# Wstęp do analizy numerycznej

Wykład dla studentów kierunku Matematyka

Michał Góra
Wydział Matematyki Stosowanej AGH

Kraków 2026

# Spis treści

**Rozdział 1. Reprezentacja liczb w komputerze, arytmetyka zmiennopozycyjna** fl . 5
1.1. Reprezentacja liczb całkowitych . . . 5
1.2. Reprezentacja liczb rzeczywistych . . . 6
1.3. Reprezentacja wektorów oraz macierzy . . . 9
1.4. Arytmetyka zmiennopozycyjna fl . . . 9
1.5. Upraszczanie oszacowań . . . 10
**Rozdział 2. Uwarunkowanie zadania obliczeniowego, kryteria akceptowalności algorytmów** . . . 13
2.1. Uwarunkowanie zadania obliczeniowego . . . 13
2.2. Kryteria akceptowalności algorytmów . . . 17
2.2.1. Algorytmy numerycznie poprawne . . . 17
2.2.2. Algorytmy numerycznie stabilne . . . 18
**Rozdział 3. Interpolacja wielomianowa Lagrange'a** . . . 20
3.1. Sformułowanie problemu, istnienie i jedyność rozwiązania . . . 20
3.1.1. Macierz Vandermonde'a . . . 21
3.2. Błąd interpolacji . . . 22
3.3. Różnice dzielone . . . 22
3.3.1. Algorytm różnic dzielonych . . . 25
3.4. Zbieżność procesu interpolacji ☠ . . . 26
**Rozdział 4. Interpolacja wielomianowa Hermite'a** . . . 28
4.1. Sformułowanie problemu, istnienie i jedyność rozwiązania . . . 28
4.2. Błąd interpolacji . . . 30
**Rozdział 5. Interpolacja funkcjami sklejanymi** . . . 32
5.1. Funkcje sklejane oraz naturalne funkcje sklejane . . . 32
5.1.1. Reprezentacje funkcji sklejanych . . . 32
5.2. Interpolacja funkcjami sklejanymi . . . 34
5.3. Kubiczne funkcje sklejane . . . 36
5.3.1. Wyznaczanie interpolacyjnej naturalnej funkcji sklejanej stopnia 3 ☠ . . . 36
5.3.2. Twierdzenie o błędzie interpolacji dla naturalnej kubicznej funkcji sklejanej ☠ . . . 38
5.3.3. Zbieżność procesu interpolacji ☠ . . . 39
**Rozdział 6. Interpolacja trygonometryczna** . . . 40
6.1. Sformułowanie problemu, istnienie i jedyność rozwiązania . . . 40
6.2. Postać rozwiązania dla węzłów równoodległych . . . 41
6.3. Rzeczywiste trygonometryczne wielomiany interpolacyjne dla węzłów równoodległych . . . 42
**Rozdział 7. Aproksymacja w przestrzeni unormowanej** . . . 45
7.1. Sformułowanie problemu, istnienie i jedyność rozwiązania . . . 45
7.2. Przestrzenie silnie unormowane . . . 47
**Rozdział 8. Aproksymacja średniokwadratowa** . . . 49
8.1. Sformułowanie problemu, istnienie i jedyność rozwiązania . . . 49
8.1.1. Element optymalny jako rzut ortogonalny na podprzestrzeń . . . 50
8.1.2. Wyznaczanie elementu optymalnego – układ równań normalnych . . . 51

8.2. Wielomiany ortogonalne . . . 53
8.2.1. Przykłady wielomianów ortogonalnych . . . 53
8.2.2. Własności wielomianów ortogonalnych . . . 53
8.2.3. Wielomiany Czebyszewa pierwszego rodzaju . . . 54
**Rozdział 9. Aproksymacja jednostajna** . . . 58
9.1. Sformułowanie problemu, istnienie i jedyność rozwiązania . . . 58
9.1.1. Warunek Haara . . . 58
9.1.2. Twierdzenie Czebyszewa o alternansie . . . 59
**Rozdział 10. Kwadratury, czyli metody przybliżonego obliczania całek oznaczonych** 61
10.1. Motywacja . . . 61
10.2. Kwadratury liniowe, rząd kwadratury . . . 61
10.2.1. Klasyczne kryterium oceny jakości kwadratury . . . 62
10.3. Kwadratury interpolacyjne . . . 62
10.3.1. Rząd kwadratury interpolacyjnej . . . 62
10.3.2. Kwadratury Gaussa . . . 64
10.3.3. Kwadratury Newtona-Cotes . . . 66
10.4. Kwadratury złożone . . . 67
10.4.1. Sposób konstrukcji kwadratur złożonych . . . 67
10.4.2. Przykłady kwadratur złożonych . . . 68
**Rozdział 11. Rozwiązywanie układów równań liniowych – metody skończone** . . . 71
11.1. Rozwiązywanie układów równań o macierzach o szczególnej postaci . . . 71
11.1.1. Rozwiązywanie układów o macierzy trójkątnej . . . 71
11.1.2. Rozwiązywanie układów o macierzy ortogonalnej . . . 73
11.2. Metoda eliminacji Gaussa . . . 73
11.2.1. Realizacja metody eliminacji . . . 73
11.2.2. Zapis macierzowy metody eliminacji . . . 74
11.2.3. Koszt i właściwości numeryczne . . . 76
11.2.4. Wybór elementu głównego . . . 77
11.2.5. Zastosowanie metody eliminacji Gaussa . . . 80
11.3. Metoda Householdera . . . 81
11.3.1. Elementarne przekształcenie Householdera . . . 81
11.3.2. Algorytm Householdera . . . 83
11.3.3. Zastosowanie algorytmu Householdera . . . 85
11.4. Metoda Grama-Schmidta . . . 86
11.4.1. Algorytm ortogonalizacyjny . . . 86
11.4.2. Koszt i właściwości numeryczne . . . 87
**Rozdział 12. Metody iteracyjne rozwiązywania układów równań liniowych** . . . 89
12.1. Stacjonarne metody iteracyjne . . . 89
12.2. Zbieżność stacjonarnych metod iteracyjnych . . . 90
12.3. Przykłady stacjonarnych metod iteracyjnych . . . 93
12.3.1. Metoda Jacobiego . . . 93
12.3.2. Metoda Gaussa-Seidela . . . 94
**Rozdział 13. Metody iteracyjne rozwiązywania równań nieliniowych** . . . 96
13.1. Metody iteracyjne . . . 96
13.1.1. Rodzaje metod iteracyjnych . . . 96
13.1.2. Wykładnik zbieżności . . . 97
13.2. Przykłady metod iteracyjnych . . . 97
13.2.1. Metoda bisekcji . . . 97
13.2.2. Metoda stycznych (Newtona) . . . 98
13.2.3. Metoda siecznych . . . 100
**Rozdział 14. Algebraiczny problem własny** . . . 101
14.1. Sformułowanie zadania . . . 101
14.2. Uwarunkowanie ☠ . . . 102
14.3. Lokalizacja wartości własnych, twierdzenie Gerszgorina . . . 103
14.4. Metoda potęgowa oraz ilorazy Rayleigha . . . 105
14.5. Deflacja . . . 107

**Dodatek A. Norma macierzowa indukowana przez normę wektorową** . . . . . 108
**Dodatek B. Twierdzenie Banacha o punkcie stałym** . . . . . 111
**Dodatek C. Poetycki wielomian Wilkinsona** . . . . . 113

## Rozdział 1

# Reprezentacja liczb w komputerze, arytmetyka zmiennopozycyjna $\mathbb{fl}$

Sposób reprezentacji liczb w komputerze zależy od ich typu. Liczby całkowite (tj. typu *integer*) zapisujemy (kodujemy) inaczej niż liczby rzeczywiste (tj. typu *real*). W konsekwencji, w świecie obliczeń realizowanych na komputerze zbiór liczb całkowitych nie jest podzbiorem zbioru liczb wymiernych, a liczby niewymierne nie występują w nim w ogóle.

## 1.1. Reprezentacja liczb całkowitych

Każdą liczbę całkowitą $x \in \mathbb{Z}$ możemy w sposób jednoznaczny zapisać w postaci

$$x = s \sum_{i=0}^{n} a_i \beta^i, \tag{1.1}$$

gdzie

- $\beta \in \mathbb{N}$, $\beta \geqslant 2$ to podstawa przyjętego systemu,
- $s \in \{-1, 1\}$ to znak liczby $x$,
- $a_0, \ldots, a_n \in \{0, \ldots, \beta - 1\}$ to współczynniki.

W komputerze liczby zapisujemy w słowach (rejestrach) o skończonej długości $d + 1$. Jedno miejsce rejestru przeznaczone jest na zakodowanie znaku $s$ liczby $x$: $s = 0$ oznacza, że $x \geqslant 0$, natomiast $s = 1$, że $x < 0$; pozostałe $d$ miejsc przeznaczonych jest na zakodowanie współczynników $a_0, \ldots, a_n$. Jeżeli $d > n$, tj. liczba dostępnych komórek rejestru jest większa niż liczba współczynników, to mówimy, że liczba $x$ jest *reprezentowalna* w danym systemie (tj. przy ustalonym $\beta$ oraz $d$). Wynika stąd, że przy ustalonych podstawie systemu $\beta$ oraz długości rejestru $d+1$ istnieje największa reprezentowalna liczba całkowita dodatnia $x_{\max}$ oraz najmniejsza reprezentowalna liczba całkowita ujemna $x_{\min}$. Łatwo wykazać korzystając z (1.1) oraz z przyjętych ograniczeń, że

$$x_{\max} = \sum_{i=0}^{d-1} (\beta - 1)\beta^i = \beta^d - 1 \quad \text{oraz} \quad x_{\min} = -x_{\max}, \tag{1.2}$$

a zbiór wszystkich reprezentowalnych liczb całkowitych jest skończony. Zanotujmy również, że działania arytmetyczne wykonywane na liczbach całkowitych realizowane są w ten sposób, że dla $* \in \{+, -, \cdot\}$ zachodzi

$$x, y \in \mathbb{Z} \nRightarrow x * y \text{ dokładne},$$

ale

$$x, y \in \mathbb{Z},\ x, y, x * y \text{ – reprezentowalne } \Rightarrow x * y \text{ dokładne.}$$

Powyższa zależność zazwyczaj nie jest prawdziwa dla liczb rzeczywistych, które kodowane są w zupełnie inny sposób.

## 1.2. Reprezentacja liczb rzeczywistych

Można wykazać (zrobimy to na ćwiczeniach), że dla każdej liczby rzeczywistej $x \neq 0$ istnieją jednoznacznie wyznaczone liczby $s, c, m$, takie że

$$x = s \cdot \beta^c \cdot m,$$

gdzie

- $\beta \in \mathbb{N}$, $\beta \geqslant 2$ to podstawa przyjętego systemu,
- $s \in \{-1, 1\}$ to znak liczby $x$,
- $c \in \mathbb{Z}$ to cecha liczby $x$,
- $m \in [\beta^{-1}, 1)$ to mantysa liczby $x$.

**Przykład 1.1.** *Dla $x = \pi$ oraz $\beta = 10$ mamy*

$$\pi = 1 \cdot 10^1 \cdot \frac{\pi}{10} = 1 \cdot 10^1 \cdot 0,3141\ldots,$$

*czyli $s = 1$, $c = 1$, $m = 0,3141\ldots$. Przyjmując $\beta = 2$ otrzymujemy z kolei*

$$\pi = 1 \cdot 2^2 \cdot \frac{\pi}{4} = 1 \cdot 2^2 \cdot 0,7854\ldots,$$

*czyli $s = 1$, $c = 2$, $m = 0,7854\ldots$.*

Podobnie jak w przypadku liczb całkowitych, liczby rzeczywiste zapisujemy w rejestrach długości $d+1$, w których jedno miejsce przeznaczone jest na zapisanie informacji o znaku kodowanej liczby, na kolejnych $d-t$ miejscach zapisujemy cechę, ostatnie $t$ miejsc umożliwia przechowywanie informacji o wartości mantysy. Zakładamy (w praktyce założenie to jest najczęściej spełnione), że cecha należy do zbioru liczb reprezentowalnych w ten sposób (tj. może być dokładnie zapisana).

| | bit znaku | bity kodu cechy | bity kodu mantysy |
|---|---|---|---|
| 32 bity (single precision) | 1 bit | 8 bitów | 23 bity |
| 64 bity (double precision) | 1 bit | 11 bitów | 52 bity |

**Tabela 1.** Format zapisu zmiennopozycyjnego wg standardu IEEE 754.[1]

**Przykład 1.2.** *Przykład liczby zapisanej w formacie 32–bitowym (zob. tabela 1):*

$$x = 10000010110010000110000000000000.$$

*Ponieważ*

$$\underset{znak}{1}\ \ \underset{cecha}{00000101}\ \ \underset{mantysa}{10010000110000000000000},$$

*zatem $x$ jest liczbą ujemną, której cecha to*

$$0\ 0000101_{(2)} = 2^0 + 2^2 = 5_{(10)}$$

[1] Standard IEEE 754 określa m.in. sposób reprezentacji liczb zmiennoprzecinkowych, zasady wykonywania na nich działań arytmetycznych (dodawania, odejmowania, mnożenia i dzielenia), sposób zaokrąglania wyników, wartości specjalne takie jak $+\infty$, $-\infty$, NaN oraz $+0$ i $-0$. W standardzie IEEE 754 zakłada się, że mantysa ma ukrytą (domyślną) część całkowitą równą 1, a w polu mantysy przechowywane są tylko jej bity ułamkowe; pozwala to na dodatkowe zwiększenie precyzji reprezentacji. Wyjątkiem jest liczba zero, która ma specjalny wzorzec bitowy, w którym zarówno pole wykładnika, jak i pole mantysy zawierają same zera.

*a mantysa to*

$$10010000110000000000000_{(2)} = 2^{-1} + 2^{-4} + 2^{-9} + 2^{-10} = 0,5654296875_{(10)}.$$

*Ostatecznie*

$$x = -2^5 \cdot 0,5654296875 = -18,09375.$$

Jak wynika z (1.2), maksymalna wartość cechy zapisanej w rejestrze długości $d-t$ to

$$c_{\max} = \beta^{d-t-1} - 1,$$

co daje ograniczenie

$$-c_{\max} \leqslant c \leqslant c_{\max} \tag{1.3}$$

dla możliwych wartości cechy $c$ dowolnej liczby reprezentowalnej w danym systemie.

**Przykład 1.3.** *W przypadku* 64*-bitowego zapisu binarnego, tj. gdy* $\beta = 2$ *oraz* $d+1 = 64$*, dla* $t = 52$ *mamy*

$$c_{\max} = 2^{10} - 1 = 1023.$$

Liczby, których cecha jest mniejsza niż $-c_{\max}$ są reprezentowane zerem (mamy wówczas do czynienia z *niedomiarem*), w przypadku wystąpienia liczby o cesze większej niż $c_{\max}$ (taką sytuację nazywamy *nadmiarem*), następuje przerwanie obliczeń. Problem niedomiaru lub nadmiaru pojawia się stosunkowo rzadko (można go oczywiście w łatwy sposób wywołać), a w celu uniknięcia go wystarczy czasami prosta modyfikacja algorytmu.

Mantysa, jako liczba z przedziału $[\beta^{-1}, 1)$, może być zapisana w postaci nieskończonej sumy

$$m = \sum_{i=1}^{\infty} a_{-i}\beta^{-i} = 0, a_{-1}a_{-2}\ldots_{(\beta)}, \tag{1.4}$$

gdzie $a_{-1}, a_{-2}, \ldots \in \{0, \ldots, \beta-1\}$, $a_{-1} \neq 0$.[2]

**Przykład 1.4.** *Przyjmując* $\beta = 2$ *wyznaczymy rozwinięcie* (1.4) *liczby* $m = 0,6$.

*Zauważmy, że*

$$\begin{aligned}
2 \cdot 0,6 &= 1 + 0,2, && \textit{tj. } a_{-1} = 1,\\
2 \cdot 0,2 &= 0,4 + 0, && \textit{tj. } a_{-2} = 0,\\
2 \cdot 0,4 &= 0,8 + 0, && \textit{tj. } a_{-3} = 0,\\
2 \cdot 0,8 &= 0,6 + 1, && \textit{tj. } a_{-4} = 1,\\
2 \cdot 0,6 &= 1 + 0,2, && \textit{tj. } a_{-5} = 1,\\
&\textit{itd.}
\end{aligned}$$

*zatem* $m = 0,6_{(10)} = 0,(1001)_{(2)}$.

W celu zapisania informacji o wartości mantysy mamy do dyspozycji rejestr długości $t$, dlatego najczęściej nie będzie ona reprezentowana w sposób dokładny. Aby zakodować informację o mantysie musimy jej nieskończone rozwinięcie (1.4) w rozsądny sposób przybliżyć sumą skończoną. Dość naturalne pomysły na rozwiązanie tego problemu są przynajmniej dwa:

- obcięcie $m \rightsquigarrow \tilde{m}_t$, gdzie

$$\tilde{m}_t = \sum_{i=1}^{t} a_{-i}\beta^{-i};$$

[2] Łatwo o przykład pokazujący, że bez dodatkowych założeń dotyczących możliwych wartości ciągu współczynników $\{a_{-i}\}_{i\in\mathbb{N}}$, reprezentacja ta nie musi być jednoznaczna.

- symetryczne zaokrąglenie $m \rightsquigarrow \mathrm{rd}_t(m)$, gdzie

$$\mathrm{rd}_t(m) = \begin{cases} \sum\limits_{i=1}^{t} a_{-i}\beta^{-i}, & \text{gdy } 0 \leqslant a_{-(t+1)} \leqslant \frac{\beta}{2} - 1 \\ \sum\limits_{i=1}^{t} a_{-i}\beta^{-i} + \beta^{-t}, & \text{gdy } \frac{\beta}{2} \leqslant a_{-(t+1)} \leqslant \beta - 1 \end{cases}.$$

Ponieważ

$$|\tilde{m}_t - m| = \sum_{i=t+1}^{\infty} a_{-i}\beta^{-i} \leqslant \sum_{i=t+1}^{\infty} (\beta - 1)\beta^{-i} = \beta^{-t}$$

oraz (ćwiczenie)

$$|\mathrm{rd}_t(m) - m| \leqslant \frac{1}{2}\beta^{-t},$$

zatem maksymalny błąd symetrycznego zaokrąglenia jest dwukrotnie mniejszy niż maksymalny błąd obcięcia. Z tego względu, podczas kodowania mantysy wykorzystujemy symetryczne zaokrąglanie. Numerycznym *odpowiednikiem* (lub *reprezentantem*) liczby rzeczywistej $x = s \cdot \beta^c \cdot m$ będzie zatem liczba

$$\mathrm{rd}(x) = s \cdot \beta^c \cdot \mathrm{rd}_t(m),$$

o ile oczywiście wartość cechy $c$ będzie mogła być dokładnie zakodowana, czyli będzie spełniać warunek (1.3). Błąd względny takiej reprezentacji to

$$\frac{|\mathrm{rd}(x) - x|}{|x|} = \frac{|s \cdot \beta^c \cdot \mathrm{rd}_t(m) - s \cdot \beta^c \cdot m|}{|s \cdot \beta^c \cdot m|} = \frac{|\mathrm{rd}_t(m) - m|}{|m|} \leqslant \frac{1}{2}\beta^{-t+1}.$$

Dla $\beta = 2$ otrzymujemy zatem

$$\frac{|\mathrm{rd}(x) - x|}{|x|} \leqslant 2^{-t},$$

co równoważnie możemy zapisać w bardzo wygodnej postaci

$$\mathrm{rd}(x) = x(1 + \varepsilon),$$

gdzie $|\varepsilon| \leqslant 2^{-t}$.

Niech $\mathcal{R}$ oznacza zbiór liczb reprezentowalnych w danym systemie, tj. przy ustalonych wartościach $\beta$, $d$ oraz $t$. Z powyższych rozważań wynika, że $\mathrm{rd}(x)$ jest takim elementem zbioru $\mathcal{R}$ (zakładamy oczywiście, że $\mathrm{rd}(x) \in \mathcal{R}$), dla którego

$$|x - \mathrm{rd}(x)| \leqslant |x - y|, \qquad \forall y \in \mathcal{R}.$$

**Przykład 1.5.** *Dla $t = 4$ mamy*

$$\begin{aligned} \mathrm{rd}(3,1415) &= 10^1 \cdot 0,3142 \quad (= 0,3142_{10}1), \\ \mathrm{rd}(0,031415) &= 10^{-1} \cdot 0,3142 \quad (= 0,3142_{10} - 1), \\ \mathrm{rd}(0,31415) &= 10^0 \cdot 0,3142 \quad (= 0,3142_{10}0). \end{aligned}$$

Zauważmy, że liczba $t$ cyfr mantysy decyduje o dokładności zmiennopozycyjnego przedstawienia liczb, a liczba $d - t - 1$ cyfr cechy określa zakres liczb reprezentowalnych.

Konsekwencje takiego sposobu zapisu liczb rzeczywistych w komputerach są następujące:

- w zbiorze $\mathcal{R}$ istnieją liczba największa

$$x^+_{\max} = \beta^{\beta^{d-t-1}-1}\left(1 - \beta^{-t}\right)$$

oraz najmniejsza liczba dodatnia

$$x^+_{\min} = \beta^{-\beta^{d-t-1}},$$

- zbiór $\mathcal{R}$ jest skończony, symetryczny względem zera oraz *nierównomierny*, tj. zagęszcza się w otoczeniu liczb $\pm x^{+}_{\min}$.

**Przykład 1.6.** *W przypadku* 64*-bitowego zapisu binarnego, tj. gdy* $\beta = 2$ *oraz* $d + 1 = 64$*, dla* $t = 52$ *mamy*

$$x^{+}_{\max} = 2^{2^{10}-1}\left(1 - 2^{-52}\right) \approx 9 \cdot 10^{307} \qquad \textit{oraz} \qquad x^{+}_{\min} = 2^{-2^{10}} \approx 6 \cdot 10^{-309}.$$

Niektóre języki programowania umożliwiają przeprowadzanie obliczeń z zadaną przez użytkownika dokładnością. W Pythonie odpowiada za to moduł `decimal`.[3] Moduł ten przechowuje liczby w formacie dziesiętnym, pozwala również na kontrolowane zaokrąglanie. Z kolei dostępny w Pythonie typ zmiennych `fraction` umożliwia reprezentację liczb wymiernych z nieskończoną precyzją, ograniczoną jedynie dostępną pamięcią; liczby wymierne przechowuje jako ułamek dwóch liczb całkowitych – taka reprezentacja nie ma błędów zaokrągleń typowych dla float i decimal.

## 1.3. Reprezentacja wektorów oraz macierzy

Niech $d = (d_1, \ldots, d_n) \in \mathbb{R}^n$. Każda współrzędna $d_i$ wektora $d$ jest liczbą rzeczywistą reprezentowaną przez przybliżenie

$$\mathrm{rd}\,(d_i) = d_i\,(1 + \varepsilon_i)\,,$$

gdzie $|\varepsilon_i| \leqslant 2^{-t}$. Wynika stąd, że numeryczną reprezentację wektora $d$ możemy zapisać jako

$$\mathrm{rd}\,(d) = (\mathrm{rd}\,(d_1), \ldots, \mathrm{rd}\,(d_n)) = (d_1\,(1 + \varepsilon_1), \ldots, d_n\,(1 + \varepsilon_n)) = d + \triangle d,$$

gdzie $\triangle d = (d_1\varepsilon_1, \ldots, d_n\varepsilon_n)$ oraz

$$\begin{aligned}\|\triangle d\| &\leqslant K_1 \cdot \|\triangle d\|_\infty = K_1 \cdot \max_{i=1}^{n} |d_i\varepsilon_i| \leqslant K_1 \cdot 2^{-t} \cdot \max_{i=1}^{n} |d_i| = \\ &= K_1 \cdot 2^{-t} \cdot \|d\|_\infty \leqslant K_2 \cdot 2^{-t} \cdot \|d\|\,;\end{aligned}$$

stałe $K_1$ oraz $K_2$ są konsekwencją równoważności norm w przestrzeni $\mathbb{R}^n$.

Podobnie, dla macierzy $A = [a_{ij}] \in \mathbb{R}^{m\times n}$ oraz jej elementów o reprezentacji $\mathrm{rd}\,(a_{ij}) = a_{ij}\,(1 + \varepsilon_{ij})\,$, $|\varepsilon_{ij}| \leqslant 2^{-t}$, możemy zapisać

$$\mathrm{rd}\,(A) = [\mathrm{rd}\,(a_{ij})] = [a_{ij}\,(1 + \varepsilon_{ij})] = A + \triangle A,$$

gdzie $\triangle A = [a_{ij}\varepsilon_{ij}] \in \mathbb{R}^{m\times n}$ oraz

$$\begin{aligned}\|\triangle A\| &\leqslant K_1 \cdot \|\triangle A\|_1 = K_1 \cdot \max_{j=1}^{n} \sum\nolimits_{i=1}^{m} |a_{ij}\varepsilon_{ij}| \leqslant K_1 \cdot 2^{-t} \cdot \max_{j=1}^{n} \sum\nolimits_{i=1}^{m} |a_{ij}| = \\ &= K_1 \cdot 2^{-t} \cdot \|A\|_1 \leqslant K_2 \cdot 2^{-t} \cdot \|A\|\,;\end{aligned}$$

występująca powyżej norma $\|\cdot\|$ oznacza dowolną normę macierzową (zob. dodatek A), stałe $K_1$ oraz $K_2$ są konsekwencją równoważności norm w przestrzeni $\mathbb{R}^{m\times n}$.

