$

$for\ \ k = 0,..,n-3\ \ do:$

$$\alpha_{k+1} = \frac{b_k}{c_{k+1} - \alpha_k a_{k+1}} \quad ;$$

$$\beta_{k+1} = \frac{r_{k+1} + \beta_k a_{k+1}}{c_{k+1} - \alpha_k a_{k+1}} \quad ;$$

$end\ ;$

$\underline{ETAP\ 2\ (sweep-down)}$

$$m_{n-1} = \frac{r_{n-1} + \beta_{n-2} a_{n-1}}{c_{n-1} - \alpha_{n-2} a_{n-1}};$$

$for\ k = n-2,..,0\ \ do:$ $!\ ta\ \ petla\ chodzi\ wstecz!$

$$m_k = \beta_k + \alpha_k m_{k+1}$$

$end\ ;$

UWAGA:

Ponieważ metoda przeganiania jest pewnym wariantem metody eliminacji Gaussa (bez wyboru elementu głównego – vide Wykład 7), to na **macierz trójdiagonalną należy nałożyć pewne ograniczenia gwarantujące powodzenie obliczeń**. Chodzi przede wszystkim o gwarancję, że wszystkie operacje dzielenia będą wykonalne (nie wystąpi dzielenie przez zero).

Można pokazać, że **warunki wystarczające** dla powodzenia przebiegu obliczeń metodą przeganiania mają następującą postać:

1) $c_0 \neq 0, c_{n-1} \neq 0 , a_k \neq 0 , b_k \neq 0 , k = 1,..,n-2$

2) *warunki diagonalnej dominacji*

$$|c_k| \geq |a_k| + |b_k| \quad , \quad i = 1,..,n-2$$

$$|c_0| \geq |b_0| \quad , \quad |c_{n-1}| \geq |a_{n-1}|$$

Przynajmniej jedna z tych nierownosci musi byc OSTRA!