## 1.4. Arytmetyka zmiennopozycyjna fl

Symbolem $\mathrm{fl}\,(a * b)$ oznaczać będziemy wynik działania arytmetycznego $* \in \{+, -, \cdot, :\}$ wykonanego w arytmetyce numerycznej, tj. na liczbach $a, b$ równych ich numerycznym reprezentacjom $a = \mathrm{rd}\,(a)\,$, $b = \mathrm{rd}\,(b)$. Jeżeli $a * b = 0$ lub $x^{+}_{\min} \leqslant |a * b| \leqslant x^{+}_{\max}$, to działania te realizowane są w ten sposób, że

$$\mathrm{fl}\,(a * b) = (a * b)\,(1 + \varepsilon)\,,$$

gdzie $\varepsilon = \varepsilon\,(a, b, *)$ oraz $|\varepsilon| \leqslant 2^{-t}$. Można wykazać, że dla $* \in \{+, -, \cdot\}$

$$\varepsilon\,(a, b, *) = \varepsilon\,(b, a, *)$$

oraz dla $* \in \{\cdot, :\}$ i $k \in \mathbb{Z}$ (ale zachowującego powyższe ograniczenie dla $\left|a * 2^k\right|$)

$$\varepsilon(a, 2^k, *) = 0.$$

[3] https://docs.python.org/3/library/decimal.html

**Przykład 1.7.** *Mamy*

$$\begin{aligned}\mathrm{fl}\left((a+b)+c\right) &= \left((a+b)\left(1+\varepsilon_1\right)+c\right)\left(1+\varepsilon_2\right) = \left(a+b+c+(a+b)\,\varepsilon_1\right)\left(1+\varepsilon_2\right)\\ &= (a+b+c)\left(1+E_1\right),\end{aligned}$$

*gdzie*

$$E_1 = \varepsilon_2 + \frac{(a+b)\,\varepsilon_1}{a+b+c}\left(1+\varepsilon_2\right).$$

*Podobnie,*

$$\begin{aligned}\mathrm{fl}\left(a+(b+c)\right) &= \left(a+(b+c)\left(1+\varepsilon_3\right)\right)\left(1+\varepsilon_4\right) = \left(a+b+c+(b+c)\,\varepsilon_3\right)\left(1+\varepsilon_4\right)\\ &= (a+b+c)\left(1+E_2\right),\end{aligned}$$

*gdzie*

$$E_2 = \varepsilon_4 + \frac{(b+c)\,\varepsilon_3}{a+b+c}\left(1+\varepsilon_4\right).$$

*Widzimy zatem, że dla* $a+b\approx 0$ *błąd* $E_1\approx\varepsilon_2$ *jest mały, podczas gdy*

$$E_2 \approx \varepsilon_4 + \left(\frac{b}{c}+1\right)\varepsilon_3\left(1+\varepsilon_4\right)$$

*może być duży (np. dla* $b\gg c$*). Oznacza to, że na ogół*

$$\mathrm{fl}\left((a+b)+c\right) \neq \mathrm{fl}\left(a+(b+c)\right).$$

## 1.5. Upraszczanie oszacowań

**Przykład 1.8 (iloczyn liczb).** *Wyznaczymy w arytmetyce zmiennopozycyjnej* fl *wartość* $s$ *iloczynu* $n$ *liczb rzeczywistych* $a_1,\ldots,a_n$ *obliczoną naturalnym algorytmem*

$$\begin{aligned}&s_1 = a_1;\\ &\texttt{for } i = 1,\ldots,n-1 : s_{i+1} = s_i\cdot a_{i+1}.\end{aligned}$$

*Oczywiście* $s = s_n$*. Mamy*

$$\begin{aligned}\mathrm{fl}\left(s\right) &= \mathrm{fl}\left(\left(\ldots\left(\left(a_1\cdot a_2\right)\cdot a_3\right)\ldots\right)\cdot a_n\right) =\\ &= \left(\ldots\left(a_1\cdot a_2\left(1+\varepsilon_2\right)\cdot a_3\left(1+\varepsilon_3\right)\ldots\right)\cdot a_n\left(1+\varepsilon_n\right) =\\ &= s\cdot\prod_{i=2}^{n}\left(1+\varepsilon_i\right),\end{aligned}$$

*gdzie* $|\varepsilon_i|\leqslant 2^{-t}$*. Zauważmy, że*

$$\prod_{i=2}^{n}\left(1+\varepsilon_i\right) = 1+\varepsilon_2+\ldots+\varepsilon_n+\sum_{2\leqslant i<j\leqslant n}\varepsilon_i\varepsilon_j+\ldots+\varepsilon_2\cdot\ldots\cdot\varepsilon_n \approx 1+\varepsilon_2+\ldots+\varepsilon_n,$$

*gdyż wszystkie pominięte składniki są bardzo małe (zaniedbywalne) w stosunku do pozostawionych* $\varepsilon_2,\ldots,\varepsilon_n$*. Tym samym pokazaliśmy, że*

$$\mathrm{fl}\left(s\right)\approx s\cdot\left(1+E\right),$$

*gdzie* $E=\varepsilon_2+\ldots+\varepsilon_n$ *oraz* $|E|\leqslant(n-1)\,2^{-t}$*.*

Występującą w powyższym przykładzie równość przybliżoną $\approx$ musimy doprecyzować, aby skutecznie badać właściwości numeryczne (takie jak numeryczna stabilność, numeryczna poprawność) algorytmów obliczeniowych.

W tym celu oznaczmy przez $\mathcal{F}$ przestrzeń funkcji rzeczywistych określonych na przedziale $[1,\infty)$.

**Definicja 1.1 (równość w sensie jeden).**
*Funkcje $a, b \in \mathcal{F}$ są równe w sensie jeden, tj $a =_1 b$, jeżeli spełniony jest następujący warunek:*

$$\exists K > 0 \; \exists t_0 \geqslant 1 \; \forall t \geqslant t_0 : |a(t) - b(t)| \leqslant K \cdot 2^{-t} \cdot |a(t)| .$$

*Zapis $a \leqslant_1 b$ oznaczać będzie, że istnieje funkcja $c \in \mathcal{F}$, taka że $a \leqslant c$ oraz $c =_1 b$.*

**Przykład 1.9.** *Łatwo wykazać, że*

$$5 \cdot 2^{-t} + 3 \cdot 2^{-2t} =_1 5 \cdot 2^{-t}$$

*ale*

$$3 \cdot 2^{-2t} \neq_1 0.$$

**Stwierdzenie 1.1.** *Relacja równości w sensie jeden jest relacją równoważności w przestrzeni $\mathcal{F}$.*

**Dowód**: Zostanie przeprowadzony na ćwiczeniach. ∎

**Stwierdzenie 1.2.** *Niech $|E_i| \leqslant K_i \cdot 2^{-t}$ dla $i = 1, \ldots, n$. Wówczas:*

**a)** *jeżeli* $1 + E = \prod_{i=1}^{n} (1 + E_i)$, *to* $|E| \leqslant_1 (K_1 + \ldots + K_n) \cdot 2^{-t}$;

**b)** *jeżeli* $1 + E = \dfrac{1 + E_1}{1 + E_2}$, *to* $|E| \leqslant_1 (K_1 + K_2) \cdot 2^{-t}$;

**c)** *jeżeli* $1 + E = \sqrt{1 + E_1}$, *to* $|E| \leqslant_1 \dfrac{1}{2} K_1 \cdot 2^{-t}$.

**Dowód**: Zostanie przeprowadzony na ćwiczeniach. ∎

**Przykład 1.10 (kontynuacja przykładu 1.8).** *Na podstawie stwierdzenia 1.2 możemy zapisać*

$$\mathrm{fl}(s) = s \cdot (1 + E),$$

*gdzie* $\prod_{i=2}^{n} (1 + \varepsilon_i) = 1 + E$ *oraz* $|E| \leqslant_1 (n-1) 2^{-t}$, *eliminując z zapisu równość przybliżoną* $\approx$. *Analizując błąd względny uzyskanej wartości otrzymujemy dla $s \neq 0$:*

$$\frac{|\mathrm{fl}(s) - s|}{|s|} = \frac{|s \cdot (1 + E) - s|}{|s|} = |E| \leqslant_1 (n-1) 2^{-t},$$

*co daje mały błąd względny, niezależnie od wartości czynników.*

**Przykład 1.11 (suma liczb).** *Wyznaczymy w arytmetyce zmiennopozycyjnej* fl *wartość $s$ sumy $n$ liczb rzeczywistych $a_1, \ldots, a_n$ obliczoną naturalnym algorytmem*

$$\begin{array}{l} s_1 = a_1; \\ \texttt{for } i = 1, \ldots, n-1 : s_{i+1} = s_i + a_{i+1}. \end{array}$$

*Oczywiście $s = s_n$. Mamy*

$$\begin{aligned} \mathrm{fl}(s) &= \mathrm{fl}((\ldots((a_1 + a_2) + a_3) + \ldots) + a_n) = \\ &= ((\ldots(((a_1 + a_2)(1 + \varepsilon_2) + a_3)(1 + \varepsilon_3) + \ldots) + a_n)(1 + \varepsilon_n) = \\ &= \sum_{i=1}^{n} a_i (1 + E_i), \end{aligned}$$

*gdzie*

$$1 + E_1 = (1 + \varepsilon_2) \cdot \ldots \cdot (1 + \varepsilon_n)$$

*oraz*

$$1 + E_i = (1 + \varepsilon_i) \cdot \ldots \cdot (1 + \varepsilon_n)$$

*dla* $i = 2, \ldots, n$. *Zakładając, że* $|\varepsilon_i| \leqslant 2^{-t}$, *na podstawie stwierdzenia 1.2, otrzymujemy oszacowania*

$$|E_1| \leqslant_1 (n-1)\, 2^{-t}$$

*oraz*

$$|E_i| \leqslant_1 (n-i+1)\, 2^{-t}$$

*dla* $i = 2, \ldots, n$. *Uzyskana w arytmetyce* fl *wartość sumy* $n$ *liczb jest zatem dokładną wartością sumy nieco zaburzonych liczb* $\tilde{a}_i = a_i\,(1+E_i)$. *W niedalekiej przyszłości algorytm o takiej własności będziemy nazywać numerycznie poprawnym.*

*Analizując błąd względny uzyskanej wartości* $\mathrm{fl}\,(s)$ *otrzymujemy dla* $s \neq 0$

$$\frac{|\mathrm{fl}\,(s)-s|}{|s|} = \frac{\left|\sum\limits_{i=1}^{n} a_i\,(1+E_i) - \sum\limits_{i=1}^{n} a_i\right|}{\left|\sum\limits_{i=1}^{n} a_i\right|} \leqslant \frac{\sum\limits_{i=1}^{n} |a_i E_i|}{\left|\sum\limits_{i=1}^{n} a_i\right|} \leqslant_1 \frac{\sum\limits_{i=1}^{n} |a_i|}{\left|\sum\limits_{i=1}^{n} a_i\right|}\,(n-1)\,2^{-t}.$$

*Oszacowanie to pokazuje, że sumowanie liczb tego samego znaku nie będzie stwarzało trudności – zawsze otrzymamy mały błąd względny, jednakże w przypadku ogólnym pewne trudności mogą się pojawić (np. w sytuacji, gdy* $|\sum a_i| \ll \sum |a_i|$).

**Przykład 1.12 (iloczyn skalarny).** *Wyznaczymy w arytmetyce zmiennopozycyjnej* fl *wartość* $s$,

$$s = \sum_{i=1}^{n} a_i b_i,$$

*naturalnego iloczynu skalarnego wektorów* $(a_1, \ldots, a_n), (b_1, \ldots, b_n) \in \mathbb{R}^n$ *obliczoną naturalnym algorytmem*

$$\begin{array}{l} s_1 = a_1 b_1; \\ \texttt{for}\ i = 1, \ldots, n-1 : s_{i+1} = s_i + a_{i+1} b_{i+1}. \end{array}$$

*Oczywiście* $s = s_n$. *Mamy*

$$\begin{aligned} \mathrm{fl}\,(s) &= \mathrm{fl}\,((\ldots((a_1b_1 + a_2b_2) + a_3b_3) + \ldots) + a_nb_n) = \\ &= \Big(\big(((a_1b_1(1+\delta_1) + a_2b_2(1+\delta_2))(1+\varepsilon_2) + a_3b_3(1+\delta_3))\,(1+\varepsilon_3) + \ldots \\ &\quad \ldots\big)(1+\epsilon_{n-1}) + a_nb_n(1+\delta_n)\Big)\,(1+\varepsilon_n) = \sum_{i=1}^{n} a_ib_i\,(1+E_i)\,, \end{aligned}$$

*gdzie*

$$1 + E_1 = (1+\delta_1)\,(1+\varepsilon_2)\cdot\ldots\cdot(1+\varepsilon_n)$$

*oraz*

$$1 + E_i = (1+\delta_i)\,(1+\varepsilon_i)\cdot\ldots\cdot(1+\varepsilon_n)$$

*dla* $i = 2, \ldots, n$. *Zakładając, że* $|\delta_i| \leqslant 2^{-t}$ *oraz* $|\varepsilon_i| \leqslant 2^{-t}$, *na podstawie stwierdzenia 1.2, otrzymujemy oszacowania*

$$|E_1| \leqslant_1 n2^{-t}$$

*oraz*

$$|E_i| \leqslant_1 (n-i+2)\, 2^{-t}$$

*dla* $i = 2, \ldots, n$. *Analizując błąd względny uzyskanej wartości* $\mathrm{fl}\,(s)$ *otrzymujemy dla* $s \neq 0$

$$\frac{|\mathrm{fl}\,(s)-s|}{|s|} = \frac{\left|\sum\limits_{i=1}^{n} a_ib_i\,(1+E_i) - \sum\limits_{i=1}^{n} a_ib_i\right|}{\left|\sum\limits_{i=1}^{n} a_ib_i\right|} \leqslant \frac{\sum\limits_{i=1}^{n} |a_ib_iE_i|}{\left|\sum\limits_{i=1}^{n} a_ib_i\right|} \leqslant_1 \frac{\sum\limits_{i=1}^{n} |a_ib_i|}{\left|\sum\limits_{i=1}^{n} a_ib_i\right|} n2^{-t}.$$

## Rozdział 2

# Uwarunkowanie zadania obliczeniowego, kryteria akceptowalności algorytmów

### 2.1. Uwarunkowanie zadania obliczeniowego

Zadaniem obliczeniowym będzie dla nas funkcja $\varphi : D \to \mathbb{R}^n$, która dla danych $d \in D \subset \mathbb{R}^k$ zwraca wynik $w = \varphi(d)$. Rozwiązując zadanie obliczeniowe, tj. wyznaczając numerycznie wartości funkcji $\varphi$, musimy zdawać sobie sprawę z tego, że najczęściej nie dysponujemy dokładnymi danymi, lecz jedynie ich przybliżonymi wartościami (np. zamiast $d$ dysponujemy $\mathrm{rd}(d)$). Okazuje się, że nawet bardzo niewielkie względne zmiany danych mogą powodować istotne względne zmiany wyniku.

**Przykład 2.1 (perfidny wielomian Wilkinsona).** *Rozważmy tzw. wielomian Wilkinsona*

$$\begin{aligned} w(s) &= s^{20} + a_{19}s^{19} + \ldots + a_1 s + a_0 = \\ &= \prod_{k=1}^{20} (s-k) = s^{20} - 210s^{19} + \ldots + 20! \end{aligned}$$

*oraz niech*

$$\begin{aligned} w^*(s) &= s^{20} + a^*_{19}s^{19} + \ldots + a^*_1 s + a^*_0 = \\ &= w(s) - 2^{-23}s^{19} = s^{20} - \left(210 + 2^{-23}\right) s^{19} + \ldots + 20! \end{aligned}$$

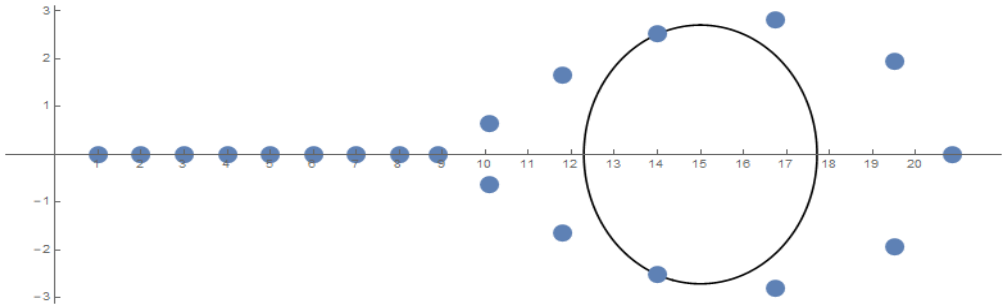
*Niewielka względna zmiana wartości współczynnika $a_{19}$ wielomianu $w$*

$$\frac{|a^*_{19} - a_{19}|}{|a_{19}|} = \frac{2^{-23}}{210} \approx 6 \cdot 10^{-10}$$

*powoduje bardzo duże względne zmiany wartości jego pierwiastków (zob. rysunek 2.1): pierwiastek $s^*_{15} \approx 13,992 + 2,519i$ wielomianu $w^*$ leżący najbliżej pierwiastka $s_{15} = 15$ wielomianu $w$ spełnia warunek*

$$|s_{15} - s^*_{15}| > 2,71.$$

Przykład 2.1 pokazuje, że niewielka względna zmiana wartości współczynników wielomianu może powodować duże względne zmiany wartości jego pierwiastków. Zadania takie będziemy nazywać zadaniami źle uwarunkowanymi.



Rysunek 2.1. Pierwiastki zaburzonego wielomianu Wilkinsona (granatowe kropki) oraz okrąg o środku w punkcie $(15, 0)$ i promieniu $r = 2,71$.

**Definicja 2.1 (uwarunkowanie zadania).**
*Uwarunkowanie zadania obliczeniowego* $\varphi : D \ni d \to \varphi(d) \in \mathbb{R}^n$ *to wrażliwość wyniku na małe względne zaburzenia danych. Miarą uwarunkowania zadania jest tzw. wskaźnik uwarunkowania* $\operatorname{cond}(\varphi, d)$,

$$\frac{\|\varphi(d + \triangle d) - \varphi(d)\|}{\|\varphi(d)\|} \leqslant \operatorname{cond}(\varphi, d) \frac{\|\triangle d\|}{\|d\|};$$

*im większa wartość wskaźnika uwarunkowania, tym zadanie jest gorzej uwarunkowane.*

Warto podkreślić, że uwarunkowanie to cecha samego zadania: zależy ono od zadania oraz od danych (dla pewych danych to samo zadanie może być dobrze uwarunkowane, a dla innych źle uwarunkowane), nie zależy natomiast od algorytmu (sposobu) jego rozwiązywania.

**Przykład 2.2 (kontynuacja przykładu 2.1).** *Niech* $s_{15} : \mathbb{R} \ni a_{19} \to s_{15}(a_{19}) \in \mathbb{C}$ *będzie zadaniem polegającym na wyznaczeniu pierwiastka* $s_{15}$ *wielomianu* $w$. *Zadanie to potraktujemy jako funkcję zależną jedynie od współczynnika* $a_{19}$ *(pozostałe współczynniki są ustalone). Możemy zatem zapisać*

$$\frac{\|s_{15}(a_{19} + \triangle a_{19}) - s_{15}(a_{19})\|}{\|s_{15}(a_{19})\|} \leqslant \operatorname{cond}(s_{15}, a_{19}) \frac{\|\triangle a_{19}\|}{\|a_{19}\|}.$$

*Uwzględniając wartości z przykładu 2.1 otrzymujemy oszacowanie wskaźnika uwarunkowania*

$$\operatorname{cond}(s_{15}, a_{19}) \geqslant \frac{\|s_{15}(a_{19} + \triangle a_{19}) - s_{15}(a_{19})\|}{\|s_{15}(a_{19})\|} \frac{\|a_{19}\|}{\|\triangle a_{19}\|} > 3 \cdot 10^8.$$

**Przykład 2.3 (sumowanie liczb).** *Zbadamy uwarunkowanie zadania obliczenia sumy* $n$ *liczb rzeczywistych* $a_1, \ldots, a_n$. *Niech*

$$s = \sum_{i=1}^{n} a_i \qquad \text{oraz} \qquad \bar{s} = \sum_{i=1}^{n} a_i (1 + \varepsilon_i),$$

*gdzie* $|\varepsilon_i| \leqslant 2^{-t}$ *to zaburzenie* $a_i$ *dla* $i = 1, \ldots, n$. *Wówczas, dla* $s \neq 0$,

$$\frac{|\bar{s} - s|}{|s|} = \frac{\left|\sum_{i=1}^{n} a_i (1 + \varepsilon_i) - \sum_{i=1}^{n} a_i\right|}{\left|\sum_{i=1}^{n} a_i\right|} = \frac{\left|\sum_{i=1}^{n} a_i \varepsilon_i\right|}{\left|\sum_{i=1}^{n} a_i\right|} \leqslant \frac{\sum_{i=1}^{n} |a_i \varepsilon_i|}{\left|\sum_{i=1}^{n} a_i\right|} \leqslant \frac{\sum_{i=1}^{n} |a_i|}{\left|\sum_{i=1}^{n} a_i\right|} \cdot 2^{-t}.$$

*Wskaźnik uwarunkowania tego zadania to zatem*

$$\operatorname{cond}(\varphi, a_1, \ldots, a_n) = \frac{\sum_{i=1}^{n} |a_i|}{\left|\sum_{i=1}^{n} a_i\right|}.$$

*Oznacza to, że zadanie sumowania liczb o tych samych znakach jest bardzo dobrze uwarunkowane; wówczas* $\operatorname{cond}(\varphi, a_1, \ldots, a_n) = 1$. *Z kolei, dla danych, których suma jest bliska zero, wskaźnik uwarunkowania może być bardzo duży.*

**Przykład 2.4 (rozwiązywanie układu równań liniowych).** *Dla zadanej macierzy nieosobliwej* $A \in \mathbb{R}^{n\times n}$ *oraz wektora* $b \in \mathbb{R}^n$ *rozważmy zadanie wyznaczenia rozwiązania* $w$ *równania*

$$Aw = b. \tag{2.1}$$

*Razem z równaniem* (2.1) *musimy zatem rozważać równanie zaburzone*

$$(A + \triangle A)(w + \triangle w) = b + \triangle b, \tag{2.2}$$

*przy czym dopuszczamy jedynie takie zaburzenia* $\triangle A$ *macierzy* $A$, *dla których równanie* (2.2) *jakościowo nie różni się od równania* (2.1), *tj. nadal posiada dokładnie jedno rozwiązanie. Aby to zapewnić, wystarczy ograniczyć się do zaburzeń spełniających warunek*

$$\|A^{-1}\|\,\|\triangle A\| < 1, \tag{2.3}$$

*gdzie* $\|\cdot\|$ *to norma macierzowa indukowana przez dowolną normę wektorową. Istotnie, mamy wówczas*

$$\rho(A^{-1} \triangle A) \leqslant \|A^{-1} \triangle A\| \leqslant \|A^{-1}\|\,\|\triangle A\| < 1$$

*co oznacza, że wśród wartości własnych macierzy*

$$A + \triangle A = A\left(I_n + A^{-1} \triangle A\right)$$

*nie występuje zero; macierz* $A + \triangle A$ *jest zatem nieosobliwa.*

*Odejmując stronami równania* (2.1) *oraz* (2.2) *otrzymujemy kolejno*

$$A \triangle w + \triangle A w + \triangle A \triangle w = \triangle b$$

$$A \triangle w = -\triangle A w - \triangle A \triangle w + \triangle b$$

*oraz*

$$\triangle w = -A^{-1} \triangle A w - A^{-1} \triangle A \triangle w + A^{-1} \triangle b.$$

*Stąd*

$$\|\triangle w\| \leqslant \|A^{-1}\|\,\|\triangle A\|\,\|w\| + \|A^{-1}\|\|\triangle A\|\,\|\triangle w\| + \|A^{-1}\|\,\|\triangle b\|$$

*czyli*

$$(1 - \|A^{-1}\|\|\triangle A\|)\|\triangle w\| \leqslant \|A^{-1}\|\,\|\triangle A\|\,\|w\| + \|A^{-1}\|\,\|\triangle b\|\,.$$

*Stąd, na podstawie warunku* (2.3),

$$\|\triangle w\| \leqslant \frac{1}{1 - \|A^{-1}\|\|\triangle A\|}\left(\|A^{-1}\|\,\|\triangle A\|\,\|w\| + \|A^{-1}\|\,\|\triangle b\|\right)$$

*i ostatecznie*

$$\frac{\|\triangle w\|}{\|w\|} \leqslant \frac{1}{1 - \|A^{-1}\|\|\triangle A\|}\left(\|A^{-1}\|\,\|A\|\,\frac{\|\triangle A\|}{\|A\|} + \|A^{-1}\|\frac{\|b\|}{\|w\|}\frac{\|\triangle b\|}{\|b\|}\right) \overset{\|b\|\leqslant\|A\|\|w\|}{\leqslant}$$
$$\leqslant \frac{\|A^{-1}\|\,\|A\|}{1 - \|A^{-1}\|\|\triangle A\|}\left(\frac{\|\triangle A\|}{\|A\|} + \frac{\|\triangle b\|}{\|b\|}\right) = \frac{\operatorname{cond}(A)}{1 - \operatorname{cond}(A)\frac{\|\triangle A\|}{\|A\|}}\left(\frac{\|\triangle A\|}{\|A\|} + \frac{\|\triangle b\|}{\|b\|}\right),$$

*gdzie*

$$\operatorname{cond}(A) = \|A^{-1}\|\,\|A\|$$

*to tzw. wskaźnik uwarunkowania macierzy* $A$. *Wielkości* $\|\triangle A\| / \|A\|$ *oraz* $\|\triangle b\| / \|b\|$ *to względne zaburzenia danych* (*odpowiednio macierzy oraz wektora prawej strony*) *zatem wskaźnik* $\operatorname{cond}(A)$ *determinuje uwarunkowanie rozważanego zadania.*

*Z otrzymanych oszacowań wynika, że dla zadania polegającego na rozwiązaniu układu $Aw = b$, w którym dopuszczamy zaburzenia jedynie wektora $b$ (tj. $\triangle A = 0$) mamy*

$$\frac{\|\triangle w\|}{\|w\|} \leqslant \operatorname{cond}(A) \frac{\|\triangle b\|}{\|b\|}.$$

*To oznacza, że* $\operatorname{cond}(A)$ *jest wskaźnikiem uwarunkowania tego zadania. Jak łatwo się przekonać, wskaźnik ten może być dowolnie duży.*[1]

**Przykład 2.5.** *Rozważmy układ równań liniowych $Aw = b$, gdzie*

$$A = \begin{pmatrix} 1 & 2 \\ 3 & 4 \end{pmatrix}, \qquad b = \begin{pmatrix} 3 \\ 7 \end{pmatrix}.$$

*Jego rozwiązaniem jest*

$$w = \begin{pmatrix} 1 \\ 1 \end{pmatrix}.$$

*Rozważmy teraz układ równań $A(w + \triangle w) = b + \triangle b$, gdzie*

$$b + \triangle b = \begin{pmatrix} 3,0001 \\ 7,0001 \end{pmatrix}.$$

*Proste obliczenia pokazują, że*

$$\frac{\| \triangle b\|_2}{\|b\|_2} \approx 1,8570 \cdot 10^{-5}$$

*oraz*

$$\operatorname{cond}(A) = \|A^{-1}\|_2 \|A\|_2 \approx 14,9330$$

*co oznacza, że niewielkie zaburzenie wektora $b$ nie wpłynie istotnie na wynik. Istotnie, rozwiązanie układu zaburzonego to*

$$w + \triangle w = \begin{pmatrix} 0,9999 \\ 1,0001 \end{pmatrix}, \qquad \frac{\| \triangle w\|_2}{\|w\|_2} = 10^{-4}.$$

**Przykład 2.6.** *Rozważmy układ równań $Aw = b$, gdzie*

$$A = \begin{pmatrix} 1 & 2 & 1 \\ 2 & 4,0001 & 2,002 \\ 1 & 2,002 & 2,004 \end{pmatrix}, \qquad b = \begin{pmatrix} 4 \\ 8,0021 \\ 5,006 \end{pmatrix}.$$

*Jego dokładne rozwiązanie to*

$$w = \begin{pmatrix} 1 \\ 1 \\ 1 \end{pmatrix}.$$

*Z kolei, rozwiązaniem układu zaburzonego $A(w + \triangle w) = b + \triangle b$, gdzie*

$$b + \triangle b = \begin{pmatrix} 4 \\ 8,0020 \\ 5,0061 \end{pmatrix}, \qquad \frac{\|\triangle b\|_2}{\|b\|_2} \approx 1,3795 \cdot 10^{-5}$$

*jest*

$$w + \triangle w = \begin{pmatrix} 3,0850 \\ -0,0436 \\ 1,0022 \end{pmatrix};$$

*pomimo niewielkiego względnego zaburzenia danych, rozwiązanie to jest istotnie różne od rozwiązania układu niezaburzonego. Dzieje się tak, gdyż w rozważanym przypadku wskaźnik uwarunkowania macierzy $A$ układu jest bardzo duży:*

$$\operatorname{cond}(A) = \|A^{-1}\|_2 \|A\|_2 \approx 3,2210 \cdot 10^5.$$

[1] Dla $\alpha > 1$ oraz dla macierzy diagonalnej $A = \operatorname{diag}(\alpha, 1, \ldots, 1)$ mamy $\|A\|_1 \left\|A^{-1}\right\|_1 = \alpha$.

## 2.2. Kryteria akceptowalności algorytmów

Rozwiązując numerycznie zadanie obliczeniowe musimy zdawać sobie sprawę z następujących problemów:

- dane, którymi dysponujemy na ogół nie są dokładne (błędy pomiaru, błędy reprezentacji),
- błędy reprezentacji danych przenoszą się, niezależnie od algorytmu, na błędy wyniku (wiemy już, że wpływ ten jest tym większy, im zadanie jest gorzej uwarunkowane),
- sam algorytm, nawet najlepszy, wytwarza błędy (jest to cecha operacji realizowanych w arytmetyce fl: jak wiemy, zwykle $\mathrm{fl}(a * b) \neq a * b$ nawet dla dokładnych danych $a = \mathrm{rd}(a)$, $b = \mathrm{rd}(b)$).

Mając to na uwadze zasadne wydaje się pytanie, w jaki sposób oceniać jakość algorytmów?

### 2.2.1. Algorytmy numerycznie poprawne

Podana poniżej niezbyt precyzyjna definicja algorytmu numerycznie poprawnego (tzw. formuła Kahana) dobrze oddaje ideę numerycznej poprawności; jest ona wystarczająca na potrzeby tego wykładu (formalna definicja algorytmu numerycznie poprawnego będzie wprowadzona na wykładzie z *analizy numerycznej* w ramach studiów drugiego stopnia).

**Definicja 2.2 (numeryczna poprawność).** *Algorytm jest numerycznie poprawny, jeżeli jego wynik obliczony w arytmetyce* fl *jest nieco zaburzonym wynikiem dokładnym zadania o nieco zaburzonych danych. Przez „nieco zaburzenie" rozumiemy zaburzenie na poziomie błędu reprezentacji.*

Numeryczna poprawność algorytmu oznacza, że algorytm działa tak, jakby wszystkie obliczenia były wykonane dokładnie, a tylko dane i wyniki miały niedokładności reprezentacji liczbowej. Jest to cecha pożądana, gdyż świadczy o tym, że algorytm jest odporny na błędy reprezentacji danych.

**Przykład 2.7.** *Zadanie: dla danych* $a, b, c \in \mathbb{R}$, $(a+1)\,b \neq 0$ *obliczyć wartość*

$$w(a,b,c) = \frac{ac}{ab+b}.$$

*W tym celu zaproponujemy algorytm polegający na wyliczeniu kolejno*

$$ac,\ ab \to ab+b \to \frac{ac}{ab+b}.$$

*Przeanalizujmy realizację tego algorytmu w arytmetyce* fl*. Mamy kolejno*

$$\mathrm{fl}(w(a,b,c)) = \frac{ac(1+\varepsilon_1)}{(ab(1+\varepsilon_2)+b)(1+\varepsilon_3)}(1+\varepsilon_4),$$

*gdzie* $|\varepsilon_i| \leqslant 2^{-t}$*, dla* $i = 1,2,3,4$*. Stąd*

$$\mathrm{fl}(w(a,b,c)) = \frac{a(1+\varepsilon_2)c}{a(1+\varepsilon_2)b+b}\frac{(1+\varepsilon_1)(1+\varepsilon_4)}{(1+\varepsilon_2)(1+\varepsilon_3)} = w(a(1+\varepsilon_2), b, c(1+\varepsilon_c)),$$

*gdzie*

$$1+\varepsilon_c = \frac{(1+\varepsilon_1)(1+\varepsilon_4)}{(1+\varepsilon_2)(1+\varepsilon_3)}$$

*oraz, na podstawie stwierdzenia 1.2,* $|\varepsilon_c| \leqslant_1 4 \cdot 2^{-t}$*. Oznacza to, że wartość* $w(a,b,c)$ *obliczona w arytmetyce* fl *zaproponowanym algorytmem jest dokładną wartością zadania o nieco zaburzonych danych. To, na podstawie formuły Kahana, oznacza numeryczną poprawność tego algorytmu.*

**Przykład 2.8 (algorytm Hornera).** *Zadanie: dla danych* $s, a_0, \ldots, a_n \in \mathbb{R}$ *obliczyć wartość*

$$w(s, a_0, \ldots, a_n) = a_n s^n + a_{n-1}s^{n-1} + \ldots + a_0.$$

*Z równości*

$$\begin{aligned} w = w(s, a_0, \ldots, a_n) &= a_n s^n + a_{n-1}s^{n-1} + \ldots + a_0 = \\ &= a_0 + s \cdot (a_1 + \ldots + s \cdot (a_{n-1} + s \cdot a_n)) \end{aligned}$$

*wynika następujący algorytm*

$$\begin{aligned} &s_n = a_n;\\ &\texttt{for } i = n-1,\ldots,0 : s_i = s_{i+1}\cdot s + a_i. \end{aligned}$$

*Oczywiście* $w(s,a_0,\ldots,a_n) = s_0$. *Przeanalizujemy realizację zaproponowanego algorytmu w arytmetyce* fl*:*

$$\begin{aligned} \mathrm{fl}(w) =& (((a_n s(1+\varepsilon_n) + a_{n-1})(1+\delta_{n-1}) s(1+\varepsilon_{n-1}) + \ldots\\ &\ldots + a_1)(1+\delta_1) s(1+\varepsilon_1) + a_0)(1+\delta_0) =\\ =& a_n s^n (1+\varepsilon_n)(1+\varepsilon_{n-1})\cdots(1+\varepsilon_1)(1+\delta_{n-1})\cdots(1+\delta_0) +\\ &+ a_{n-1}s^{n-1}(1+\varepsilon_{n-1})\cdots(1+\varepsilon_1)(1+\delta_{n-1})\cdots(1+\delta_0)+\cdots\\ &\cdots + a_k s^k (1+\varepsilon_k)\cdots(1+\varepsilon_1)(1+\delta_k)\cdots(1+\delta_0)+\cdots\\ &\cdots + a_0(1+\delta_0) = \sum_{k=0}^{n} a_k (1+E_k) s^k, \end{aligned}$$

*gdzie, na podstawie stwierdzenia 1.2,*

$$|E_n| \leqslant_1 2n\cdot 2^{-t}$$

*oraz*

$$|E_k| \leqslant_1 (2k+1)\cdot 2^{-t}$$

*dla* $k = 0,\ldots,n-1$. *Uzyskany wynik jest zatem rozwiązaniem dokładnym zadania o nieco zaburzonych danych. To oznacza numeryczną poprawność badanego algorytmu. Zanotujmy również, że koszt wyliczenia wartości* $w(s,a_0,\ldots,a_n)$ *to w najgorszym przypadku* $2n$ *działań arytmetycznych (po* $n$ *dodawań oraz mnożeń). Można udowodnić, że nie istnieje algorytm wyliczania wartości wielomianu zapisanego w postaci naturalnej wymagający mniejszej liczby operacji arytmetycznych (zob. A. Borodin, Horner's rule is uniquely optimal, in Theory of Machines and Computations, New York, Academic Press 1971).*

### 2.2.2. Algorytmy numerycznie stabilne

Rozważmy zadanie obliczeniowe

$$\varphi : D \ni d \to \varphi(d) \in \mathbb{R}^n.$$

Niech $w = \varphi(d)$ oraz $\widehat{w} = \varphi(d+\triangle d)$, gdzie $\|\triangle d\| \leqslant K_d 2^{-t}\|d\|$ oraz niech $\widetilde{w}$ oznacza nieco zaburzoną wartość $\varphi(d+\triangle d)$, tj.

$$\|\widetilde{w} - \widehat{w}\| \leqslant K_w 2^{-t}\|\widehat{w}\|.$$

Mając na uwadze ograniczoność błędów reprezentacji (danych oraz wyniku) oraz wpływ tych błędów na dokładność wyniku zastanowimy się teraz jak duży może być błąd $\|\widetilde{w} - w\|$. Mamy

$$\begin{aligned} \|\widetilde{w} - w\| &\leqslant \|\widetilde{w} - \widehat{w}\| + \|\widehat{w} - w\| \leqslant K_w 2^{-t}\|\widehat{w}\| + K_d 2^{-t}\|w\|\,\mathrm{cond}(\varphi,d) \leqslant\\ &\leqslant K_w 2^{-t}(\|\widehat{w} - w\| + \|w\|) + K_d 2^{-t}\|w\|\,\mathrm{cond}(\varphi,d) \leqslant\\ &\leqslant K_w 2^{-t}\left(K_d 2^{-t}\|w\|\,\mathrm{cond}(\varphi,d) + \|w\|\right) + K_d 2^{-t}\|w\|\,\mathrm{cond}(\varphi,d) =_1\\ &=_1 K_w 2^{-t}\|w\| + K_d 2^{-t}\|w\|\,\mathrm{cond}(\varphi,d) \leqslant\\ &\leqslant \max\{K_w, K_d\}\, 2^{-t}\|w\|(1+\mathrm{cond}(\varphi,d)). \end{aligned}$$

**Definicja 2.3 (numeryczna stabilność).** *Algorytm* $A(\varphi(d))$ *obliczenia wartości* $\varphi(d)$ *w klasie danych* $D$ *nazywamy numerycznie stabilnym, jeżeli spełniony jest następujący warunek:*
$\exists K > 0\ \exists t_0 \geqslant 1\ \forall d \in D\ \forall t \geqslant t_0$

$$\|\mathrm{fl}(A(\varphi(d))) - \varphi(d)\| \leqslant K 2^{-t}(1+\mathrm{cond}(\varphi,d))\|\varphi(d)\|.$$

Numeryczna stabilność algorytmu to jego zdolność do kontrolowania narastania błędów zaokrągleń w trakcie całego procesu obliczeniowego. Błąd w algorytmie numerycznie stabilnym jest na poziomie nieuniknionego błędu rozwiązania wynikającego z przybliżonej reprezentacji danych (składnik $K2^{-t}\operatorname{cond}(\varphi, d)\,\|\varphi(d)\|$ to maksymalna wielkość błędu przeniesionego z danych na wynik) oraz wyniku (składnik $K2^{-t}\,\|\varphi(d)\|$ to maksymalny błąd reprezentacji wyniku). Rezygnacja z numerycznej stabilności algorytmu oznaczałaby rezygnację z ograniczoności strat dokładności w stosunku do tego, co możemy uzyskać w stosowanej arytmetyce.

Z powyższej definicji wynika również, że jeżeli zadanie jest dobrze uwarunkowane (tj. wartość $\operatorname{cond}(\varphi, d)$ jest mała), to rozwiązując to zadanie algorytmem numerycznie stabilnym otrzymujemy rozwiązanie o niewielkim błędzie względnym. Każdy algorytm numerycznie poprawny jest zatem numerycznie stabilny (implikacja odwrotna nie zachodzi, ale na nietrywialny przykład trzeba będzie jeszcze poczekać; trywialnego przykładu nie ma).

## Rozdział 3

# Interpolacja wielomianowa Lagrange'a

Problem interpolacji pojawia się bardzo często w tych zagadnieniach, w których chcemy wykorzystywać funkcję dysponując jedynie informacją o jej wartościach w skończonej liczbie punktów.

### 3.1. Sformułowanie problemu, istnienie i jedyność rozwiązania

**Sformułowanie problemu**

Dla zadanej funkcji $f : [a,b] \to \mathbb{R}$ oraz węzłów $x_0, \ldots, x_n \in [a,b]$ poszukujemy wielomianu $w$ (możliwie niskiego stopnia) spełniającego warunki interpolacji

$$w(x_i) = f(x_i), \qquad \text{dla } i = 0, \ldots, n.$$

Wielomian ten będziemy nazywać wielomianem interpolacyjnym dla funkcji $f$.

**Istnienie rozwiązania**

Dla parami różnych węzłów $x_0, \ldots, x_n$ (tj. $x_i \neq x_j$ dla $i \neq j$) rozważmy funkcje $\phi_0, \ldots, \phi_n$ postaci

$$\phi_i(x) = \prod_{j \neq i, j=0}^{n} \frac{x - x_j}{x_i - x_j}.$$

**Stwierdzenie 3.1 (własności funkcji $\phi_0, \ldots, \phi_n$).** *Przy powyższych założeniach oraz oznaczeniach mamy*

**1.** $\phi_0, \ldots, \phi_n \in \pi_n$ *oraz* $\deg \phi_i = n$.
**2.** $\phi_i(x_k) = \delta_{ik}$, *tj.*

$$\phi_i(x_k) = \begin{cases} 1 & dla\ i = k \\ 0 & dla\ i \neq k \end{cases}.$$

**3.** *Funkcje $\phi_0, \ldots, \phi_n$ są liniowo niezależne.*

**Dowód**: W ramach prostego ćwiczenia wykażemy jedynie punkt 3 (uzasadnienia pozstałych dwóch punktów są zupełnie elementarne). Należy wykazać, że jedynym rozwiązaniem równania

$$\sum_{i=0}^{n} \alpha_i \phi_i = 0 \tag{3.1}$$

jest rozwiązanie zerowe $\alpha_0 = \ldots = \alpha_n = 0$. Z warunku (3.1) wynika, że dla wszystkich $x \in \mathbb{R}$

$$\sum_{i=0}^{n} \alpha_i \phi_i(x) = 0.$$

Przyjmując $x = x_k$, na podstawie punktu 2. wynika stąd, że

$$0 = \sum_{i=0}^{n} \alpha_i \phi_i (x_k) = \alpha_k \phi_k (x_k) = \alpha_k.$$

Wobec dowolności $k$ otrzymujemy tezę. ∎

Z powyższych własności funkcji $\phi_0, \ldots, \phi_n$ wynika, że stanowią one bazę przestrzeni wielomianów $\pi_n$. Łatwo stwierdzić, że rozpięty przez nie wielomian $w \in \pi_n$ postaci

$$w(x) = \sum_{k=0}^{n} f(x_k)\, \phi_k(x) \tag{3.2}$$

spełnia warunki interpolacji. Istotnie, dla dowolnego $i = 0, \ldots, n$ mamy

$$w(x_i) = \sum_{k=0}^{n} f(x_k)\, \phi_k(x_i) = f(x_i)\, \phi_i(x_i) + \sum_{k \neq i} f(x_k)\, \phi_k(x_i) = f(x_i).$$

Wzór (3.2) to tzw. *postać Lagrange'a wielomianu interpolacyjnego.*

**Jedyność rozwiązania**

Przypuśćmy, że $w_1, w_2 \in \pi_n$ to wielomiany interpolacyjne dla funkcji $f$, tj.

$$w_1(x_i) = w_2(x_i) = f(x_i), \qquad \text{dla } i = 0, \ldots, n.$$

Niech $w = w_1 - w_2$. Jest jasne, że $w \in \pi_n$. Ponadto, dla $i = 0, \ldots, n$

$$w(x_i) = w_1(x_i) - w_2(x_i) = f(x_i) - f(x_i) = 0.$$

Wielomian $w$ posiada zatem co najmniej $n+1$ różnych pierwiastków, musi więc być wielomianem zerowym. To oznacza, że $w_1 = w_2$, czyli wielomian interpolacyjny jest wyznaczony jednoznacznie.

### 3.1.1. Macierz Vandermonde'a

Zauważmy, że wielomian interpolacyjny może być również jednoznacznie zapisany jako kombinacja liniowa jednomianów bazowych $1, x, x^2, \ldots, x^n$:

$$w(x) = a_0 + a_1 x + \ldots + a_n x^n.$$

Warunki interpolacji prowadzą wówczas do następującego układu $n+1$ równań liniowych o $n+1$ niewiadomych $a_0, a_1, \ldots, a_n$

$$\begin{cases} a_0 + a_1 x_0 + a_2 x_0^2 + \ldots + a_n x_0^n = f(x_0) \\ a_0 + a_1 x_1 + a_2 x_1^2 + \ldots + a_n x_1^n = f(x_1) \\ \qquad\qquad \vdots \\ a_0 + a_1 x_n + a_2 x_n^2 + \ldots + a_n x_n^n = f(x_n) \end{cases}.$$

Układ ten zapisany w postaci macierzowej to

$$Aw = b, \tag{3.3}$$

gdzie $w = (a_0, \ldots, a_n)^T$ to poszukiwany wektor niewiadomych, $b = (f(x_0), \ldots, f(x_n))^T$ to wektor prawej strony, a $A \in \mathbb{R}^{(n+1)\times(n+1)}$ to macierz układu

$$A = \begin{pmatrix} 1 & x_0 & x_0^2 & \ldots & x_0^n \\ 1 & x_1 & x_1^2 & \ldots & x_1^n \\ \vdots & \vdots & \vdots & \cdots & \vdots \\ \vdots & \vdots & \vdots & \ldots & \vdots \\ 1 & x_n & x_n^2 & \ldots & x_n^n \end{pmatrix}.$$

Macierz $A$ to tzw. *macierz Vandermonde'a*. Na podstawie wykazanej jedyności wielomianu interpolacyjnego wnioskujemy, że jeżeli węzły $x_0, \ldots, x_n$ są parami różne, to macierz Vandermonde'a jest nieosobliwa (układ (3.3) posiada przecież dokładnie jedno rozwiązanie). Można również dość łatwo wykazać, że

$$\det A = V(x_0, \ldots, x_n) = \prod_{0 \leqslant j < i \leqslant n} (x_i - x_j);$$

prosty dowód indukcyjny pozostawiam jako ćwiczenie (dla chętnych).

## 3.2. Błąd interpolacji

Bardzo ważne pytanie, jakie możemy postawić, dotyczy błędu interpolacji, czyli różnicy pomiędzy funkcją a interpolującym ją wielomianem. W punktach węzłowych błąd ten jest oczywiście równy zero.

**Twierdzenie 3.2 (o błędzie interpolacji).** *Niech $f \in \mathcal{C}^{n+1}_{[a,b]}$, niech $x_0, \ldots, x_n \in [a,b]$ będą parami różnymi węzłami oraz niech $w \in \pi_n$ będzie wielomianem interpolacyjnym dla $f$, tj. $w(x_i) = f(x_i)$, dla $i = 0, \ldots, n$. Wówczas:*

$\forall x \in [a,b]\ \exists \xi = \xi(x) \in \operatorname{conv}(x, x_0, \ldots, x_n)$

$$f(x) - w(x) = \frac{f^{(n+1)}(\xi)}{(n+1)!} \prod_{i=0}^{n} (x - x_i). \tag{3.4}$$

**Dowód**: Dla $x \in \{x_0, \ldots, x_n\}$ teza jest oczywista. Rozważmy zatem $x \notin \{x_0, \ldots, x_n\}$. Możemy, dla ustalonego $x$, zdefiniować funkcję $G : [a,b] \ni z \to G(z) \in \mathbb{R}$ postaci

$$G(z) = f(z) - w(z) - R(x)\,\Phi(z),$$

gdzie

$$R(x) = \frac{f(x) - w(x)}{\Phi(x)} \quad \text{oraz} \quad \Phi(z) = \prod_{i=0}^{n} (z - x_i).$$

Jest jasne, że $G \in \mathcal{C}^{n+1}_{[a,b]}$ oraz

$$G(x) = G(x_0) = G(x_1) = \ldots = G(x_n) = 0.$$

Oznacza to, na podstawie twierdzenia Rolle'a, że funkcja $G'$ ma co najmniej $n+1$ różnych zer w przedziale $\operatorname{conv}(x, x_0, \ldots, x_n)$. Rozumując indukcyjnie wnioskujemy, że funkcja $G^{(n+1)}$ ma co najmniej jeden pierwiastek w przedziale $\operatorname{conv}(x, x_0, \ldots, x_n)$; oznaczmy go jako $\xi$, tj.

$$0 = G^{(n+1)}(\xi) = f^{(n+1)}(\xi) - R(x)\,(n+1)!$$

Stąd wynika teza. ∎

Postać Lagrange'a wielomianu interpolacyjnego posiada istotną wadę: dołączenie kolejnego węzła wymaga wyznaczania od nowa wszystkich funkcji $\{\phi_i\}$ rozpinających poszukiwany wielomian interpolacyjny. Dzieje się tak, gdyż każda funkcja $\phi_i$ zależna jest od wartości wszystkich węzłów. Wady tej nie posiada postać Newtona wielomianu interpolacyjnego oparta na tzw. różnicach dzielonych.

## 3.3. Różnice dzielone

**Definicja 3.1 (różnica dzielona rzędu $p$).** *Różnicą dzieloną rzędu $p \geqslant 1$ funkcji $f$ opartą na parami różnych węzłach $x_0, \ldots, x_p$ nazywamy liczbę*

$$f[x_0, \ldots, x_p] = \sum_{i=0}^{p} \frac{f(x_i)}{\prod\limits_{j \neq i, j=0}^{p} (x_i - x_j)}.$$

*Dla ujednolicenia zapisu przyjmujemy, że $f[x_0] = f(x_0)$.*

**Stwierdzenie 3.3 (własności różnic dzielonych).** *Niech $x_0, \ldots, x_p$ będą parami różnymi węzłami.*

**W1.** *Dla dowolnej permutacji $\sigma : \{0, \ldots, p\} \to \{0, \ldots, p\}$ zachodzi*

$$f[x_0, \ldots, x_p] = f[x_{\sigma(0)}, \ldots, x_{\sigma(p)}].$$

**W2.** *Przy ustalonych wartościach $x_0, \ldots, x_p$ funkcjonał $f \to f[x_0, \ldots, x_p]$ jest liniowy.*
**W3.** *Dla $p \geqslant 1$*

$$f[x_0, \ldots, x_p] = \frac{f[x_1, \ldots, x_p] - f[x_0, \ldots, x_{p-1}]}{x_p - x_0}. \tag{3.5}$$

**Dowód**: Dowody własności W1 oraz W2 to proste konsekwencje definicji. Własność W3 udowodnimy indukcyjnie.

Proste rachunki pokazują, że dla $p = 1$ oraz $p = 2$ wzór (3.5) zachodzi. Z kolei

$$\begin{aligned}
&\frac{f[x_1, \ldots, x_p] - f[x_0, \ldots, x_{p-1}]}{x_p - x_0} = \\
&= \frac{1}{x_p - x_0} \left( \sum_{i=1}^{p} \frac{f(x_i)}{\prod\limits_{j \neq i, j=1}^{p} (x_i - x_j)} - \sum_{i=0}^{p-1} \frac{f(x_i)}{\prod\limits_{j \neq i, j=0}^{p-1} (x_i - x_j)} \right) = \\
&= \frac{1}{x_p - x_0} \left( \sum_{i=1}^{p} \frac{f(x_i)(x_i - x_0)}{\prod\limits_{j \neq i, j=0}^{p} (x_i - x_j)} - \sum_{i=0}^{p-1} \frac{f(x_i)(x_i - x_p)}{\prod\limits_{j \neq i, j=0}^{p} (x_i - x_j)} \right) = \\
&= \frac{1}{x_p - x_0} \left( \sum_{i=1}^{p-1} \frac{f(x_i)}{M_i} (x_i - x_0 - x_i + x_p) + \frac{f(x_p)(x_p - x_0)}{M_p} - \frac{f(x_0)(x_0 - x_p)}{M_0} \right),
\end{aligned}$$

gdzie $M_i = \prod\limits_{j \neq i, j=0}^{p} (x_i - x_j)$. Stąd

$$\frac{f[x_1, \ldots, x_p] - f[x_0, \ldots, x_{p-1}]}{x_p - x_0} = \sum_{i=0}^{p} \frac{f(x_i)}{M_i} = \sum_{i=0}^{p} \frac{f(x_i)}{\prod\limits_{j \neq i, j=0}^{p} (x_i - x_j)}$$

co kończy dowód. ∎

**Twierdzenie 3.4 (postać Newtona wielomianu interpolacyjnego).** *Przy powyższych założeniach niech $w_n \in \pi_n$ będzie wielomianem zdefiniowanym w następujący sposób: $w_0(x) = f(x_0)$ oraz dla $n \geqslant 1$*

$$\begin{aligned}
w_n(x) = &f[x_0] + f[x_0, x_1](x - x_0) + f[x_0, x_1, x_2](x - x_0)(x - x_1) + \ldots \\
&\ldots + f[x_0, \ldots, x_n](x - x_0) \cdots (x - x_{n-1}).
\end{aligned} \tag{3.6}$$

*Wówczas*

**a)** *dla $x \notin \{x_0, \ldots, x_n\}$*

$$f(x) - w_n(x) = f[x, x_0, \ldots, x_n] \prod_{i=0}^{n} (x - x_i); \tag{3.7}$$

**b)** *dla $i = 0, \ldots, n : f(x_i) = w_n(x_i)$.*

**Dowód**: Punkt a) udowodnimy indukcyjnie. Dla $n = 0$ mamy

$$f[x, x_0](x - x_0) = \frac{f(x_0) - f(x)}{x_0 - x}(x - x_0) = f(x) - w_0(x).$$

Podobnie sprawdzamy prawdziwość tezy dla $n = 1$. Z kolei

$$\begin{aligned} f[x, x_0, \ldots, x_n] \prod_{i=0}^{n} (x - x_i) &= \frac{f[x_0, \ldots, x_n] - f[x, x_0, \ldots, x_{n-1}]}{x_n - x} \prod_{i=0}^{n} (x - x_i) = \\ &= (f[x, x_0, \ldots, x_{n-1}] - f[x_0, \ldots, x_n]) \prod_{i=0}^{n-1} (x - x_i) = \\ &= f[x, x_0, \ldots, x_{n-1}] \prod_{i=0}^{n-1} (x - x_i) - f[x_0, \ldots, x_n] \prod_{i=0}^{n-1} (x - x_i), \end{aligned}$$

skąd, na podstawie założenia indukcyjnego, wnioskujemy, że

$$\begin{aligned} f[x, x_0, \ldots, x_n] \prod_{i=0}^{n} (x - x_i) &= f(x) - w_{n-1}(x) - f[x_0, \ldots, x_n] \prod_{i=0}^{n-1} (x - x_i) = \\ &= f(x) - w_n(x). \end{aligned}$$

Aby uzasadnić punkt b) zauważmy, że $w_n(x_0) = f(x_0) = w_0(x_0)$ oraz dla $k \geqslant 1$

$$\begin{aligned} w_n(x_k) =& f(x_0) + f[x_0, x_1](x_k - x_0) + f[x_0, x_1, x_2](x_k - x_0)(x_k - x_1) + \ldots \\ &\ldots + f[x_0, \ldots, x_k](x_k - x_0) \cdots (x_k - x_{k-1}) = w_k(x_k). \end{aligned}$$

Zapiszmy teraz udowodniony przed chwilą wzór (3.7) dla $n = k$ oraz dla dowolnego $x \notin \{x_0, \ldots, x_k\}$:

$$f(x) - w_k(x) = f[x, x_0, \ldots, x_k] \prod_{i=0}^{k} (x - x_i).$$

Stąd, dla $x = x_{k+1}$ otrzymujemy równość

$$f(x_{k+1}) - w_k(x_{k+1}) = f[x_{k+1}, x_0, \ldots, x_k] \prod_{i=0}^{k} (x_{k+1} - x_i),$$

którą równoważnie możemy zapisać jako

$$\begin{aligned} f(x_{k+1}) &= w_k(x_{k+1}) + f[x_0, \ldots, x_{k+1}] \prod_{i=0}^{k} (x_{k+1} - x_i) = \\ &= w_{k+1}(x_{k+1}) = w_n(x_{k+1}). \end{aligned}$$

To kończy dowód. ∎

**Uwaga 3.1.** *Z punktu b) twierdzenia 3.4 wynika, że wielomian $w_n$ określony przez* (3.6) *to wielomian interpolacyjny dla funkcji $f$. Jest to tzw. postać Newtona wielomianu interpolacyjnego.*

**Uwaga 3.2.** *Punkt a) twierdzenia 3.4 podaje wzór na błąd interpolacji uzyskany przy braku jakichkolwiek założeń dotyczących regularności funkcji $f$. Porównując ten wzór ze wzorem* (3.4) *wnioskujemy, że dla funkcji $f \in \mathcal{C}^n_{[a,b]}$ oraz dla parami różnych węzłów $x_0, \ldots, x_n \in [a, b]$ zachodzi*

$$f[x_0, \ldots, x_n] = \frac{f^{(n)}(\xi)}{n!},$$

*dla pewnego $\xi \in \operatorname{conv}(x_0, \ldots, x_n)$.*

**Uwaga 3.3.** *Różnice dzielone*

$$f[x_0], f[x_0, x_1], f[x_0, x_1, x_2], \ldots, f[x_0, \ldots, x_n]$$

*to współczynniki stojące przy wielomianach tworzących bazę Newtona przestrzeni $\pi_n$, tj.*

$$1, x - x_0, (x - x_0)(x - x_1), \ldots, (x - x_0) \cdots (x - x_{n-1})$$

*rozpinających wielomian interpolacyjny w postaci Newtona. Z postaci tej wynika, że dołączenie kolejnych punktów węzłowych nie wymaga korekty ani już wyznaczonych wielomianów bazowych, ani stojących przy nich współczynników.*

Pokażemy teraz sposób na efektywne wyznaczanie współczynników wielomianu interpolacyjnego zapisanego w bazie Newtona.

### 3.3.1. Algorytm różnic dzielonych

Dla danych parami różnych węzłów $x_0, \ldots, x_n$ oraz wartości $y_0, \ldots, y_n$ (dla uproszczenia zapisu oznaczamy tak wartości funkcji $f$ w punktach węzłowych, tj. $f(x_0), \ldots, f(x_n)$), chcemy wyznaczyć różnice dzielone kolejnych rzędów

$$f[x_0, x_1], f[x_0, x_1, x_2], \ldots, f[x_0, \ldots, x_n].$$

Niech $y_{p,q} = f[x_{p-q}, \ldots, x_p]$, dla $p = 0, \ldots, n$; $q = 0, \ldots, p$. Poszukiwane wartości to

$$y_{p,p} = f[x_0, \ldots, x_p]$$

dla $p = 1, \ldots, n$. Na podstawie zależności rekurencyjnej (3.5) dla różnic dzielonych możemy zapisać

$$y_{p,q} = \frac{f[x_{p-q+1}, \ldots, x_p] - f[x_{p-q}, \ldots, x_{p-1}]}{x_p - x_{p-q}} = \frac{y_{p,q-1} - y_{p-1,q-1}}{x_p - x_{p-q}}$$

dla $p = 1, \ldots, n$; $q = 1, \ldots, p$. Wynika stąd następujący

**Algorytm różnic dzielonych**

$$\begin{array}{l} \texttt{For } p = 0, \ldots, n : y_{p,0} = f(x_p). \\ \texttt{For } q = 1, \ldots, n \\ \quad \texttt{for } p = q, \ldots, n : \; y_{p,q} = \frac{y_{p,q-1} - y_{p-1,q-1}}{x_p - x_{p-q}}. \end{array}$$

Kolejność wyliczania poszczególnych wartości różnic dzielonych wygodnie zapamiętać zapisując je w postaci macierzy trójkątnej

$$\begin{array}{cc|ccccc} x_0 & f(x_0) & & & & & \\ x_1 & f(x_1) & f[x_0, x_1] & & & & \\ x_2 & f(x_2) & f[x_1, x_2] & f[x_0, x_1, x_2] & & & \\ \vdots & \vdots & \vdots & \vdots & & & \\ \vdots & \vdots & \vdots & \vdots & \ddots & & \\ x_n & f(x_n) & f[x_{n-1}, x_n] & f[x_{n-2}, x_{n-1}, x_n] & \cdots & f[x_0, \ldots, x_n] \\ & \uparrow & \uparrow & \uparrow & & \uparrow \\ & y_{p,0} & y_{p,1} & y_{p,2} & \cdots & y_{n,n} \end{array}.$$

**Przykład 3.1.** *Wyznaczymy wielomian (możliwie niskiego stopnia) interpolujący poniższą tabelę*

| $x_i$ | $-1$ | $0$ | $1$ | $2$ |
|---|---|---|---|---|
| $f(x_i)$ | $0$ | $1$ | $2$ | $9$ |

.

*Poszukiwany wielomian $w_3 \in \pi_3$ zapiszemy wykorzystując bazę Newtona, tj.*

$$\begin{aligned} w_3(x) =& f(x_0) + f[x_0, x_1](x - x_0) + f[x_0, x_1, x_2](x - x_0)(x - x_1) + \\ &+ f[x_0, x_1, x_2, x_3](x - x_0)(x - x_1)(x - x_2) = \\ =& f(-1) + f[-1, 0](x + 1) + f[-1, 0, 1](x + 1)x + f[-1, 0, 1, 2](x + 1)x(x - 1). \end{aligned}$$

*Ponieważ*

$$\begin{array}{cc|l} -1 & 0 & \\ 0 & 1 & f[-1,0] = 1 \\ 1 & 2 & f[0,1] = 1 \quad f[-1,0,1] = 0 \\ 2 & 9 & f[1,2] = 7 \quad f[0,1,2] = 3 \quad f[-1,0,1,2] = 1 \end{array},$$

*zatem*

$$w_3(x) = 0 \cdot 1 + 1 \cdot (x + 1) + 0 \cdot (x + 1)x + 1 \cdot (x + 1)x(x - 1) = x^3 + 1.$$

**Koszt i właściwości numeryczne:** należy wyznaczyć $\left(n^2 + n\right)/2$ elementy, każdy z nich wyliczamy wykonując trzy działania (dwie sumy, jeden iloraz) co daje całkowity koszt nie większy niż $3\left(n^2 + n\right)/2$ operacji arytmetycznych.

Właściwości numeryczne algorytmu zawarte są w poniższym twierdzeniu.

**Twierdzenie 3.5.** *Jeżeli węzły $x_0, \ldots, x_n$ są uporządkowane monotonicznie to obliczone w arytmetyce* fl *algorytmem różnic dzielonych wartości $y_{1,1}, \ldots, y_{n,n}$ spełniają zależność*

$$\mathrm{fl}(y_{p,p}) = \sum_{i=0}^{p} \frac{\widetilde{y_i}}{\prod\limits_{j \neq i, j=0}^{p} (x_i - x_j)},$$

*gdzie $\widetilde{y_i} = y_i(1 + \varepsilon_{ip})$, $|\varepsilon_{ip}| \leqslant_1 3p \cdot 2^{-t}$.*

**Dowód**: Dowód twierdzenia pomijamy; zainteresowanych dowodem odsyłamy do monografii J.M. Jankowscy, *Przegląd metod i algorytmów numerycznych. Część 1.* WNT Warszawa 1981 (zob. twierdzenie 2.8 na str. 51). ■

Jak wynika z powyższego twierdzenia, algorytmem różnic dzielonych w sposób numerycznie poprawny wyznaczamy każdy współczynnik $y_{p,p}$. Ponieważ zaburzenia $\varepsilon_{ip}$ są zależne od $p$, zatem algorytm ten nie jest numerycznie poprawny ze względu na cały wektor wartości $(y_{1,1}, \ldots, y_{n,n})$; algorytm taki nazywamy *prawie numerycznie poprawnym.*

## 3.4. Zbieżność procesu interpolacji ☠

Niech $\{w_n\}_{n \in \mathbb{N}}$ będzie ciągiem wielomianów interpolacyjnych dla funkcji $f$ opartych na węzłach

$$a \leqslant x_0^{(n)} < x_1^{(n)} < \ldots < x_n^{(n)} \leqslant b.$$

Zastanowimy się teraz, czy ciąg wielomianów interpolacyjnych jest zbieżny jednostajnie do funkcji interpolowanej.

Zauważmy, że jeżeli $f \in \mathcal{C}^{\infty}_{[a,b]}$ to na podstawie twierdzenia 3.2

$$\|f - w_n\|_{[a,b]} \leqslant \frac{\|f^{(n+1)}\|_{[a,b]}}{(n+1)!} \max_{x \in [a,b]} \prod_{i=0}^{n} \left|x - x_i^{(n)}\right| \leqslant \|f^{(n+1)}\|_{[a,b]} \frac{(b-a)^{n+1}}{(n+1)!}.$$

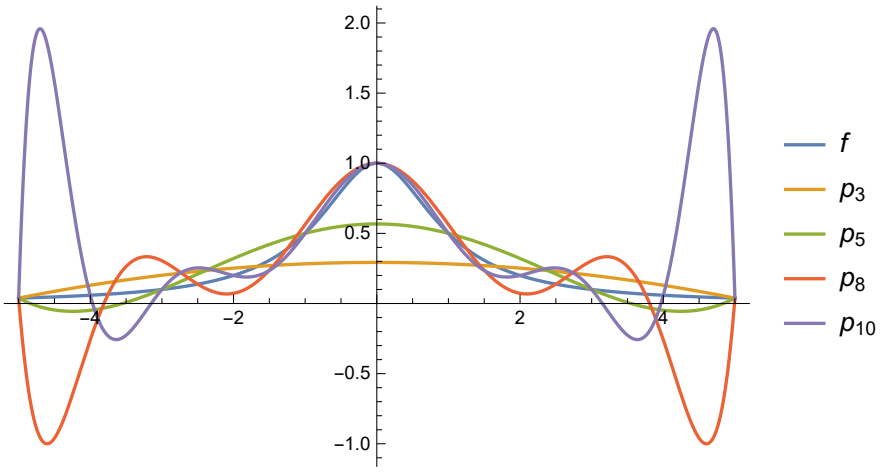
Wynika stąd, że dla funkcji klasy $\mathcal{C}^{\infty}$, której pochodne spełniają warunek

$$\|f^{(n)}\|_{[a,b]} = \mathcal{O}\big(M_f^n,\big) \tag{3.8}$$

dla pewnej stałej $M_f > 0$, proces interpolacji jest zbieżny jednostajnie, tj.

$$\lim_{n \to \infty} \|f - w_n\|_{[a,b]} = 0.$$

Poniższy przykład pokazuje, że sama wysoka regularność funkcji $f$ nie gwarantuje tej zbieżności.




Rysunek 3.1. Wykresy funkcji $f(x) = 1/(1+x^2)$ oraz kilku wielomianów interpolacyjnych $p_n$ opartych na węzłach równoodległych.

**Przykład 3.2 (przykład Rungego).** *Niech*

$$f(x) = \frac{1}{1+x^2}, \qquad x \in [-5,5].$$

*Funkcja $f$ jest klasy $C^\infty_{[-5,5]}$ i dla $|x| < 1$ może być zapisana w postaci*

$$\frac{1}{1+x^2} = \sum_{k=0}^{\infty} (-1)^k x^{2k}.$$

*Wynika stąd, że $f^{(2k)}(0) = (-1)^k (2k)!$, czyli*

$$\|f^{(2k)}\|_{[-5,5]} \geqslant |f^{(2k)}(0)| = (2k)!;$$

*warunek* (3.8) *nie jest więc spełniony. Okazuje się, że dla węzłów równoodległych*

$$x_k = -5 + 10\frac{k}{n}, \qquad \text{dla } k = 0, \ldots, n$$

*ciąg wielomianów interpolacyjnych $\{p_n\}$ dla $f$ spełnia warunek*

$$\lim_{n\to\infty} \|f - p_n\|_{[-5,5]} = +\infty.$$

W przypadku ogólnym mają zastosowanie następujące twierdzenia.

**Twierdzenie 3.6 (Faber 1914).** *Dla dowolnego ciągu węzłów istnieje funkcja ciągła, do której ciąg jej wielomianów interpolacyjnych nie jest zbieżny jednostajnie.*

**Twierdzenie 3.7.** *Dla dowolnej funkcji ciągłej $f$ istnieje ciąg węzłów, dla których ciąg wielomianów interpolacyjnych funkcji $f$ jest do niej zbieżny jednostajnie.*

## Rozdział 4

# Interpolacja wielomianowa Hermite'a

### 4.1. Sformułowanie problemu, istnienie i jedyność rozwiązania

**Sformułowanie problemu**

Dla zadanej funkcji $f : [a,b] \to \mathbb{R}$, węzłów $x_0, \ldots, x_k \in [a,b]$ oraz liczb naturalnych (tzw. *krotności*) $m_0, \ldots, m_k : \sum_{i=0}^{k} m_i = n+1$ poszukujemy wielomianu $H_n \in \pi_n$ spełniającego warunki

$$H_n^{(j)}(x_i) = f^{(j)}(x_i), \qquad \text{dla } i = 0, \ldots, k;\ j = 0, \ldots, m_i - 1. \tag{4.1}$$

Wielomian ten będziemy nazywać wielomianem interpolacyjnym Hermite'a dla funkcji $f$.

**Przykład 4.1.** *W przypadku, gdy $n = k$ oraz $m_0 = \ldots = m_k = 1$ problem interpolacji Hermite'a redukuje się do problemu interpolacji wielomianowej Lagrange'a. Z kolei, dla $m_0 = n+1$ mamy do czynienia z problemem interpolacji Taylora.*

**Jedyność rozwiązania**

Przypuśćmy, że $H_{n,1}, H_{n,2} \in \pi_n$ to wielomiany interpolacyjne Hermite'a dla funkcji $f$, tj.

$$H_{n,1}^{(j)}(x_i) = H_{n,2}^{(j)}(x_i) = f^{(j)}(x_i), \qquad \text{dla } i = 0, \ldots, k;\ j = 0, \ldots, m_i - 1.$$

Niech $H = H_{n,1} - H_{n,2}$. Jest jasne, że $H \in \pi_n$. Ponadto, dla $i = 0, \ldots, k;\ j = 0, \ldots, m_i - 1$

$$H^{(j)}(x_i) = H_{n,1}^{(j)}(x_i) - H_{n,2}^{(j)}(x_i) = f^{(j)}(x_i) - f^{(j)}(x_i) = 0.$$

Wielomian $H$ posiada zatem pierwiastki o łącznej krotności nie mniejszej niż $n+1$, musi więc być wielomianem zerowym. To oznacza, że $H_{n,1} = H_{n,2}$, czyli wielomian interpolacyjny Hermite'a, o ile istnieje, jest wyznaczony jednoznacznie.

**Istnienie rozwiązania** ☠

Niech $s : \{0, \ldots, k\} \to \mathbb{N}$ będzie funkcją postaci

$$s(i) = \begin{cases} 0 & \text{dla } i = 0 \\ m_0 + \ldots + m_{i-1} & \text{dla } 1 \leqslant i \leqslant k \end{cases}.$$

Wykorzystując funkcję $s$, dowolną liczbę $l \in \{0, \ldots, n\}$ możemy zapisać jednoznacznie jako

$$l = s(i) + j, \quad \text{gdzie } j \in \{0, \ldots, m_i - 1\}.$$

Jak łatwo się przekonać, indeks $i$ determinujący wartości $s(i)$ oraz $j$ występujące w powyższym rozkładzie spełnia zależność

$$i = \max\{k : s(k) \leqslant l\}.$$

Niech $p_0, \ldots, p_n \in \pi_n$ będą wielomianami postaci

$$p_l(x) = \begin{cases} (x - x_0)^l, & \text{dla } l \in \{0, \ldots, m_0\} \\ (x - x_0)^{m_0} \cdots (x - x_{i-1})^{m_{i-1}} (x - x_i)^j, & \text{dla } l = s(i) + j \in \{m_0 + 1, \ldots, n\} \end{cases}.$$

Wielomiany $p_0, \ldots, p_n$ tworzą bazę przestrzeni $\pi_n$. Niech

$$H_n(x) = \sum_{i=0}^{n} b_i p_i(x); \tag{4.2}$$

pokażemy sposób wyznaczania współczynników $b_0, \ldots, b_n$, dla których wielomian $H_n$ spełnia warunki interpolacji (4.1). Przedstawiona konstrukcja będzie rekurencyjna i efektywna co zagwarantuje istnienie rozwiązania problemu interpolacji wielomianowej Hermite'a.

Oczywiście

$$H_n(x_0) = b_0 = f(x_0),$$

co oznacza, że wartość współczynnika $b_0$ znamy. Przypuśćmy, że współczynniki $b_0, \ldots, b_{l-1}$ mamy już wyznaczone; pokażemy jak wyznaczyć współczynnik $b_l$. W tym celu zapiszmy wielomian $H_n$ jako

$$H_n(x) = A(x) + b_l p_l(x) + B(x), \tag{4.3}$$

gdzie składnik

$$A(x) = \sum_{i=0}^{l-1} b_i p_i(x)$$

jest znany, a współczynników wielomianu

$$B(x) = \sum_{i=l+1}^{n} b_i p_i(x)$$

nie znamy. Zauważmy, że

$$\begin{aligned} p_l(x) = p_{s(i)+j}(x) &= (x - x_0)^{m_0} \cdots (x - x_{i-1})^{m_{i-1}} (x - x_i)^j = \\ &= p_{s(i)}(x)(x - x_i)^j \end{aligned} \tag{4.4}$$

oraz

$$B(x) = (x - x_i)^{j+1} R(x), \tag{4.5}$$

gdzie $R$ jest pewnym wielomianem. W celu wyznaczenia współczynnika $b_l = b_{s(i)+j}$ wystarczy skorzystać z równości

$$H_n^{(j)}(x_i) = f^{(j)}(x_i). \tag{4.6}$$

Z (4.3) wynika, że

$$H_n^{(j)}(x_i) = A^{(j)}(x_i) + b_l p_l^{(j)}(x_i) + B^{(j)}(x_i).$$

Na podstawie (4.4) oraz (4.5) otrzymujemy odpowiednio

$$p_l^{(j)}(x_i) = p_{s(i)}(x_i) \cdot j!$$

oraz

$$B^{(j)}(x_i) = 0,$$

zatem z warunku (4.6) wynika, że

$$b_l = \frac{f^{(j)}(x_i) - A^{(j)}(x_i)}{p_{s(i)}(x_i) \cdot j!}.$$

W ten sposób wyznaczamy kolejne współczynniki wielomianu $H_n$.

## 4.2. Błąd interpolacji

**Twierdzenie 4.1 (o błędzie interpolacji).** *Przy powyższych oznaczeniach niech $f \in \mathcal{C}^{n+1}_{[a,b]}$, niech $x_0, \ldots, x_k \in [a,b]$ będą parami różnymi węzłami oraz niech $H_n \in \pi_n$ będzie wielomianem interpolacyjnym Hermite'a dla funkcji $f$, tj.*

$$H_n^{(j)}(x_i) = f^{(j)}(x_i), \qquad \text{dla } i = 0, \ldots, k;\ j = 0, \ldots, m_i - 1.$$

*Wówczas:* $\forall x \in [a,b]\ \exists \xi = \xi(x) \in \operatorname{conv}(x, x_0, \ldots, x_k)$

$$f(x) - H_n(x) = \frac{f^{(n+1)}(\xi)}{(n+1)!} \prod_{i=0}^{k} (x - x_i)^{m_i}.$$

**Dowód**: Dowód jest pewną modyfikacją dowodu twierdzenia 3.2. Przeprowadzimy go w szczególnym przypadku, gdy $m_0 = \ldots = m_k = 2$; w ogólnym przypadku dowód jest nieco bardziej techniczny.

Zauważmy, że dla $x \in \{x_0, \ldots, x_k\}$ teza twierdzenia zachodzi dla dowolnego $\xi$. Niech zatem $x \notin \{x_0, \ldots, x_k\}$. Możemy wówczas zdefiniować funkcję $G$ postaci

$$G(z) = f(z) - H_{2k+1}(z) - R(x)\, w^2(z),$$

gdzie

$$R(x) = \frac{f(x) - H_{2k+1}(x)}{w^2(x)} \quad \text{oraz} \quad w(z) = \prod_{i=0}^{k} (z - x_i).$$

Jest jasne, że $G \in \mathcal{C}^{2k+2}_{[a,b]}$ oraz

$$G(x) = G(x_0) = G(x_1) = \ldots = G(x_k) = 0.$$

Na podstawie twierdzenia Rolle'a wnioskujemy, że istnieją punkty $\eta_1, \ldots, \eta_{k+1}$, wszystkie różne od punktów $x, x_0, \ldots, x_k$, dla których

$$G'(\eta_1) = G'(\eta_2) = \ldots = G'(\eta_{k+1}) = 0.$$

Mamy również

$$G'(x_0) = G'(x_1) = \ldots = G'(x_k) = 0.$$

Funkcja $G'$ posiada zatem co najmniej $2k+2$ różnych miejsc zerowych. Na podstawie twierdzenia Rolle'a funkcja $G''$ posiada co najmniej $2k+1$ różnych miejsc zerowych. Powtarzając to rozumowanie wnioskujemy, że funkcja $G^{(2k+2)}$ posiada co najmniej jedno miejsce zerowe $\xi \in \operatorname{conv}(x, x_0, \ldots, x_k)$, tj.

$$0 = G^{(2k+2)}(\xi) = f^{(2k+2)}(\xi) - R(x)\,(2k+2)!$$

Stąd wynika teza. ∎

Pokażemy teraz efektywny sposób wyznaczania współczynników wielomianu interpolacyjnego Hermite'a $H_n$. W tym celu wszystkie węzły ustawmy w ciągu niemalejącym (każdy węzeł występuje tyle razy ile wynosi jego krotność):

$$\underbrace{x_0 \leqslant \ldots \leqslant x_0}_{\times m_0} \leqslant \underbrace{x_1 \leqslant \ldots \leqslant x_1}_{\times m_1} \leqslant \ldots \leqslant \underbrace{x_k \leqslant \ldots \leqslant x_k}_{\times m_k};$$

kolejne wyrazy tego ciągu oznaczmy jako $x_0^*,\ x_1^*, \ldots$, tj.

$$x_0 = x_0^* \leqslant x_1^* \leqslant \ldots \leqslant x_n^* = x_k.$$

Poszukiwany wielomian interpolacyjny Hermite'a $H_n$ ma wówczas postać

$$\begin{aligned} H_n(x) =& f[x_0^*] + f[x_0^*, x_1^*](x - x_0^*) + f[x_0^*, x_1^*, x_2^*](x - x_0^*)(x - x_1^*) + \dots \\ & \dots + f[x_0^*, \dots, x_n^*](x - x_0^*) \cdots (x - x_{n-1}^*). \end{aligned} \tag{4.7}$$

Współczynniki wielomianu $H_n$ wyznaczamy wprowadzonym w poprzednim rozdziale algorytmem różnic dzielonych; wszędzie tam, gdzie pojawi się konieczność wyliczenia wartości $f[x_i, \dots, x_i]$, wyznaczamy ją w następujący sposób:

$$f[\underbrace{x_i, \dots, x_i}_{\times k+1}] = \frac{f^{(k)}(x_i)}{k!}. \tag{4.8}$$

**Przykład 4.2 (reprezentatywny).** *Wyznaczymy wielomian możliwie niskiego stopnia interpolujący poniższą tabelę*

| $x_i$ | $-1$ | $0$ | $1$ |
|---|---|---|---|
| $f(x_i)$ | $0$ | $1$ | $2$ |
| $f'(x_i)$ | $3$ | $0$ | |

*Mamy*

$$x_0 = -1, m_0 = 2; \qquad x_1 = 0, m_1 = 2; \qquad x_2 = 1, m_2 = 1$$

*lub, uwzględniając przyjęte oznaczenia,*

$$x_0^* = -1,\ x_1^* = -1,\ x_2^* = 0,\ x_3^* = 0,\ x_4^* = 1.$$

*Poszukiwany wielomian $H_4 \in \pi_4$ zapiszemy w postaci (4.7), tj.*

$$\begin{aligned} H_4(x) =& f(-1) + f[-1,-1](x+1) + f[-1,-1,0](x+1)^2 + \\ & + f[-1,-1,0,0](x+1)^2 x + f[-1,-1,0,0,1](x+1)^2 x^2. \end{aligned}$$

*Wykorzystując zmodyfikowany algorytm różnic dzielonych otrzymujemy*[1]

| $x_i$ ↓ | $f(x_i)$ ↓ | | | | |
|---|---|---|---|---|---|
| $-1$ | $0$ | | | | |
| $-1$ | $0$ | $f[-1,-1] = 3$ | | | |
| $0$ | $1$ | $f[-1,0] = 1$ | $f[-1,-1,0] = -2$ | | |
| $0$ | $1$ | $f[0,0] = 0$ | $f[-1,0,0] = -1$ | $f[-1,-1,0,0] = 1$ | |
| $1$ | $2$ | $f[0,1] = 1$ | $f[0,0,1] = 1$ | $f[-1,0,0,1] = 1$ | $f[-1,-1,0,0,1] = 0.$ |

*Ostatecznie*

$$H_4(x) = 3(x+1) - 2(x+1)^2 + (x+1)^2 x = x^3 + 1.$$

**Przykład 4.3 (wzór Taylora).** *Dla $k = 0$ oraz $m_0 = n + 1$ mamy*

$$\begin{aligned} H_n(x) =& f[x_0] + f[x_0, x_0](x - x_0) + f[x_0, x_0, x_0](x - x_0)^2 + \dots \\ & \dots + f[x_0, \dots, x_0](x - x_0)^n. \end{aligned}$$

*Stąd, uwzględniając (4.8),*

$$H_n(x) = f(x_0) + f'(x_0)(x - x_0) + \frac{f''(x_0)}{2}(x - x_0)^2 + \dots + \frac{f^{(n)}(x_0)}{n!}(x - x_0)^n.$$

*Z kolei, na podstawie twierdzenia 4.1 otrzymujemy*

$$f(x) - H_n(x) = \frac{f^{(n+1)}(\xi)}{(n+1)!}(x - x_0)^{n+1},$$

*gdzie $\xi \in \operatorname{conv}(x, x_0)$.*

[1] Dla przykładu,

$$f[-1,-1,0] = \frac{f[-1,0] - f[-1,-1]}{0 - (-1)} = f[-1,0] - f'(-1) = 1 - 3 = -2.$$

## Rozdział 5

# Interpolacja funkcjami sklejanymi

### 5.1. Funkcje sklejane oraz naturalne funkcje sklejane

Niech $\triangle = \{x_0, \ldots, x_n\}$, gdzie $x_0 < \ldots < x_n$.

**Definicja 5.1 (funkcja sklejana stopnia $m$).** *Funkcję $S : \mathbb{R} \to \mathbb{R}$ nazywamy funkcją sklejaną stopnia $m \in \mathbb{N}$ dla podziału $\triangle$, jeżeli*

**(a)** $S|_{(x_{i-1},x_i)} \in \pi_m$, *dla* $i = 0, \ldots, n+1$; $x_{-1} := -\infty$, $x_{n+1} := +\infty$;
**(b)** $S \in \mathcal{C}^{m-1}_{\mathbb{R}}$.

**Definicja 5.2 (naturalna funkcja sklejana stopnia $2m-1$).** *Funkcję $S : \mathbb{R} \to \mathbb{R}$ nazywamy naturalną funkcją sklejaną stopnia $2m-1$ dla podziału $\triangle$, jeżeli*

**(a)** *$S$ jest funkcją sklejaną stopnia $2m-1$ dla podziału $\triangle$, tj.*
- $S|_{(x_{i-1},x_i)} \in \pi_{2m-1}$, *dla* $i = 0, \ldots, n+1$;
- $S \in \mathcal{C}^{2m-2}_{\mathbb{R}}$;

**(b)** $S|_{(-\infty,x_0)}, S|_{(x_n,+\infty)} \in \pi_{m-1}$.

Przez $\mathcal{S}_m(\triangle)$ oraz $\mathcal{N}_{2m-1}(\triangle)$ będziemy oznaczyć odpowiednio przestrzeń funkcji sklejanych stopnia $m$ oraz przestrzeń naturalnych funkcji sklejanych stopnia $2m-1$ dla podziału $\triangle$.

#### 5.1.1. Reprezentacje funkcji sklejanych

**Stwierdzenie 5.1 (postać funkcji sklejanej stopnia $m$).** *Dla każdej funkcji $S \in \mathcal{S}_m(\triangle)$ istnieją wielomian $p \in \pi_m$ oraz współczynniki $a_0, \ldots, a_n \in \mathbb{R}$, dla których*

$$S(x) = p(x) + \sum_{j=0}^{n} a_j (x - x_j)^m_+ , \tag{5.1}$$

*gdzie* $x^m_+ = \begin{cases} x^m & dla\ x \geqslant 0 \\ 0 & dla\ x < 0 \end{cases}$.

**Dowód**: Z definicji funkcji sklejanej stopnia $m$ wynika, że na każdym z przedziałów

$$(-\infty, x_0], \quad [x_0, x_1], \quad \ldots, \quad [x_{n-1}, x_n], \quad [x_n, \infty)$$

funkcja $S$ jest wielomianem stopnia co najwyżej $m$.

Rozważmy przedział $(-\infty, x_1]$. Na jego części $(-\infty, x_0]$ funkcja $S$ to wielomian, powiedzmy $S_0$, który po rozwinięciu w szereg Taylora w otoczeniu punktu $x_0$ możemy zapisać jako

$$S_0(x) = \alpha_0 + \alpha_1(x - x_0) + \alpha_2(x - x_0)^2 + \ldots + \alpha_m(x - x_0)^m,$$

oczywiście $\alpha_k = \frac{S_0^{(k)}(x_0)}{k!}$ dla $k = 0, \ldots, m$. Z kolei, na przedziale $[x_0, x_1]$ funkcja $S$ to wielomian, powiedzmy $S_1$, który rozwinięty w szereg Taylora w otoczeniu punktu $x_0$ przyjmuje postać

$$S_1(x) = \beta_0 + \beta_1(x - x_0) + \beta_2(x - x_0)^2 + \ldots + \beta_m(x - x_0)^m,$$

gdzie $\beta_k = \frac{S_1^{(k)}(x_0)}{k!}$ dla $k = 0, \ldots, m$. Sklejenie tych dwóch wielomianów – zawężonych odpowiednio do przedziału $(-\infty, x_0]$ oraz $[x_0, x_1]$ – w punkcie $x_0$ jest klasy $\mathcal{C}^{m-1}$. To oznacza, że

$$\alpha_k = \beta_k \qquad \text{dla} \quad k = 0, \ldots, m-1.$$

Stąd, dla dowolnego $x \in (-\infty, x_1]$ otrzymujemy

$$\begin{aligned} S(x) = \begin{cases} S_0(x) & \text{dla } x \in (-\infty, x_0] \\ S_1(x) & \text{dla } x \in (x_0, x_1] \end{cases} &= \\ = S_0(x) + \begin{cases} 0 & \text{dla } x \in (-\infty, x_0] \\ (\beta_m - \alpha_m)(x - x_0)^m & \text{dla } x \in (x_0, x_1] \end{cases} &= \\ = p(x) + a_0(x - x_0)_+^m, & \end{aligned}$$

gdzie $p = S_0 \in \pi_m$ oraz $a_0 = \beta_m - \alpha_m$. Powtarzając powyższe rozumowanie otrzymujemy postać funkcji sklejanej $S$ na coraz szerszych przedziałach $(-\infty, x_2], (-\infty, x_3], \ldots$ Po $n$ krokach otrzymujemy postać (5.1). ∎

**Stwierdzenie 5.2 (postać naturalnej funkcji sklejanej stopnia $2m-1$).** *Dla każdej funkcji $S \in \mathcal{N}_{2m-1}(\triangle)$ istnieją wielomian $p \in \pi_{m-1}$ oraz współczynniki $a_0, \ldots, a_n \in \mathbb{R}$, dla których*

$$S(x) = p(x) + \sum_{j=0}^{n} a_j (x - x_j)_+^{2m-1},$$

*gdzie*

$$\sum_{j=0}^{n} a_j x_j^r = 0, \qquad \textit{dla } r = 0, \ldots, m-1. \tag{5.2}$$

**Dowód**: Ponieważ $\mathcal{N}_{2m-1}(\triangle) \subset \mathcal{S}_{2m-1}(\triangle)$, zatem na podstawie stwierdzenia 5.1 funkcję $S$ możemy zapisać w postaci

$$S(x) = p(x) + \sum_{j=0}^{n} a_j (x - x_j)_+^{2m-1},$$

gdzie $p \in \pi_{2m-1}$. Ponieważ dla $x < x_0$

$$S(x) = p(x) + \sum_{j=0}^{n} a_j (x - x_j)_+^{2m-1} = p(x),$$

zatem z definicji naturalnej funkcji sklejanej wynika, że $p \in \pi_{m-1}$. Podobnie, dla $x > x_n$ funkcja $S$ jest wielomianem postaci

$$S(x) = p(x) + \sum_{j=0}^{n} a_j (x - x_j)^{2m-1};$$

wielomian ten będzie stopnia $m-1$, gdy wszystkie jego współczynniki stojące przy $x^r$ będą zerowe, dla $r = m, \ldots, 2m-1$. Ponieważ

$$\begin{aligned} \sum_{j=0}^{n} a_j (x - x_j)^{2m-1} &= \sum_{j=0}^{n} a_j \left( \sum_{k=0}^{2m-1} \binom{2m-1}{k} x^k (-x_j)^{2m-1-k} \right) = \\ &= \sum_{k=0}^{2m-1} \left( \binom{2m-1}{k} (-1)^{2m-1-k} \sum_{j=0}^{n} a_j x_j^{2m-1-k} \right) x^k, \end{aligned}$$

zatem

$$\sum_{j=0}^{n} a_j x_j^{2m-1-k} = 0$$

dla $k = m, \ldots, 2m-1$. Stąd wynika teza. ∎

W celu wyznaczenia naturalnej funkcji sklejanej $S \in \mathcal{N}_{2m-1}(\triangle)$ musimy obliczyć jej $m+n+1$ współczynników: $m$ współczynników wielomianu $p \in \pi_{m-1}$ oraz $n+1$ wartości $a_0, \ldots, a_n$ spełniających $m$ warunków (5.2). Pozostaje jeszcze $n+1$ wolnych parametrów, które możemy wykorzystać rozwiązując zadanie interpolacyjne.

## 5.2. Interpolacja funkcjami sklejanymi

### Sformułowanie problemu

Dla zadanej funkcji $f : [a,b] \to \mathbb{R}$ oraz parami różnych węzłów $x_0, \ldots, x_n \in [a,b]$ poszukujemy funkcji sklejanej $S$ spełniającej warunki interpolacji

$$S(x_i) = f(x_i), \qquad \text{dla } i = 0, \ldots, n. \tag{5.3}$$

Zauważmy, że z takiego sformułowania problemu nie wynika, że węzły $x_0, \ldots, x_n$ muszą stanowić podział $\triangle$ definiujący funkcję $S$, nie mamy również dodatkowej informacji na temat jej stopnia.

### Istnienie i jedyność rozwiązania

Zachodzi następujące twierdzenie.

**Twierdzenie 5.3 (istnienie i jedyność rozwiązania).** *Niech $\triangle = \{x_0, \ldots, x_n\}$, gdzie $x_0 < \ldots < x_n$. Jeżeli liczba $m \in \mathbb{N}$ spełnia warunek $1 \leqslant m \leqslant n+1$, to dla dowolnej funkcji $f : [x_0, x_n] \to \mathbb{R}$ istnieje dokładnie jedna funkcja $S \in \mathcal{N}_{2m-1}(\triangle)$, taka że*

$$S(x_i) = f(x_i), \qquad \textit{dla } i = 0, \ldots, n.$$

W dowodzie twierdzenia 5.3 wykorzystamy następujący lemat.

**Lemat 5.4.** *Przy powyższych oznaczeniach, niech $[x_0, x_n] \subset (a,b)$, niech $g \in \mathcal{C}^{m-1}_{[a,b]}$ oraz niech $g^{(m)}$ będzie funkcją ciągłą na każdym z przedziałów*

$$(a, x_0), (x_0, x_1), \ldots, (x_{n-1}, x_n), (x_n, b).$$

*Jeżeli $S \in \mathcal{N}_{2m-1}(\triangle)$ jest funkcją postaci*

$$S(x) = p(x) + \sum_{j=0}^{n} a_j (x - x_j)_+^{2m-1},$$

*to*

$$\int_a^b g^{(m)}(x) S^{(m)}(x)\, dx = (-1)^m (2m-1)! \sum_{j=0}^{n} a_j g(x_j).$$

**Dowód lematu**: Całkując przez części otrzymujemy

$$\int_a^b g^{(m)}(x) S^{(m)}(x)\, dx = g^{(m-1)}(x) S^{(m)}(x)\Big|_a^b - \int_a^b g^{(m-1)}(x) S^{(m+1)}(x)\, dx.$$

Ponieważ $S \in \mathcal{N}_{2m-1}(\triangle)$, zatem

$$S^{(m)}(a) = S^{(m)}(b) = 0,$$

a stąd

$$\int_a^b g^{(m)}(x) S^{(m)}(x)\, dx = -\int_a^b g^{(m-1)}(x) S^{(m+1)}(x)\, dx.$$

Powtarzając to rozumowanie $m-1$ razy otrzymujemy

$$\begin{aligned}
&\int_a^b g^{(m)}(x)\, S^{(m)}(x)\, dx = -\int_a^b g^{(m-1)}(x)\, S^{(m+1)}(x)\, dx = \ldots = \\
&= (-1)^{m-1} \int_a^b g'(x)\, S^{(2m-1)}(x)\, dx = (-1)^{m-1} \int_a^{x_0} g'(x)\, S^{(2m-1)}(x)\, dx + \\
&\quad + (-1)^{m-1} \int_{x_0}^{x_n} g'(x)\, S^{(2m-1)}(x)\, dx + (-1)^{m-1} \int_{x_n}^{b} g'(x)\, S^{(2m-1)}(x)\, dx = \\
&= (-1)^{m-1} \int_{x_0}^{x_n} g'(x)\, S^{(2m-1)}(x)\, dx = (-1)^{m-1} \sum_{k=0}^{n-1} \int_{x_k}^{x_{k+1}} g'(x)\, S^{(2m-1)}(x)\, dx.
\end{aligned}$$

Z postaci funkcji $S$ wynika, że dla $x \in (x_k, x_{k+1})$

$$S^{(2m-1)}(x) = (2m-1)!\,(a_0 + \ldots + a_k).$$

Stąd

$$\begin{aligned}
\int_a^b g^{(m)}(x) S^{(m)}(x)\, dx &= (-1)^{m-1}(2m-1)! \sum_{k=0}^{n-1} (\underbrace{a_0 + \ldots + a_k}_{b_k})\,(g(x_{k+1}) - g(x_k)) = \\
&= (-1)^{m-1}(2m-1)! \left( \sum_{k=0}^{n-1} b_k g(x_{k+1}) - \sum_{k=0}^{n-1} b_k g(x_k) \right) = \\
&= (-1)^{m-1}(2m-1)! \left( \sum_{k=0}^{n-1} (b_{k+1} - a_{k+1}) g(x_{k+1}) - \sum_{k=0}^{n-1} b_k g(x_k) \right) = \\
&= (-1)^{m-1}(2m-1)! \left( \sum_{k=1}^{n} b_k g(x_k) - \sum_{k=1}^{n} a_k g(x_k) - \sum_{k=0}^{n-1} b_k g(x_k) \right) = \\
&= (-1)^{m-1}(2m-1)! \left( b_n g(x_n) - \sum_{k=0}^{n} a_k g(x_k) \right).
\end{aligned}$$

Na podstawie warunku (5.2), dla $r = 0$ wnioskujemy, że

$$b_n = a_0 + \ldots + a_n = 0.$$

To kończy dowód. ∎

**Dowód twierdzenia 5.3**: Dowód przeprowadzimy dla $m \geqslant 2$; przypadek $m = 1$ pozostawiam jako proste ćwiczenie.

Warunki interpolacji oraz definicja naturalnej funkcji sklejanej stopnia $2m-1$ prowadzą do następujących zależności:

- dla $i = 0, \ldots, n$

$$p(x_i) + \sum_{j=0}^{i} a_j (x_i - x_j)^{2m-1} = f(x_i), \tag{5.4}$$

- dla $r = 0, \ldots, m-1$

$$\sum_{j=0}^{n} a_j x_j^r = 0, \tag{5.5}$$

które tworzą układ $n+m+1$ równań liniowych o $n+m+1$ niewiadomych. Pokażemy, że macierz tego układu jest nieosobliwa. Zrobimy to wykazując, że jednorodny układ równań liniowych o tej samej macierzy, co rozważany układ niejednorodny, posiada jedynie rozwiązanie zerowe. W tym celu przypuśćmy, że

$$f(x_0) = \ldots = f(x_n) = 0$$

oraz zastosujmy lemat 5.4 przyjmując $g = S$. Wynika z niego, że dla dowolnych $a < x_0$ oraz $b > x_n$

$$\int_a^b \left(S^{(m)}(x)\right)^2 dx = (-1)^m (2m-1)! \sum_{j=0}^{n} a_j S(x_j) =$$
$$= (-1)^m (2m-1)! \sum_{j=0}^{n} a_j f(x_j) = 0.$$

To oznacza, że $S^{(m)} = 0$ prawie wszędzie w $(a, b)$. Ponieważ jednak $S^{(m)} \in \mathcal{C}_{\mathbb{R}}$, zatem $S^{(m)} \equiv 0$ w $(a, b)$. Podobnie, $S^{(m)} \equiv 0$ poza przedziałem $(a, b)$ (funkcja $S$ jest tam wielomianem stopnia co najwyżej $m-1$), zatem $S^{(m)} \equiv 0$ na całej prostej $\mathbb{R}$. To oznacza, że $S \in \pi_{m-1}$. Dodatkowo, na podstawie warunków interpolacji wnioskujemy, że

$$S(x_i) = f(x_i) = 0 \tag{5.6}$$

dla $i = 0, \ldots, n$. Z założenia $m \leqslant n+1$ wynika, że stopień wielomianu $S$ jest mniejszy niż liczba jego miejsc zerowych $x_0, \ldots, x_n$. Wynika stąd, że $S \equiv 0$ co oznacza, że macierz układu (5.4)-(5.5) jest nieosobliwa. To kończy dowód. ∎

## 5.3. Kubiczne funkcje sklejane

Funkcję sklejaną (zwykłą lub naturalną) stopnia 3 nazywamy kubiczną funkcją sklejaną (zwykłą lub naturalną). Naturalna kubiczna funkcja sklejana dla podziału $\triangle = \{x_0, \ldots, x_n\}$ spełnia więc następujące warunki

**(a)** $S|_{(x_i, x_{i+1})} \in \pi_3$, dla $i = 0, \ldots, n-1$;
**(b)** $S|_{(-\infty, x_0)}, S|_{(x_n, +\infty)} \in \pi_1$;
**(c)** $S \in \mathcal{C}^2_{\mathbb{R}}$.

Rozważmy dodatkowo warunek interpolacji

$$S(x_i) = y_i, \qquad \text{dla } i = 0, \ldots, n \tag{5.7}$$

gdzie $y_0, \ldots, y_n \in \mathbb{R}$ to dowolne, zadane wartości. Jak wynika z twierdzenia 5.3, istnieje dokładnie jedna naturalna kubiczna funkcja sklejana spełniająca warunek (5.7). Pokażemy teraz metodę jej wyznaczania.

### 5.3.1. Wyznaczanie interpolacyjnej naturalnej funkcji sklejanej stopnia 3 ☠

Niech, dla $i = 0, \ldots, n-1$ oraz $x \in [x_i, x_{i+1})$,

$$S(x) = a_i + b_i (x - x_i) + c_i (x - x_i)^2 + d_i (x - x_i)^3.$$

Ciągłość funkcji $S''$ w punkcie $x_{i+1}$ oznacza, że $S''_+(x_{i+1}) = S''_-(x_{i+1})$, czyli

$$2c_{i+1} = 2c_i + 6d_i (x_{i+1} - x_i),$$

skąd wynika, że

$$d_i = \frac{c_{i+1} - c_i}{3h_i}, \tag{5.8}$$

gdzie $h_i = x_{i+1} - x_i$. Ciągłość funkcji $S$ w punkcie $x_{i+1}$ oznacza, że $S_+(x_{i+1}) = S_-(x_{i+1})$, czyli

$$a_{i+1} = y_{i+1} = a_i + b_i h_i + c_i h_i^2 + d_i h_i^3.$$

Stąd

$$y_{i+1} = y_i + b_i h_i + c_i h_i^2 + \frac{1}{3}(c_{i+1} - c_i) h_i^2$$

lub równoważnie

$$b_i = \frac{y_{i+1} - y_i}{h_i} - \frac{h_i}{3}(c_{i+1} + 2c_i). \tag{5.9}$$

Ciągłość funkcji $S'$ w punkcie $x_{i+1}$ oznacza, że $S'_+(x_{i+1}) = S'_-(x_{i+1})$, czyli

$$b_{i+1} = b_i + 2c_i h_i + 3d_i h_i^2.$$

Uwzględniając (5.8) oraz (5.9) wynika stąd równość

$$\begin{aligned} \frac{y_{i+2}-y_{i+1}}{h_{i+1}} - \frac{h_{i+1}}{3}(c_{i+2}+2c_{i+1}) = \\ = \frac{y_{i+1}-y_i}{h_i} - \frac{h_i}{3}(c_{i+1}+2c_i) + 2c_i h_i + h_i(c_{i+1}-c_i). \end{aligned} \tag{5.10}$$

Przyjmijmy $c_i^* = \frac{1}{3}c_i$. Wówczas (5.10) możemy zapisać jako

$$-c_{i+2}^* h_{i+1} - 2c_{i+1}^*(h_{i+1}+h_i) - c_i^* h_i = \frac{y_{i+1}-y_i}{h_i} - \frac{y_{i+2}-y_{i+1}}{h_{i+1}}.$$

Dzieląc stronami przez $-(h_{i+1}+h_i)$ otrzymujemy dalej

$$\frac{h_{i+1}}{h_{i+1}+h_i}c_{i+2}^* + 2c_{i+1}^* + \frac{h_i}{h_{i+1}+h_i}c_i^* = -\frac{\frac{y_{i+1}-y_i}{h_i} - \frac{y_{i+2}-y_{i+1}}{h_{i+1}}}{h_{i+1}+h_i}.$$

Niech

$$u_{i+1} = \frac{h_{i+1}}{h_{i+1}+h_i}, \quad w_{i+1} = \frac{h_i}{h_{i+1}+h_i}, \quad v_{i+1} = -\frac{\frac{y_{i+1}-y_i}{h_i} - \frac{y_{i+2}-y_{i+1}}{h_{i+1}}}{h_{i+1}+h_i}.$$

Mamy więc, dla $i = 0, \ldots, n-2$,

$$u_{i+1}c_{i+2}^* + 2c_{i+1}^* + w_{i+1}c_i^* = v_{i+1}. \tag{5.11}$$

Z warunku

$$S''(x_0) = S''(x_n) = 0$$

wynika, że[1]

$$c_0^* = c_n^* = 0. \tag{5.12}$$

Zależność (5.11) możemy zatem zapisać w postaci liniowego układu równań ze względu na wektor niewiadomych $\left(c_1^*, \ldots, c_{n-1}^*\right)^T$:

$$\begin{pmatrix} 2 & u_1 & 0 & \cdots & 0 \\ w_2 & 2 & u_2 & \ddots & \vdots \\ 0 & w_3 & \ddots & \ddots & 0 \\ \vdots & \ddots & \ddots & \ddots & u_{n-2} \\ 0 & \cdots & 0 & w_{n-1} & 2 \end{pmatrix} \begin{pmatrix} c_1^* \\ c_2^* \\ \vdots \\ c_{n-1}^* \end{pmatrix} = \begin{pmatrix} v_1 \\ v_2 \\ \vdots \\ v_{n-1} \end{pmatrix}. \tag{5.13}$$

Ponieważ

$$1 = w_i + u_i < 2,$$

zatem macierz tego układu jest macierzą o silnie dominującej przekątnej i jako taka jest macierzą nieosobliwą.[2] Metody rozwiązywania układów liniowych będą treścią dalszych rozdziałów; w tym miejscu zanotujmy jedynie, że istnieją numerycznie poprawne algorytmy rozwiązywania układu (5.13) kosztem $\mathcal{O}(n)$ operacji arytmetycznych. Znając wartości $c_i^*$ wyliczamy kolejno $c_i$ oraz $b_i$ i $d_i$.

[1] Tylko wtedy możemy skleić wielomian z przestrzeni $\pi_1$ z wielomianem z przestrzeni $\pi_3$ w ten sposób, aby sklejenie to było klasy $\mathcal{C}^2_{\mathbb{R}}$.

[2] Będzie to prosty wniosek z twierdzenia Gerszgorina przedstawionego w dalszej części wykładu.

### 5.3.2. Twierdzenie o błędzie interpolacji dla naturalnej kubicznej funkcji sklejanej ☠

Przypuśćmy, że warunki interpolacji (5.7) pochodzą od pewnej funkcji $f : [x_0, x_n] \to \mathbb{R}$, tj. $y_i = f(x_i)$, dla $i = 0, \ldots, n$. Spróbujemy teraz oszacować błąd interpolacji, tj.

$$\|f - S\|_{[x_0,x_n]}.$$

Dla $x \in [x_i, x_{i+1})$, na podstawie powyższych rozważań, mamy

$$\begin{aligned} S(x) - f(x) = f(x_i) + b_i(x - x_i) + c_i(x - x_i)^2 + d_i(x - x_i)^3 - f(x) = \\ = f(x_i) - f(x) + \left( \frac{f(x_{i+1}) - f(x_i)}{h_i} - h_i\left(c^*_{i+1} + 2c^*_i\right) \right)(x - x_i) + \\ + 3c^*_i(x - x_i)^2 + \frac{c^*_{i+1} - c^*_i}{h_i}(x - x_i)^3. \end{aligned}$$

Uwzględniając, że $|x - x_i| \leqslant h_i$ otrzymujemy stąd

$$|S(x) - f(x)| \leqslant \left| f(x_i) - f(x) + \frac{f(x_{i+1}) - f(x_i)}{h_i}(x - x_i) \right| + 8h_i^2 \max_i |c^*_i|. \tag{5.14}$$

Przypuśćmy, że $f \in \mathcal{C}^2_{[x_0,x_n]}$; możemy wówczas zapisać

$$f(x) = f(x_i) + f'(x_i)(x - x_i) + \frac{1}{2}f''(\xi_i)(x - x_i)^2$$

oraz

$$\begin{aligned} \frac{f(x_{i+1}) - f(x_i)}{h_i}(x - x_i) = \frac{f(x_i) + f'(x_i)h_i + \frac{1}{2}f''(\eta_i)h_i^2 - f(x_i)}{h_i}(x - x_i) = \\ = (f'(x_i) + \frac{1}{2}f''(\eta_i)h_i)(x - x_i), \end{aligned}$$

dla pewnych $\xi_i, \eta_i \in (x_i, x_{i+1})$. Stąd

$$\begin{aligned} |f(x_i) - f(x) + \frac{f(x_{i+1}) - f(x_i)}{h_i}(x - x_i)| = \frac{1}{2}|f''(\eta_i)h_i(x - x_i) - f''(\xi_i)(x - x_i)^2| \leqslant \\ \leqslant h_i^2 \left\| f'' \right\|_{[x_0,x_n]}. \end{aligned}$$

Stąd oraz z (5.14) otrzymujemy oszacowanie

$$|S(x) - f(x)| \leqslant h_i^2 \left\| f'' \right\|_{[x_0,x_n]} + 8h_i^2 \max_i |c^*_i|.$$

Pokażemy jeszcze, że

$$\max_i |c^*_i| \leqslant \frac{1}{2} \left\| f'' \right\|_{[x_0,x_n]}. \tag{5.15}$$

W tym celu wróćmy do zależności (5.11)

$$u_i c^*_{i+1} + 2c^*_i + w_i c^*_{i-1} = v_i$$

z której wynika, że

$$\begin{aligned} |v_i| &\geqslant 2|c^*_i| - \left| w_i c^*_{i-1} + u_i c^*_{i+1} \right| \geqslant \\ &\geqslant 2|c^*_i| - \left( w_i \left| c^*_{i-1} \right| + u_i \left| c^*_{i+1} \right| \right) \geqslant \\ &\geqslant 2|c^*_i| - (w_i + u_i) \max_i |c^*_i| = 2|c^*_i| - \max_i |c^*_i| \end{aligned}$$

czyli

$$\max_i |v_i| \geqslant \max_i \left( 2|c^*_i| - \max_i |c^*_i| \right) = \max_i |c^*_i|. \tag{5.16}$$

Zauważmy na koniec, że

$$v_i = \frac{\frac{f(x_{i+1})-f(x_i)}{x_{i+1}-x_i} - \frac{f(x_i)-f(x_{i-1})}{x_i-x_{i-1}}}{x_{i+1}-x_{i-1}} = \frac{f(x_i, x_{i+1}) - f(x_{i-1}, x_i)}{x_{i+1}-x_{i-1}} = \\ = f(x_{i-1}, x_i, x_{i+1}),$$

skąd, na podstawie własności różnic dzielonych (zob. uwagę 3.2 na str. 24) otrzymujemy

$$v_i = \frac{1}{2} f''(\delta_i)$$

dla pewnego $\delta_i \in (x_{i-1}, x_{i+1})$. Stąd oraz z (5.16) wynika (5.15). Wykazaliśmy tym samym następujące twierdzenie.

**Twierdzenie 5.5.** *Niech $f \in \mathcal{C}^2_{[x_0,x_n]}$ oraz niech $S \in \mathcal{N}_3(\{x_0, \ldots, x_n\})$ będzie interpolacyjną funkcją sklejaną dla $f$, tj.*

$$S(x_i) = f(x_i), \qquad \text{dla } i = 0, \ldots, n.$$

*Wówczas*

$$\|f - S\|_{[x_0,x_n]} \leqslant 5 \left\|f''\right\|_{[x_0,x_n]} \cdot \max_i (x_{i+1} - x_i)^2.$$

### 5.3.3. Zbieżność procesu interpolacji ☠

Przypomnijmy, że w przypadku interpolacji wielomianowej były problemy ze zbieżnością ciągu wielomianów interpolacyjnych do funkcji interpolowanej – zbieżności tej nie gwarantowała nawet bardzo wysoka regularność funkcji $f$. W przypadku interpolacji funkcjami sklejanymi problemów takich nie ma.

Z twierdzenia 5.5 wynika, że jeżeli ciąg podziałów $\triangle_n = \{x_0^{(n)}, x_1^{(n)}, \ldots, x_n^{(n)}\}$,

$$a \leqslant x_0^{(n)} < x_1^{(n)} < \ldots < x_n^{(n)} \leqslant b,$$

spełnia warunek

$$\lim_{n\to\infty} \delta(\triangle_n) = 0,$$

gdzie

$$\delta(\triangle_n) = \max_{i=0}^{n-1} \left(x_{i+1}^{(n)} - x_i^{(n)}\right),$$

to dla dowolnej funkcji $f \in \mathcal{C}^2_{[a,b]}$ ciąg interpolujących ją kubicznych naturalnych funkcji sklejanych $S_n \in \mathcal{N}_3(\triangle_n)$ jest zbieżny jednostajnie do $f$, tj.

$$\lim_{n\to\infty} \|f - S_n\|_{[a,b]} = 0.$$

## Rozdział 6

# Interpolacja trygonometryczna

Niech $f : \mathbb{R} \to \mathbb{C}$ będzie funkcją okresową o okresie $T > 0$, tj.

$$\forall x \in \mathbb{R} : \quad f(x+T) = f(x)$$

oraz niech, dla $M > 0$, $F(x) = f(xT/M)$. Dla dowolnego $x \in \mathbb{R}$ mamy

$$F(x+M) = f\Big((x+M)\frac{T}{M}\Big) = f\Big(\frac{xT}{M}+T\Big) = f\Big(\frac{xT}{M}\Big) = F(x),$$

zatem rozważając problem interpolacji funkcji okresowych możemy ograniczyć się do funkcji okresowych o dowolnym ustalonym okresie $T$; dla wygody zapisu przyjmujemy $T = 2\pi$.

### 6.1. Sformułowanie problemu, istnienie i jedyność rozwiązania

**Sformułowanie problemu**

Dla zadanej funkcji $f : [0, 2\pi) \to \mathbb{C}$ oraz parami różnych węzłów $x_0, \ldots, x_n \in [0, 2\pi)$ poszukujemy wielomianu trygonometrycznego $t_n$ stopnia co najwyżej $n$

$$t_n(x) = \sum_{k=0}^{n} c_k e^{ikx}$$

spełniającego warunki interpolacji

$$t_n(x_i) = f(x_i), \qquad \text{dla } i = 0, \ldots, n. \tag{6.1}$$

**Istnienie i jedyność rozwiązania**

Warunki interpolacji (6.1) prowadzą do układu $n+1$ równań liniowych o $n+1$ niewiadomych $c_0, \ldots, c_n$:

$$\begin{cases} c_0 + c_1 e^{ix_0} + c_2 e^{i2x_0} + \ldots + c_n e^{inx_0} = f(x_0) \\ \qquad\qquad\qquad\qquad \vdots \\ c_0 + c_1 e^{ix_n} + c_2 e^{i2x_n} + \ldots + c_n e^{inx_n} = f(x_n) \end{cases}.$$

Macierz $A \in \mathbb{C}^{(n+1)\times(n+1)}$ tego układu ma postać

$$A = \begin{pmatrix} 1 & e^{ix_0} & \cdots & e^{inx_0} \\ \vdots & \vdots & \cdots & \vdots \\ 1 & e^{ix_n} & \cdots & e^{inx_n} \end{pmatrix},$$

jest to macierz Vandermonde'a, której wyznacznik, na podstawie przyjętych założeń dotyczących węzłów, jest niezerowy. To oznacza, że problem interpolacji trygonometrycznej posiada rozwiązanie i jest ono wyznaczone jednoznacznie.

## 6.2. Postać rozwiązania dla węzłów równoodległych

Przypuśćmy, że węzły $x_0, \ldots, x_n$ są równomiernie rozłożone w przedziale $[0, 2\pi)$, tj.

$$x_k = \frac{2k\pi}{n+1}, \qquad \text{dla } k = 0, \ldots, n. \tag{6.2}$$

Wówczas współczynniki $c_0, \ldots, c_n$ będące rozwiązaniem układu (6.1) można łatwo wyznaczyć.

**Twierdzenie 6.1.** *Współczynniki $c_0, \ldots, c_n$ wielomianu trygonometrycznego*

$$t_n(x) = \sum_{k=0}^{n} c_k e^{ikx}$$

*interpolującego funkcję $f$ w węzłach równoodległych $x_k = \frac{2k\pi}{n+1}$ (dla $k = 0, \ldots, n$) mają postać*

$$c_k = \frac{1}{n+1} \sum_{j=0}^{n} f(x_j)\, e^{-ikx_j}, \qquad \textit{dla } k = 0, \ldots, n. \tag{6.3}$$

**Dowód**: Niech $\mathcal{F} = \{f : \{x_0, \ldots, x_n\} \to \mathbb{C}\}$ oraz niech

$$\langle f, g \rangle = \sum_{k=0}^{n} f(x_k)\, \overline{g(x_k)}. \tag{6.4}$$

Odwzorowanie $\langle \cdot, \cdot \rangle$ jest iloczynem skalarnym w $\mathcal{F}$. Niech $\varphi_0, \ldots, \varphi_n$ będą funkcjami postaci

$$\varphi_k(x) = e^{ikx}.$$

Wówczas

$$\langle \varphi_m, \varphi_l \rangle = \sum_{k=0}^{n} \varphi_m(x_k)\, \overline{\varphi_l(x_k)} = \sum_{k=0}^{n} e^{imx_k} e^{-ilx_k} = \sum_{k=0}^{n} e^{i(m-l)x_k}.$$

Stąd

- dla $m = l$

$$\langle \varphi_m, \varphi_m \rangle = n + 1,$$

- dla $m \neq l$

$$\langle \varphi_m, \varphi_l \rangle = \sum_{k=0}^{n} e^{i(m-l)\frac{2k\pi}{n+1}} = \sum_{k=0}^{n} \left( e^{\frac{i2\pi(m-l)}{n+1}} \right)^k = \frac{1 - e^{i2\pi(m-l)}}{1 - e^{\frac{i2\pi(m-l)}{n+1}}} = 0.$$

Funkcje $\varphi_0, \ldots, \varphi_n$ stanowią zatem układ ortogonalny w przestrzeni $\mathcal{F}$. Stąd, dla ustalonego $k = 0, \ldots, n$,

$$\begin{aligned} \langle t_n, \varphi_k \rangle &= \left\langle \sum_{j=0}^{n} c_j \varphi_j, \varphi_k \right\rangle = \sum_{j=0}^{n} c_j \langle \varphi_j, \varphi_k \rangle = \\ &= c_k \langle \varphi_k, \varphi_k \rangle + \sum_{j \neq k} c_j \langle \varphi_j, \varphi_k \rangle = (n+1)\, c_k \end{aligned} \tag{6.5}$$

oraz

$$\langle t_n, \varphi_k \rangle = \sum_{j=0}^{n} t_n(x_j)\, \overline{\varphi_k(x_j)} = \sum_{j=0}^{n} f(x_j)\, e^{-ikx_j}. \tag{6.6}$$

Porównując (6.5) oraz (6.6) otrzymujemy równość kończącą dowód. ∎

Wyznaczanie współczynników $c_0, \ldots, c_n$ określonych wzorami (6.3) to tzw. *dyskretna analiza Fouriera* (współczynniki te równe są współczynnikom Fouriera funkcji $f$ względem dyskretnego iloczynu skalarnego (6.4)). Metodą $FFT$ (ang. fast Fourier transformation)[1] współczynniki te można wyznaczyć kosztem $\mathcal{O}(n \log n)$ operacji arytmetycznych.

## 6.3. Rzeczywiste trygonometryczne wielomiany interpolacyjne dla węzłów równoodległych

W przypadku interpolowania funkcji okresowych o wartościach rzeczywistych rozwiązania problemu interpolacji najczęściej poszukuje się w przestrzeni $\mathcal{T}_k$ rzeczywistych wielomianów trygonometrycznych stopnia co najwyżej $k$, gdzie

$$\mathcal{T}_{2n} = \operatorname{lin}\{1, \sin x, \cos x, \sin 2x, \cos 2x, \ldots, \cos nx, \sin nx\}$$

oraz

$$\mathcal{T}_{2n+1} = \operatorname{lin}\{1, \sin x, \cos x, \sin 2x, \cos 2x, \ldots, \cos nx, \sin nx, \cos(n+1)x\}.$$

Zachodzi następujące twierdzenie.

**Twierdzenie 6.2.** *Współczynniki wielomianu trygonometrycznego*

$$t_{2n}(x) = \frac{a_0}{2} + \sum_{k=1}^{n} (a_k \cos kx + b_k \sin kx)$$

*interpolującego funkcję $f$ w węzłach*

$$x_k = \frac{2k\pi}{2n+1}, \qquad dla\ k = 0, \ldots, 2n$$

*wyrażają się wzorami*

$$\begin{aligned} a_k &= \frac{2}{2n+1} \sum_{j=0}^{2n} f(x_j) \cos(kx_j), \qquad dla\ k = 0, \ldots, n, \\ b_k &= \frac{2}{2n+1} \sum_{j=0}^{2n} f(x_j) \sin(kx_j), \qquad dla\ k = 1, \ldots, n. \end{aligned} \tag{6.7}$$

**Dowód**: Dowód jest podobny do dowodu twierdzenia 6.1. Niech

$$\langle f, g \rangle = \sum_{j=0}^{2n} f(x_j)\, g(x_j)$$

oraz niech

$$\cos_k(x) = \cos kx, \qquad \text{dla } k = 0, \ldots, n$$

oraz

$$\sin_k(x) = \sin kx, \qquad \text{dla } k = 1, \ldots, n.$$

Łatwo wykazać, że

$$\langle \cos_k, \cos_l \rangle = \begin{cases} 2n+1 & \text{dla } k = l = 0 \\ \frac{2n+1}{2} & \text{dla } k = l \geqslant 1 \\ 0 & \text{dla } k \neq l \end{cases}, \tag{6.8}$$

[1] Algorytm $FFT$ ma liczne zastosowania, m.in. w procesach cyfrowego przekształcania sygnałów oraz kompresji danych audio-wideo do popularnych formatów takich jak jpeg czy mp3. Jest on również często wykorzystywany do konstruowania algorytmów rozwiązujących pozornie zupełnie odległe zadania; przykładem może być tu algorytm szybiego mnożenia wielomianów (zob. przykład 2.3 w J. i M. Jankowscy, *Przegląd metod i algorytmów numerycznych. Część 1.*, WNT, Warszawa 1981).

$$\langle \sin_k, \sin_l \rangle = \begin{cases} \frac{2n+1}{2} & \text{dla } k = l \geqslant 1 \\ 0 & \text{dla } k \neq l \end{cases}$$

oraz

$$\langle \sin_k, \cos_l \rangle = 0 \quad \text{dla } k = 1, \ldots, n;\ l = 0, \ldots, n.$$

Wykażemy równość (6.8); pozostałe dowodzimy analogicznie. W tym celu przypomnijmy wynikające ze wzoru Eulera zależności

$$\cos x = \frac{e^{ix} + e^{-ix}}{2} \quad \text{oraz} \quad \sin x = \frac{e^{ix} - e^{-ix}}{2i}.$$

Stąd

$$\begin{aligned} \langle \cos_k, \cos_l \rangle &= \sum_{j=0}^{2n} \cos(kx_j) \cos(lx_j) = \sum_{j=0}^{2n} \frac{e^{ikx_j} + e^{-ikx_j}}{2} \frac{e^{ilx_j} + e^{-ilx_j}}{2} = \\ &= \frac{1}{4} \sum_{j=0}^{2n} \left( e^{i(k+l)x_j} + e^{-i(k+l)x_j} + e^{i(k-l)x_j} + e^{-i(k-l)x_j} \right) = \\ &= \frac{1}{2} \operatorname{Re} \sum_{j=0}^{2n} e^{i(k+l)x_j} + \frac{1}{2} \operatorname{Re} \sum_{j=0}^{2n} e^{i(k-l)x_j}. \end{aligned}$$

Dla $k + l \neq 0$ mamy

$$\sum_{j=0}^{2n} e^{i(k+l)x_j} = \sum_{j=0}^{2n} e^{i(k+l)\frac{2j\pi}{2n+1}} = \sum_{j=0}^{2n} \left( e^{\frac{i(k+l)2\pi}{2n+1}} \right)^j = \frac{1 - e^{i(k+l)2\pi}}{1 - e^{\frac{i(k+l)2\pi}{2n+1}}} = 0$$

oraz dla $k \neq l$

$$\sum_{j=0}^{2n} e^{i(k-l)x_j} = \sum_{j=0}^{2n} e^{i(k-l)\frac{2j\pi}{2n+1}} = \sum_{j=0}^{2n} \left( e^{\frac{i(k-l)2\pi}{2n+1}} \right)^j = \frac{1 - e^{i(k-l)2\pi}}{1 - e^{\frac{i(k-l)2\pi}{2n+1}}} = 0.$$

Z kolei dla $k = l = 0$

$$\langle \cos_0, \cos_0 \rangle = \sum_{j=0}^{2n} 1 = 2n + 1$$

oraz dla $k = l \geqslant 1$

$$\begin{aligned} \langle \cos_k, \cos_k \rangle &= \sum_{j=0}^{2n} \cos^2(kx_j) = \frac{1}{4} \sum_{j=0}^{2n} \left( e^{ikx_j} + e^{-ikx_j} \right)^2 = \\ &= \frac{1}{4} \sum_{j=0}^{2n} \left( e^{i2kx_j} + e^{-i2kx_j} + 2 \right) = \frac{2n+1}{2} + \frac{1}{2} \operatorname{Re} \underbrace{\sum_{j=0}^{2n} e^{i2kx_j}}_{=0} = \frac{2n+1}{2}. \end{aligned}$$

Stąd wynika równość (6.8).

Obliczmy teraz $\langle t_{2n}, \cos_l \rangle$, dla $l = 0, \ldots, n$. Z jednej strony

$$\langle t_{2n}, \cos_l \rangle = \sum_{j=0}^{2n} t_{2n}(x_j) \cos(lx_j) = \sum_{j=0}^{2n} f(x_j) \cos(lx_j) \tag{6.9}$$

a z drugiej

$$\begin{aligned} \langle t_{2n}, \cos_l \rangle &= \left\langle \frac{a_0}{2} + \sum_{k=1}^{n} (a_k \cos_k + b_k \sin_k), \cos_l \right\rangle = \\ &= \frac{a_0}{2} \langle \cos_0, \cos_l \rangle + \sum_{k=1}^{n} a_k \langle \cos_k, \cos_l \rangle + \sum_{k=1}^{n} b_k \langle \sin_k, \cos_l \rangle. \end{aligned}$$

Stąd, dla $l = 0$

$$\langle t_{2n}, \cos_0 \rangle = \frac{a_0}{2} \langle \cos_0, \cos_0 \rangle = a_0 \frac{2n+1}{2} \tag{6.10}$$

oraz dla $l \geqslant 1$

$$\langle t_{2n}, \cos_l \rangle = \sum_{k=1}^{n} a_k \langle \cos_k, \cos_l \rangle = a_l \langle \cos_l, \cos_l \rangle + \sum_{k \neq l} a_k \langle \cos_k, \cos_l \rangle = a_l \frac{2n+1}{2}. \tag{6.11}$$

Porównując (6.9) z (6.10) i (6.11) otrzymujemy dla $l = 0, \ldots, n$

$$a_l = \frac{2}{2n+1} \sum_{j=0}^{2n} f(x_j) \cos(l x_j).$$

Postać współczynników $b_1, \ldots, b_n$ wyprowadzamy analogicznie. ■

**Uwaga 6.1.** *Twierdzenie 6.2 podaje przepis na współczynniki wielomianu trygonometrycznego stopnia $2n$ interpolującego funkcję $f$ w przypadku, gdy liczba węzłów jest nieparzysta i równa $2n+1$. Rzeczywisty wielomian trygonometryczny stopnia $2n+1$ będący rozwiązaniem zadania interpolacyjnego w przypadku parzystej liczby węzłów równej $2n+2$ ma postać*

$$t_{2n+1}(x) = \frac{a_0}{2} + \sum_{k=1}^{n} (a_k \cos kx + b_k \sin kx) + \frac{a_{n+1}}{2} \cos(n+1)x;$$

*jego współczynniki $a_0, \ldots, a_{n+1}$ oraz $b_1, \ldots, b_n$ określone są wzorami*

$$\begin{aligned} a_k &= \frac{1}{n+1} \sum_{j=0}^{2n+1} f(x_j) \cos(k x_j), \qquad \text{dla } k = 0, \ldots, n+1, \\ b_k &= \frac{1}{n+1} \sum_{j=0}^{2n+1} f(x_j) \sin(k x_j), \qquad \text{dla } k = 1, \ldots, n. \end{aligned} \tag{6.12}$$

**Uwaga 6.2.** *Jeżeli funkcja $f$ ma wartości rzeczywiste, to opisane zależnością* (6.7) *współczynniki również są rzeczywiste.*

## Rozdział 7

# Aproksymacja w przestrzeni unormowanej

### 7.1. Sformułowanie problemu, istnienie i jedyność rozwiązania

**Sformułowanie problemu**

Niech $(X, \|\cdot\|)$ będzie przestrzenią unormowaną, niech $V \subset X$ będzie niepustą podprzestrzenią liniową oraz niech $f \in X$ będzie dowolnym ustalonym elementem. Problem aproksymacji polega na znalezieniu takiego elementu $h^* \in V$, który spośród wszystkich elementów przestrzeni $V$ najlepiej aproksymuje (czyli przybliża) element $f$, tj.

$$\forall h \in V: \quad \|f - h^*\| \leqslant \|f - h\|.$$

Element $h^*$, o ile istnieje, nazywamy *elementem optymalnym dla elementu $f$ w przestrzeni $V$*, liczbę

$$\varepsilon_V(f) = \|f - h^*\| = \inf_{h \in V} \|f - h\|$$

nazywamy *błędem aproksymacji.*

**Istnienie rozwiązania**

Przy braku dodatkowych założeń element optymalny może nie istnieć.

**Przykład 7.1 (brak elementu optymalnego).** *Niech $X = \mathcal{C}_{[0,1]}$, niech $\|\cdot\| = \|\cdot\|_{[0,1]}$ oraz niech*

$$V = \bigcup_n \pi_n$$

*będzie przestrzenią wszystkich wielomianów. Pokażemy, że dla funkcji*

$$f(x) = e^x = \sum_{k=0}^{\infty} \frac{1}{k!} x^k$$

*nie istnieje element optymalny w $V$.*

*Dla dowodu nie wprost przypuśćmy, że istnieje wielomian $h^* \in V$, dla którego warunek*

$$\varepsilon_V(f) = \|f - h^*\|_{[0,1]} \leqslant \|f - h\|_{[0,1]}$$

*jest spełniony dla wszystkich $h \in V$. Jest jasne, że $\varepsilon_V(f) > 0$. Istnieje więc na tyle duża liczba $n \in \mathbb{N}$, aby*

$$\frac{e}{(n+1)!} < \varepsilon_V(f).$$

*Niech $h_n \in V$ będzie wielomianem postaci*

$$h_n(x) = \sum_{k=0}^{n} \frac{1}{k!} x^k.$$

*Wówczas, na podstawie twierdzenia Taylora, mamy:*
$\forall x \in [0,1]\ \exists \xi \in (0,1)$ :

$$f(x) - h_n(x) = \frac{e^\xi}{(n+1)!} x^{n+1}.$$

*Stąd*

$$\|f - h_n\|_{[0,1]} = \max_{x\in[0,1]} \left| \frac{e^\xi}{(n+1)!} x^{n+1} \right| \leqslant \frac{e}{(n+1)!} < \varepsilon_V(f).$$

*To przeczy optymalności $h^*$.*

Zachodzi następujące twierdzenie.

**Twierdzenie 7.1.** *Jeżeli, przy powyższych oznaczeniach,* $\dim V < \infty$*, to dla każdego elementu $f \in X$ istnieje element optymalny w $V$.*

**Dowód**: Ustalmy dowolny element $f \in X$. Z własności normy oraz na podstawie definicji elementu optymalnego dla $f$ wynika, że

$$\|h^*\| - \|f\| \leqslant \|f - h^*\| \leqslant \|f\|$$

a stąd

$$\|h^*\| \leqslant 2\|f\|.$$

Oznacza to, że poszukiwania elementu optymalnego dla $f$ możemy ograniczyć do kuli

$$K_f = \{h \in V : \|h\| \leqslant 2\|f\|\}.$$

Kula $K_f$ jest zbiorem zwartym, zatem określona na niej funkcja ciągła

$$K_f \ni h \to \|f - h\|$$

osiąga swoje kresy. W szczególności, istnieje element $h^* \in K_f$, dla którego

$$\|f - h^*\| = \inf_{h\in K_f} \|f - h\| = \inf_{h\in V} \|f - h\|.$$

To kończy dowód. ∎

**Jedyność rozwiązania**

Jak wiemy, istnienie elementu optymalnego wynika z bardzo ogólnej własności podprzestrzeni $V$. Nie ma niestety twierdzenia mówiącego o jedyności elementu optymalnego w ogólnym przypadku; jego jedyność zależy od normy przestrzeni.

**Przykład 7.2 (brak jedyności elementu optymalnego).** *Niech $X = \mathcal{C}_{[0,\pi/2]}$, niech $\|\cdot\| = \|\cdot\|_{[0,\pi/2]}$ oraz niech*

$$V = \{h \in \pi_1 : h(0) = 0\} = \{x \to \lambda x : \lambda \in \mathbb{R}\}.$$

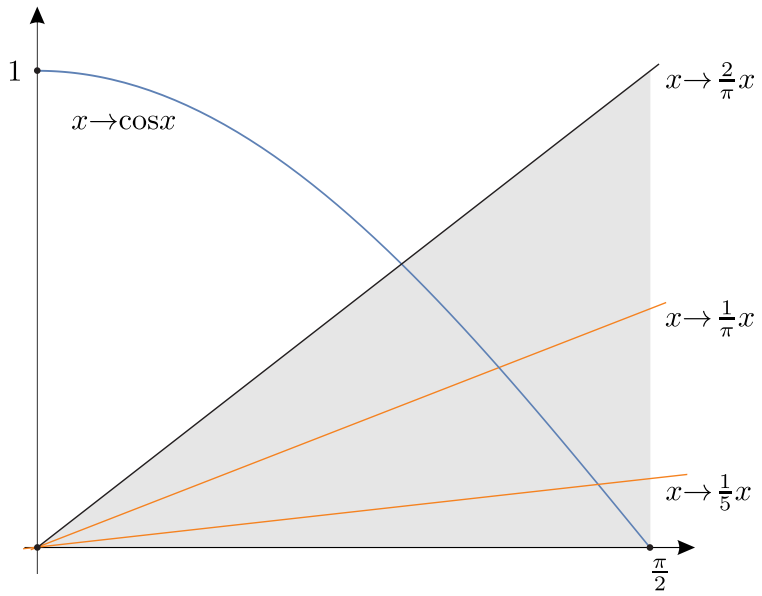
*Ponieważ* $\dim V = 1 < \infty$*, zatem dla każdej funkcji $f \in X$ istnieje element optymalny $h^* \in V$. Niech $f(x) = \cos x$. Wówczas*

$$\|f - h^*\|_{[0,\pi/2]} = \max_{x\in[0,\pi/2]} |f(x) - h^*(x)| \geqslant |f(0) - h^*(0)| = 1.$$

*Łatwo teraz zauważyć (zob. rysunek 7.1), że dla dowolnej wartości $\lambda \in [0, 2/\pi]$ wielomian $h_\lambda(x) = \lambda x$ spełnia warunek*

$$\|f - h_\lambda\|_{[0,\pi/2]} = 1.$$

*To oznacza, że element optymalny dla funkcji $f$ w przestrzeni $V$ nie jest wyznaczony jednoznacznie (w rozważanym przykładzie zbiór elementów optymalnych jest nieprzeliczalny).*




Rysunek 7.1.

**Przykład 7.3.** *Rozważmy ponownie przestrzeń $X$ oraz podprzestrzeń $V$ takie jak w poprzednim przykładzie. Tym razem wyznaczymy element optymalny dla funkcji $f(x) = \cos x$ w sensie normy przestrzeni $\mathcal{L}_2$, tj.*

$$\|f\| = \sqrt{\int_0^{\pi/2} f^2(x)\, dx}.$$

*Dla dowolnego elementu $h \in V$, $h(x) = \lambda x$, mamy*

$$\|f - h\|^2 = \int_0^{\pi/2} (f(x) - h(x))^2\, dx = \int_0^{\pi/2} (\cos x - \lambda x)^2\, dx =$$
$$= \lambda^2 \int_0^{\pi/2} x^2 dx - 2\lambda \int_0^{\pi/2} x \cos x dx + \int_0^{\pi/2} \cos^2 x dx. \tag{7.1}$$

*Jest jasne, że traktowany jako funkcja zmiennej $\lambda$ trójmian kwadratowy (7.1) posiada dokładnie jedno minimum globalne osiągane w punkcie*

$$\lambda^* = \frac{\int_0^{\pi/2} x \cos x dx}{\int_0^{\pi/2} x^2 dx} = \frac{12\pi - 24}{\pi^3}.$$

*Oznacza to, że w rozważanym przypadku funkcja $f(x) = \cos x$ posiada dokładnie jeden element optymalny, jest nim*

$$h^*(x) = \frac{12\pi - 24}{\pi^3} x.$$

## 7.2. Przestrzenie silnie unormowane

**Definicja 7.1 (przestrzeń silnie unormowana).** *Przestrzeń $(X, \|\cdot\|)$ nazywamy silnie unormowaną, jeżeli spełnia następujący warunek*

$$\forall f, g \in X: \qquad f \neq g, \|f\| = \|g\| = 1 \Rightarrow \|f + g\| < 2.$$

Silne unormowanie przestrzeni to własność zależna od normy – ta sama przestrzeń liniowa z jedną normą może być przestrzenią silnie unormowaną, a z inną już nie. Geometrycznie, silne unormowanie przestrzeni oznacza, że sfery nie zawierają odcinka. Okazuje się, że własność ta ma istotne znaczenie w przypadku rozwiązywania problemów aproksymacyjnych.

Zachodzi następujące twierdzenie.

**Twierdzenie 7.2.** *Każdy element przestrzeni silnie unormowanej $X$ posiada co najwyżej jeden element optymalny względem dowolnej podprzestrzeni $V$.*

**Dowód**: Dla dowodu nie wprost przypuśćmy, że $h_1^* \neq h_2^*$ to elementy optymalne dla elementu $f$ w dowolnej ustalonej podprzestrzeni $V \subset X$. Wówczas

$$\varepsilon_V(f) = \|f - h_1^*\| = \|f - h_2^*\| > 0;$$

dla $\varepsilon_V(f) = 0$ mielibyśmy $h_1^* = h_2^*$. Niech teraz

$$f_1 = \frac{f - h_1^*}{\varepsilon_V(f)} \quad \text{oraz} \quad f_2 = \frac{f - h_2^*}{\varepsilon_V(f)}.$$

Jest jasne, że $f_1 \neq f_2$ oraz $\|f_1\| = \|f_2\| = 1$. Stąd, na podstawie założonego silnego unormowania przestrzeni $X$, wynika nierówność

$$\left\| \frac{f - h_1^*}{\varepsilon_V(f)} + \frac{f - h_2^*}{\varepsilon_V(f)} \right\| = \|f_1 + f_2\| < 2,$$

którą równoważnie możemy zapisać jako

$$\left\| f - \frac{h_1^* + h_2^*}{2} \right\| < \varepsilon_V(f).$$

Ponieważ $(h_1^* + h_2^*)/2 \in V$, zatem otrzymana zależność przeczy optymalności elementów $h_1^*$ i $h_2^*$. ■

# Rozdział 8

# Aproksymacja średniokwadratowa

## 8.1. Sformułowanie problemu, istnienie i jedyność rozwiązania

### Sformułowanie problemu

Niech $(X, \langle \cdot, \cdot \rangle)$ będzie rzeczywistą przestrzenią unitarną, niech $V \subset X$ będzie niepustą podprzestrzenią liniową oraz niech $f \in X$ będzie dowolnym ustalonym elementem. Problem aproksymacji średniokwadratowej polega na znalezieniu takiego elementu $h^* \in V$, który spełnia warunek

$$\forall h \in V : \quad \|f - h^*\| \leqslant \|f - h\|$$

z normą $\|\cdot\|$ zdefiniowaną przez iloczyn skalarny:

$$\|f\| = \sqrt{\langle f, f \rangle}.$$

Aproksymacja średniokwadratowa może być rozważana w dowolnej przestrzeni unitarnej, niekoniecznie przestrzeni funkcyjnej.

**Przykład 8.1 (przykłady przestrzeni unitarnych).** *Poniżej podajemy przykłady przestrzeni liniowych $X$, ich podprzestrzeni liniowych $V$ oraz określonych w nich iloczynów skalarnych:*

**(a)** $X = \mathbb{R}^n$, $V = V_A = \{x \in \mathbb{R}^n : Ax = 0\}$, *gdzie* $A \in \mathbb{R}^{k \times n}$ *oraz*

$$\langle x, y \rangle = x^T y = \sum_{i=1}^{n} x_i y_i;$$

**(b)** $X = \mathbb{R}^{n \times n}$, $V = \left\{A \in \mathbb{R}^{n \times n} : A = A^T\right\}$ *oraz*

$$\langle A, B \rangle = \operatorname{tr}\left(A^T B\right);$$

**(c)** $X = \mathcal{L}_2\left([a,b], \rho\right) = \left\{f : [a,b] \to \mathbb{R} : \int_a^b f^2(x)\,\rho(x)\,dx < +\infty\right\}$[1], $V = \pi_n$ *oraz*

$$\langle f, g \rangle = \int_a^b f(x)\,g(x)\,\rho(x)\,dx.$$

[1] Funkcja $\rho : (a,b) \to (0, +\infty)$ to tzw. *funkcja wagowa*; spełnia ona następujący warunek:

$$\int_a^b \rho(x)\,dx < +\infty.$$

**Istnienie rozwiązania**

Istnienie elementu optymalnego w sensie aproksymacji średniokwadratowej wynika z twierdzenia 7.1 i jest zagwarantowane założeniem skończonego wymiaru przestrzeni $V$.

**Jedyność rozwiązania**

W przypadku aproksymacji średniokwadratowej element optymalny zawsze jest wyznaczony jednoznacznie. Jest to prosta konsekwencja silnego unormowania każdej rzeczywistej przestrzeni unitarnej.

**Twierdzenie 8.1.** *Przestrzeń euklidesowa (tj. przestrzeń unitarna nad $\mathbb{R}$) jest silnie unormowana.*

**Dowód**: Niech $f, g \in X$, $f \neq g$, $\|f\| = \|g\| = 1$. Wówczas

$$\begin{aligned} 0 < \|f-g\|^2 &= \langle f-g, f-g\rangle = \langle f,f\rangle - 2\langle f,g\rangle + \langle g,g\rangle = \\ &= \|f\|^2 - 2\langle f,g\rangle + \|g\|^2 = 2 - 2\langle f,g\rangle \end{aligned}$$

skąd wynika, że

$$\langle f,g\rangle < 1.$$

Stąd

$$\|f+g\|^2 = \|f\|^2 + 2\langle f,g\rangle + \|g\|^2 = 2 + 2\langle f,g\rangle < 4$$

co oznacza, że $\|f+g\| < 2$. To kończy dowód. ∎

Z twierdzenia 8.1 wynika następujący ważny wniosek.

**Wniosek 8.2.** *Jeżeli $\dim V < \infty$, to każdy element $f \in X$ posiada dokładnie jeden element optymalny w $V$ w sensie aproksymacji średniokwadratowej.*

**Dowód**: Teza jest konsekwencją twierdzeń 8.1 oraz 7.2. ∎

### 8.1.1. Element optymalny jako rzut ortogonalny na podprzestrzeń

W tej części podamy efektywny sposób wyznaczania elementu optymalnego. Zachodzi następujące twierdzenie.

**Twierdzenie 8.3 (optymalność rzutu ortogonalnego).** *Niech $X$ będzie przestrzenią euklidesową oraz niech $V \subset X$ będzie dowolną podprzestrzenią liniową. Element $h^* \in V$ jest elementem optymalnym dla elementu $f \in X$ w $V$ wtedy i tylko wtedy, gdy $f - h^* \perp V$, tj.*

$$\forall h \in V: \quad \langle f-h^*, h\rangle = 0. \tag{8.1}$$

**Dowód**: Przypuśćmy, że $h^* \in V$ to element optymalny dla $f$ w $V$. Wówczas dla dowolnych $h \in V$ oraz $\lambda \in \mathbb{R}$ mamy

$$0 \leqslant \|f - h^* + \lambda h\|^2 - \|f-h^*\|^2 = \lambda^2\|h\|^2 + 2\lambda\langle f-h^*, h\rangle. \tag{8.2}$$

Zauważmy, że występująca z prawej strony nierówności (8.2) funkcja zmiennej $\lambda$ jest, dla ustalonego $h \neq 0$, funkcją kwadratową. Funkcja ta jest stale nieujemna, a to jest możliwe jedynie w przypadku, gdy $\langle f-h^*, h\rangle = 0$.

Przypuśćmy teraz, że $f - h^* \perp V$. Wówczas, na podstawie abstrakcyjnej wersji twierdzenia Pitagorasa[2] otrzymujemy dla dowolnego $h \in V$:

$$\|f-h\|^2 = \|f-h^* + \underbrace{(h^*-h)}_{\in V}\|^2 = \|f-h^*\|^2 + \|h^*-h\|^2 \geqslant \|f-h^*\|^2.$$

[2] W dowolnej przestrzeni euklidesowej zachodzi:

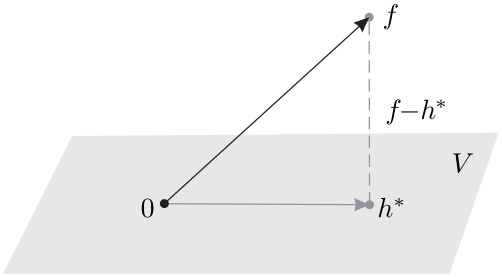
$$f \perp g \Leftrightarrow \|f+g\|^2 = \|f\|^2 + \|g\|^2.$$

To oznacza, że $h^*$ jest elementem optymalnym dla $f$ w $V$. ∎

Element $h^* \in V$ spełniający warunek

$$f - h^* \perp V$$

nazywamy *rzutem ortogonalnym elementu $f$ na $V$* (zob. rysunek 8.1). Z twierdzenia 8.3 wynika, że elementem optymalnym dla elementu $f$ w przestrzeni $V$ jest rzut ortogonalny $f$ na podprzestrzeń $V$.




Rysunek 8.1. Interpretacja geometryczna rzutu ortogonalnego

### 8.1.2. Wyznaczanie elementu optymalnego – układ równań normalnych

Przypuśćmy, że $\dim V = n$ oraz niech wektory $e_1, \ldots, e_n$ będą bazą przestrzeni $V$. Oznacza to, że istnieją współczynniki $\alpha_1, \ldots, \alpha_n \in \mathbb{R}$ dla których element optymalny dla $f$ w $V$ to

$$h^* = \alpha_1 e_1 + \ldots + \alpha_n e_n.$$

Warunek (8.1) równoważnie możemy zapisać jako

$$f - h^* \perp e_i, \quad \text{dla } i = 1, \ldots, n.$$

Stąd, dla $i = 1, \ldots, n$

$$0 = \langle f - h^*, e_i \rangle = \langle f, e_i \rangle - \sum_{k=1}^{n} \alpha_k \langle e_k, e_i \rangle .$$

Otrzymujemy w ten sposób do rozwiązania tzw. *układ równań normalnych*

$$\sum_{k=1}^{n} \alpha_k \langle e_k, e_i \rangle = \langle f, e_i \rangle , \quad i = 1, \ldots, n. \tag{8.3}$$

Jest to układ równań liniowych o macierzy $G \in \mathbb{R}^{n \times n}$ postaci

$$G = \begin{pmatrix} \langle e_1, e_1 \rangle & \langle e_2, e_1 \rangle & \cdots & \langle e_n, e_1 \rangle \\ \vdots & \vdots & \ldots & \vdots \\ \langle e_1, e_n \rangle & \langle e_2, e_n \rangle & \cdots & \langle e_n, e_n \rangle \end{pmatrix}.$$

Macierz $G$ nazywamy *macierz Grama*.

**Stwierdzenie 8.4 (własności macierzy Grama).**

**W1.** *Macierz $G$ jest dodatnio półokreślona, tj.*

$$\forall x \in \mathbb{R}^n : \quad x^T G x \geqslant 0.$$

**W2.** *Macierz $G$ jest nieosobliwa wtedy i tylko wtedy, gdy wektory $e_1, \ldots, e_n$ są liniowo niezależne.*

**Dowód**: Dla $x = (x_1, \ldots, x_n)^T$ mamy

$$x^T G x = \sum_{i,j} \langle e_i, e_j \rangle x_i x_j = \left\langle \sum_i x_i e_i, \sum_j x_j e_j \right\rangle \geqslant 0. \tag{8.4}$$

Macierz $G$ jest zatem dodatnio półokreślona. To dowodzi własności W1. Dodatkowo, wynika stąd, że $x^T G x = 0$ wtedy i tylko wtedy, gdy

$$0 = \sum_{i=1}^{n} x_i e_i. \tag{8.5}$$

W przypadku, gdy wektory $e_1, \ldots, e_n$ są liniowo niezależne z warunku (8.5) wynika, że $x = 0$. Macierz $G$ jest zatem dodatnio określona, więc i nieosobliwa.

Przypuśćmy teraz, że wektory $e_1, \ldots, e_n$ są liniowo zależne. To oznacza, że istnieje wektor $x = (x_1, \ldots, x_n)^T \neq 0$ dla którego

$$\sum_{i=1}^{n} x_i e_i = 0.$$

Z (8.4) otrzymujemy wówczas $x^T G x = 0$. Macierz $G$ jest więc macierzą osobliwą (jest macierzą dodatnio półokreśloną, która nie jest macierzą dodatnio określoną – musi więc posiadać zerową wartość własną). To kończy dowód. ∎

Z własności W2 macierzy Grama wynika, że układ równań normalnych posiada dokładnie jedno rozwiązanie. Niestety, w przypadku ogólnym może ono być trudne lub wręcz niemożliwe do wyznaczenia numerycznie, ze względu na duży wskaźnik uwarunkowania zadania.

**Przykład 8.2 (przykład Hilberta).** *Niech $X = \mathcal{L}_2([0,1])$, niech $V = \pi_n$. W przestrzeni $V$ wybieramy bazę naturalną $e_0, \ldots, e_n$,*

$$e_i(x) = x^i, \text{ dla } i = 0, \ldots, n.$$

*Wówczas*

$$\langle e_i, e_j \rangle = \int_0^1 x^{i+j} dx = \frac{1}{i+j+1},$$

*a macierz $G$ ma postać*

$$G = \begin{pmatrix} 1 & 1/2 & 1/3 & \cdots & 1/(n+1) \\ 1/2 & 1/3 & \ddots & & 1/(n+2) \\ 1/3 & \ddots & \ddots & \ddots & \vdots \\ \vdots & & \ddots & \ddots & \vdots \\ 1/(n+1) & 1/(n+2) & \cdots & \cdots & 1/(2n+1) \end{pmatrix}.$$

*Można wykazać, że*

$$\operatorname{cond}(G) = \|G\|_2 \left\|G^{-1}\right\|_2 \approx 10^n,$$

*co oznacza olbrzymie trudności podczas numerycznego rozwiązywania układu równań normalnych powstałego w rozważanym przypadku.*

Zauważmy, że postać macierz $G$ zależy od wyboru bazy przestrzeni $V$. W przypadku bazy ortogonalnej, macierz $G$ jest diagonalna a rozwiązanie układu równań normalnych (8.3) ma wówczas postać

$$\alpha_i = \frac{\langle f, e_i \rangle}{\langle e_i, e_i \rangle}, \quad \text{dla } i = 1, \ldots, n.$$

Wynika stąd następujący wniosek.

**Wniosek 8.5 (postać elementu optymalnego).** *Jeżeli wektory $e_1, \ldots, e_n$ są bazą ortogonalną przestrzeni $V$, to element optymalny dla elementu $f$ w podprzestrzeni $V$ ma postać*

$$h^* = \frac{\langle f, e_1\rangle}{\langle e_1, e_1\rangle} e_1 + \ldots + \frac{\langle f, e_n\rangle}{\langle e_n, e_n\rangle} e_n.$$

## 8.2. Wielomiany ortogonalne

W przypadku aproksymacji funkcji szczególnie istotną rolę odgrywają wielomiany, w całej ich klasie najważniejsze dla nas będą wielomiany ortogonalne. Znajdą one zastosowanie zarówno w algorytmach aproksymacyjnych jak również podczas numerycznego wyznaczania całek oznaczonych.

**Definicja 8.1 (ciąg wielomianów ortogonalnych).** *Ciąg $\{P_n\}_{n\in\mathbb{N}_0}$ nazywamy ciągiem wielomianów ortogonalnych w przestrzeni $\mathcal{L}_2([a,b],\rho)$, jeżeli*

- *$P_n$ jest wielomianem stopnia $n$;*
- *dla $m \neq n$ zachodzi*

$$\int_a^b P_n(x) P_m(x) \rho(x)\, dx = 0,$$

*tzn. funkcje $P_0, P_1, \ldots$ tworzą układ ortogonalny.*

### 8.2.1. Przykłady wielomianów ortogonalnych

**Wielomiany Czebyszewa pierwszego rodzaju**

Tworzą układ ortogonalny w przestrzeni $\mathcal{L}_2([-1,1],\rho)$ z funkcją wagową $\rho(x) = 1/\sqrt{1-x^2}$; postać jawna to:

$$T_n(x) = \cos(n \arccos x), \quad \text{dla } x \in [-1,1],\ n \in \mathbb{N}_0.$$

**Wielomiany Czebyszewa drugiego rodzaju**

Tworzą układ ortogonalny w przestrzeni $\mathcal{L}_2([-1,1],\rho)$ z funkcją wagową $\rho(x) = \sqrt{1-x^2}$; postać jawna to:

$$U_n(x) = \frac{\sin((n+1)\arccos x)}{\sqrt{1-x^2}}, \quad \text{dla } x \in (-1,1),\ n \in \mathbb{N}_0.$$

**Wielomiany Legendre'a**

Tworzą układ ortogonalny w przestrzeni $\mathcal{L}_2([-1,1])$; postać jawna to:

$$L_n(x) = \frac{1}{2^n n!} \frac{d^n}{dx^n}\left(x^2-1\right)^n, \quad \text{dla } x \in (-1,1),\ n \in \mathbb{N}_0.$$

**Wielomiany Hermite'a**

Tworzą układ ortogonalny w przestrzeni $\mathcal{L}_2(\mathbb{R},\rho)$ z funkcją wagową $\rho(x) = e^{-x^2}$; postać jawna to:

$$H_n(x) = (-1)^n e^{x^2} \frac{d^n}{dx^n} e^{-x^2}, \quad \text{dla } x \in \mathbb{R},\ n \in \mathbb{N}_0.$$

### 8.2.2. Własności wielomianów ortogonalnych

Podane poniżej ogólne własności wielomianów ortogonalnych będziemy wielokrotnie wykorzystywać w przyszłości.

**Stwierdzenie 8.6.** *Niech $\{P_n\}_{n\in\mathbb{N}_0}$ będzie ciągiem wielomianów ortogonalnych w przestrzeni $\mathcal{L}_2([a,b],\rho)$ wyposażonej w iloczyn skalarny*

$$\langle f, g\rangle = \int_a^b f(x)\, g(x)\, \rho(x)\, dx.$$

**W1.** *Dla każdego $n \in \mathbb{N}_0$ wielomiany $P_0, \ldots, P_n$ stanowią bazę przestrzeni $\pi_n$.*

**W2.** $\forall j \in \mathbb{N}\ \forall f \in \pi_{j-1} : \langle f, P_j \rangle = 0$.
**W3.** *Istnieją ciągi liczbowe* $\{\alpha_k\}, \{\beta_k\}, \{\gamma_k\}$ *dla których*

$$P_{k+2}(x) = (\alpha_k x + \beta_k) P_{k+1}(x) + \gamma_k P_k(x), \quad \text{dla } k \in \mathbb{N}_0. \tag{8.6}$$

**W4.** *W każdej przestrzeni ciąg wielomianów ortogonalnych jest wyznaczony jednoznacznie z dokładnością do niezerowych mnożników.*
**W5.** *Wszystkie zera wielomianu* $P_k$ *(dla* $k \in \mathbb{N}$*) są rzeczywiste, pojedyncze i leżą w przedziale otwartym* $(a, b)$.

Zależność (8.6) to tzw. *reguła trójczłonowa*.
**Dowód**: Uzasadnienia własności W1 oraz W2 pozostawiam jako proste ćwiczenie, dowód własności W3 pomijam (zainteresowanych odsyłam do monografii J. i M. Jankowscy, *Przegląd metod i algorytmów numerycznych. Część 1.*, WNT, Warszawa 1981, str. 92-94).

W celu uzasadnienia własności W4 przypuśćmy, że $\{P_n\}$ oraz $\{Q_n\}$ to ciągi wielomianów ortogonalnych w tej samej przestrzeni. Na podstawie własności W1, wielomian $P_k$ jest kombinacją liniową wielomianów $Q_0, \ldots, Q_k$, tj.

$$P_k = \sum_{i=0}^{k} \alpha_i Q_i. \tag{8.7}$$

Z własności W2, dla $j < k$ mamy

$$0 = \langle P_k, Q_j \rangle = \left\langle \sum_{i=0}^{k} \alpha_i Q_i, Q_j \right\rangle = \sum_{i=0}^{k} \alpha_i \langle Q_i, Q_j \rangle = \alpha_j \langle Q_j, Q_j \rangle .$$

To oznacza, że $\alpha_j = 0$ dla $j < k$. Z (8.7) wynika więc, że

$$P_k = \alpha_k Q_k.$$

Aby uzasadnić własność W5 przypuśćmy, że wielomian $P_k$ zmienia znak dokładnie $j$ razy w przedziale $(a, b)$; punkty zmiany znaku (będące oczywiście pierwiastkami wielomianu $P_k$) oznaczmy jako $x_1, \ldots, x_j$. Chcemy wykazać, że $j = k$. Rozważmy wielomian

$$Q_j(x) = \begin{cases} 1 & \text{dla } j = 0 \\ \prod_{m=1}^{j} (x - x_m) & \text{dla } j \geqslant 1 \end{cases}.$$

Oczywiście $\deg Q_j \leqslant k$. Ponadto

$$\langle P_k, Q_j \rangle = \int_a^b P_k(x) Q_j(x) \rho(x)\, dx \neq 0,$$

gdyż funkcja podcałkowa nie zmienia znaku w przedziale $(a, b)$. To, na podstawie własności W2, oznacza, że $j = k$. ∎

### 8.2.3. Wielomiany Czebyszewa pierwszego rodzaju

Wielomianom Czebyszewa pierwszego rodzaju, ze względu na ich liczne (i ważne) zastosowania, przyjrzymy się nieco dokładniej.

#### Postać jawna

Dla $x \in \mathbb{R}$ oraz $n \in \mathbb{N}_0$

$$T_n(x) = \begin{cases} (-1)^n \cosh(n \operatorname{arcosh} x) & \text{dla } x < -1 \\ \cos(n \operatorname{arc\,cos} x) & \text{dla } x \in [-1, 1] \\ \cosh(n \operatorname{arcosh} x) & \text{dla } x > 1 \end{cases}.$$

**Reguła trójczłonowa**

Zachodzi

$$T_0(x) = 1, \quad T_1(x) = x,$$
$$T_k(x) = 2xT_{k-1}(x) - T_{k-2}(x), \quad \text{dla } k \geqslant 2.$$

**Współczynnik wiodący**

Wiemy, że $\deg T_n = n$, dla $n \in \mathbb{N}_0$. Łatwo wykazać indukcyjnie (wykorzystując regułę trójczłonową), że dla $n \geqslant 1$ współczynnikiem wiodącym wielomianu $T_n$ jest $a_n = 2^{n-1}$, tj.

$$T_n(x) = 2^{n-1}x^n + a_{n-1}x^{n-1} + \ldots + a_0. \tag{8.8}$$

**Parzystość, nieparzystość**

Łatwo wykazać indukcyjnie (ponownie wykorzystując regułę trójczłonową) następującą własność wielomianów Czebyszewa:

$$T_n(-x) = (-1)^n T_n(x), \qquad \text{dla } n \in \mathbb{N}_0.$$

Wynika z niej, że dla $n$ parzystych wielomian $T_n$ jest funkcją parzystą, dla $n$ nieparzystych jest funcją nieparzystą.

**Ortogonalność**

Jak już wiemy, wielomiany Czebyszewa pierwszego rodzaju $\{T_n\}$ tworzą układ ortogonalny w przestrzeni $\mathcal{L}_2([-1,1],\rho)$ z funkcją wagową $\rho(x) = 1/\sqrt{1-x^2}$. Można łatwo wykazać, że

$$\langle T_k, T_m \rangle = \int_{-1}^{1} T_k(x)\, T_m(x)\, \frac{dx}{\sqrt{1-x^2}} = \begin{cases} 0 & \text{dla } k \neq m \\ \pi & \text{dla } k = m = 0 \\ \pi/2 & \text{dla } k = m \geqslant 1 \end{cases}. \tag{8.9}$$

**Miejsca zerowe**

Wiemy, że wielomian $T_n$ posiada dokładnie $n$ różnych, rzeczywistych pierwiastków leżących w przedziale $(-1,1)$. Łatwo wykazać, że pierwiastki te to

$$x_k = \cos\frac{1+2k}{2n}\pi, \quad \text{dla } k = 0, \ldots, n-1;$$

zauważmy, że $x_0 > x_1 > \ldots > x_{n-1}$.

**Ekstrema**

Z postaci jawnej wielomianów Czebyszewa wynika, że $\|T_n\|_{[-1,1]} = 1$. Łatwo wykazać, że wielomian $T_n$ posiada dokładnie $n+1$ ekstremów lokalnych w punktach

$$\widetilde{x_k} = \cos\frac{k\pi}{n}, \quad \text{dla } k = 0, \ldots, n$$

oraz $1 = \widetilde{x_0} > \widetilde{x_1} > \ldots > \widetilde{x_n} = -1$. Dodatkowo

$$T_n(\widetilde{x_k}) = \cos(n \arccos \widetilde{x_k}) = \cos(k\pi) = (-1)^k.$$

**Własność optymalności w normie supremum**

Niech $\pi_n^*$ oznacza rodzinę unormowanych wielomianów stopnia co najwyżej $n$, tj.

$$\pi_n^* = \left\{x \to x^n + a_{n-1}x^{n-1} + \ldots + a_0 : a_{n-1}, \ldots, a_0 \in \mathbb{R}\right\}.$$

Z (8.8) wynika, że $T_n^* = \frac{1}{2^{n-1}} T_n \in \pi_n^*$. Zachodzi następujące twierdzenie.

**Twierdzenie 8.7.** *Niech $w \in \pi_n^*$ będzie dowolnym wielomianem. Wówczas*

$$\|w\|_{[-1,1]} \geqslant \|T_n^*\|_{[-1,1]} = \frac{1}{2^{n-1}}. \tag{8.10}$$

*Dodatkowo, dla każdego wielomianu $w \in \pi_n^*$, $w \neq T_n^*$ nierówność* (8.10) *jest silna.*

**Dowód**: Dla dowodu nie wprost przypuśćmy, że istnieje wielomian $w \in \pi_n^*$, $w \neq T_n^*$, dla którego

$$\|w\|_{[-1,1]} \leqslant \frac{1}{2^{n-1}}. \tag{8.11}$$

Niech $\widetilde{x_0}, \ldots, \widetilde{x_n}$ będą punktami, w których wielomian $T_n$ osiąga swoje ekstrema lokalne; wiemy, że $T_n(\widetilde{x_k}) = (-1)^k$. Rozważmy wielomian $q = w - T_n^* \in \pi_{n-1}$. W punktach $\widetilde{x_0}, \ldots, \widetilde{x_n}$ mamy

$$\begin{aligned} q(\widetilde{x_0}) &= w(\widetilde{x_0}) - T_n^*(\widetilde{x_0}) = w(\widetilde{x_0}) - \frac{1}{2^{n-1}} \leqslant 0 \\ q(\widetilde{x_1}) &= w(\widetilde{x_1}) - T_n^*(\widetilde{x_1}) = w(\widetilde{x_1}) + \frac{1}{2^{n-1}} \geqslant 0 \\ q(\widetilde{x_2}) &= w(\widetilde{x_2}) - T_n^*(\widetilde{x_2}) = w(\widetilde{x_2}) - \frac{1}{2^{n-1}} \leqslant 0 \\ &\vdots \end{aligned} \tag{8.12}$$

Ponieważ stopień wielomianu $q$ nie przekracza $n-1$, zatem, zgodnie z uwagą 3.2 ze str. 24,

$$q[\widetilde{x_0}, \ldots, \widetilde{x_n}] = 0.$$

Wykorzystując definicję różnicy dzielonej możemy więc zapisać

$$\begin{aligned} 0 = q[\widetilde{x_0}, \ldots, \widetilde{x_n}] = &\frac{q(\widetilde{x_0})}{\prod_{j=1}^{n}(\widetilde{x_0} - \widetilde{x_j})} + \frac{q(\widetilde{x_1})}{(\widetilde{x_1} - \widetilde{x_0})\prod_{j=2}^{n}(\widetilde{x_1} - \widetilde{x_j})} + \\ &+ \frac{q(\widetilde{x_2})}{(\widetilde{x_2} - \widetilde{x_0})(\widetilde{x_2} - \widetilde{x_1})\prod_{j=3}^{n}(\widetilde{x_2} - \widetilde{x_j})} + \ldots + \frac{q(\widetilde{x_n})}{\prod_{j=0}^{n-1}(\widetilde{x_n} - \widetilde{x_j})} \end{aligned}$$

Ponieważ $\widetilde{x_0} > \ldots > \widetilde{x_n}$ oraz uwzględniając nierówności (8.12) wnioskujemy, że różnica dzielona $q[\widetilde{x_0}, \ldots, \widetilde{x_n}] = 0$ jest sumą $n+1$ liczb niedodatnich – każdy jej składnik musi więc mieć wartość zero, tj.

$$q(\widetilde{x_0}) = q(\widetilde{x_1}) = \ldots = q(\widetilde{x_n}) = 0.$$

To oznacza, że $q \equiv 0$. Sprzeczność z wyborem wielomianu $w \neq T_n^*$ kończy dowód. ∎

**Uwaga 8.1.** *Przypomnijmy, że błąd interpolacji wielomianowej w przypadku interpolowania funkcji $f \in \mathcal{C}^{n+1}$ w węzłach $x_0, \ldots, x_n$ wielomianem $w \in \pi_n$ wyraża się wzorem*

$$f(x) - w(x) = \frac{f^{(n+1)}(\xi)}{(n+1)!} \prod_{k=0}^{n}(x - x_k),$$

*gdzie $\xi \in \operatorname{conv}(x, x_0, \ldots, x_n)$. Ponieważ*

$$\begin{aligned} \max_{x\in[a,b]} |\prod_{k=0}^{n}(x - x_k)| &= \max_{x\in[-1,1]} |\prod_{k=0}^{n}\left(\frac{1}{2}(b-a)x + \frac{1}{2}(a+b) - x_k\right)| = \\ &= \left(\frac{b-a}{2}\right)^{n+1} \max_{x\in[-1,1]} |\prod_{k=0}^{n}\left(x + \frac{a+b}{b-a} - \frac{2}{b-a}x_k\right)|, \end{aligned}$$

*zatem z udowodnionego przed chwilą twierdzenia wynika, że oszacowanie w normie supremum błędu interpolacji będzie najmniejsze dla węzłów $x_0, \ldots, x_n$ postaci*

$$x_k = \frac{b-a}{2}t_k + \frac{b+a}{2}, \quad \text{dla } k = 0, \ldots, n$$

*gdzie $t_0, \ldots, t_n$ to pierwiastki wielomianu $T_{n+1}$. Otrzymujemy wówczas oszacowanie*

$$\|f - w\|_{[a,b]} \leqslant \frac{\left\|f^{(n+1)}\right\|_{[a,b]}}{(n+1)!}\frac{(b-a)^{n+1}}{2^{2n+1}}.$$

**Przykład 8.3.** *Niech* $X = \mathcal{L}_2([-1,1],\rho)$*, gdzie* $\rho(x) = 1/\sqrt{1-x^2}$*. Poszukamy elementu optymalnego dla funkcji* $f(x) = x^{n+1}$ *w przestrzeni* $V = \pi_n$*.*

*Wiemy, że w rozważanym przypadku element optymalny* $h^*$ *jest wyznaczony jednoznacznie i musi spełniać warunek*

$$f - h^* \perp \pi_n.$$

*Jest jasne, że* $f - h^* \in \pi_{n+1}$*, a wielomianem stopnia* $n+1$ *prostopadłym do wszystkich wielomianów z* $\pi_n$ *jest wielomian* $\alpha T_{n+1}$ *dla dowolnej wartości* $\alpha \neq 0$*. Stąd*

$$x^{n+1} - h^*(x) = \alpha T_{n+1}(x).$$

*Powyższa równość może być spełniona jedynie dla* $\alpha = 1/2^n$*, czyli*

$$h^*(x) = x^{n+1} - \frac{1}{2^n} T_{n+1}(x), \tag{8.13}$$

*błąd aproksymacji to*

$$\varepsilon_V(f) = \|f - h^*\|_{\mathcal{L}_2} = \frac{1}{2^n}\|T_{n+1}\|_{\mathcal{L}_2} = \frac{1}{2^n}\sqrt{\langle T_{n+1}, T_{n+1}\rangle} = \frac{1}{2^n}\sqrt{\frac{\pi}{2}}.$$

*Zauważmy, że z jednej z wymienionych powyżej własności wielomianów Czebyszewa (z której?) wynika, że wyznaczony powyżej element* (8.13) *jest wielomianem stopnia co najwyżej* $n-1$*. Poszukując elementu optymalnego dla funkcji* $f(x) = x^{n+1}$ *w przestrzeni* $V = \pi_{n-1}$ *rozwiązaniem będzie więc ten sam element.*

## Rozdział 9

# Aproksymacja jednostajna

### 9.1. Sformułowanie problemu, istnienie i jedyność rozwiązania

**Sformułowanie problemu**

Aproksymacja jednostajna to aproksymacja w sensie normy supremum w przestrzeni rzeczywistych funkcji ciągłych $\mathcal{C}_{[a,b]}$. Polega ona na znalezieniu, dla ustalonej funkcji ciągłej $f$ oraz podprzestrzeni $V \subset \mathcal{C}_{[a,b]}$, takiego elementu $h^* \in V$, który spełnia warunek

$$\forall h \in V: \quad \max_{x\in[a,b]} |f(x) - h^*(x)| \leqslant \max_{x\in[a,b]} |f(x) - h(x)|.$$

**Istnienie rozwiązania**

Istnienie elementu optymalnego w sensie aproksymacji jednostajnej wynika z twierdzenia 7.1 i jest zagwarantowane założeniem skończonego wymiaru przestrzeni $V$.

**Jedyność rozwiązania**

W przypadku aproksymacji jednostajnej element optymalny nie musi być wyznaczony jednoznacznie (zob. przykład 7.2). Z przykładu 7.2 wynika również, że norma supremum nie pochodzi od żadnego iloczynu skalarnego, tzn. nie istnieje iloczyn skalarny $\langle\cdot,\cdot\rangle$ dla którego

$$\max_{x\in[a,b]} |f(x)| = \sqrt{\langle f, f\rangle},$$

dla wszystkich $f \in \mathcal{C}_{[a,b]}$.

#### 9.1.1. Warunek Haara

Warunkiem gwarantującym jedyność elementu optymalnego jest warunek Haara.

**Definicja 9.1 (warunek Haara).** *Podprzestrzeń $n$-wymiarowa $V \subset \mathcal{C}_{[a,b]}$ spełnia warunek Haara, jeżeli dowolna funkcja $f \in V$, $f \not\equiv 0$ zeruje się w co najwyżej $n-1$ punktach, tj.*

$$\#\{x \in [a,b] : f(x) = 0\} \leqslant n-1.$$

**Przykład 9.1.** *Podprzestrzeń $\pi_n$ spełnia warunek Haara:* $\dim \pi_n = n+1$ *oraz każdy wielomian stopnia $n > 0$ ma co najwyżej $n$ pierwiastków.*

*Podprzestrzeń $V = \operatorname{span}\{x, x^2, \ldots, x^n\}$ przestrzeni $\mathcal{C}_{[0,1]}$ jest przykładem podprzestrzeni, która nie spełnia warunku Haara:* $\dim V = n$ *oraz wielomian*

$$f(x) = x(x-1)(x-1/2)\ldots(x-1/(n-1))$$

*ma $n$ pierwiastków w przedziale* $[0,1]$.

W przykładzie 7.2 rozważaliśmy podprzestrzeń $V = \operatorname{span}\{x\}$, która nie spełnia warunku Haara. Element optymalny dla funkcji $f(x) = \cos x$ nie był wówczas wyznaczony jednoznacznie. Okazuje się, że to nie przypadek.

Zachodzi następujące twierdzenie.

**Twierdzenie 9.1.** *Niech $V \subset \mathcal{C}_{[a,b]}$ będzie podprzestrzenią skończenie wymiarową. Następujące warunki są równoważne:*

- *dla każdej funkcji $f \in \mathcal{C}_{[a,b]}$ istnieje dokładnie jeden element optymalny w $V$;*
- *podprzestrzeń $V$ spełnia warunek Haara.*

**Dowód**: Dowód pomijamy. ∎

### 9.1.2. Twierdzenie Czebyszewa o alternansie

Twierdzenie, które teraz podamy stanowi kluczowy krok konstrukcji elementu optymalnego w sensie aproksymacji jednostajnej.

**Twierdzenie 9.2 (Czebyszewa o alternansie).** *Niech $V \subset \mathcal{C}_{[a,b]}$ będzie podprzestrzenią $n$-wymiarową spełniającą warunek Haara oraz niech $f \in X$. Wówczas*

**1.** *istnieje dokładnie jeden element optymalny dla $f$ w $V$.*
**2.** *element $h^* \in V$ jest elementem optymalnym dla $f$ wtedy i tylko wtedy, gdy istnieje $n+1$ punktów (tzw. alternans)*

$$a \leqslant x_0 < \ldots < x_n \leqslant b$$

*dla których*
**a)** $\|f - h^*\|_{[a,b]} = |f(x_k) - h^*(x_k)|$, *dla* $k = 0, \ldots, n$;
**b)** $f(x_{k+1}) - h^*(x_{k+1}) = -(f(x_k) - h^*(x_k))$, *dla* $k = 0, \ldots, n-1$.

**Dowód**: Pierwsza część tezy wynika z twierdzenia 9.1. W przypadku drugiej części, wykażemy jedynie implikację $\Leftarrow$.

Dla dowodu nie wprost przypuśćmy, że element $h^* \in V$ spełniający warunki a)–b) nie jest elementem optymalnym dla $f$. Istnieje zatem element $h \in V$ dla którego

$$\|f - h\|_{[a,b]} < \|f - h^*\|_{[a,b]}. \tag{9.1}$$

W punktach alternansu mamy

$$|f(x_k) - h(x_k)| \leqslant \|f - h\|_{[a,b]} < \|f - h^*\|_{[a,b]} = |f(x_k) - h^*(x_k)|. \tag{9.2}$$

Niech $\varphi = h - h^*$. Ponieważ

$$\varphi(x_k) = h(x_k) - h^*(x_k) = (f(x_k) - h^*(x_k)) - (f(x_k) - h(x_k))$$

zatem z warunku (9.2) wynika, że znak $\varphi(x_k)$ jest taki sam jak znak $f(x_k) - h^*(x_k)$. Na podstawie punktu b) wnioskujemy, że funkcja ciągła $\varphi = h - h^*$ zmienia znak co najmniej $n$ razy w przedziale $[a, b]$. To na podstawie warunku Haara oznacza, że $\varphi = 0$, czyli $h = h^*$. Sprzeczność ze sposobem wyboru elementu $h$ kończy dowód. ∎

Z twierdzenia Czebyszewa o alternansie wynika, że punktów alternansu należy poszukiwać wśród tych punktów, w których funkcja ciągła $f - h^*$ osiąga swoje wartości ekstremalne. Są to:

- punkty $a, b$;
- punkty, w których zeruje się pochodna funkcji $f - h^*$;
- punkty, w których pochodna funkcji $f - h^*$ nie istnieje.

Numerycznie punkty alternansu oraz element optymalny wyznaczamy przy pomocy iteracyjnego algorytmu Remeza.

Wyznaczanie elementu optymalnego w sensie normy supremum (czyli problem aproksymacji jednostajnej) jest na ogół dużo trudniejsze, niż w przypadku normy zadanej przez iloczyn skalarny (czyli w przypadku aproksymacji średniokwadratowej). W szczególnych przypadkach element optymalny można wyznaczyć dość łatwo.

**Przykład 9.2.** *Niech $X = \mathcal{C}_{[-1,1]}$. Poszukamy elementu optymalnego (w sensie aproksymacji jednostajnej) dla wielomianu $f(x) = x^{n+1}$ w podprzestrzeni $\pi_n$.*

*Musimy zatem wyznaczyć wielomian $h^* \in \pi_n$ minimalizujący wartość*

$$\max_{x\in[-1,1]} |f(x) - h^*(x)|.$$

*Z twierdzenia 8.7 wynika, że wielomian $h^*$ należy wybrać tak, aby*

$$f - h^* = \frac{1}{2^n} T_{n+1}.$$

*Stąd*

$$h^*(x) = x^{n+1} - \frac{1}{2^n} T_{n+1}(x)$$

*a błąd aproksymacji to*

$$\varepsilon_{\pi_n}(f) = \|f - h^*\|_{[-1,1]} = \frac{1}{2^n} \|T_{n+1}\|_{[-1,1]} = \frac{1}{2^n}.$$

## Rozdział 10

# Kwadratury, czyli metody przybliżonego obliczania całek oznaczonych

### 10.1. Motywacja

Dla zadanej funkcji ciągłej $f : [a, b] \to \mathbb{R}$ oraz funkcji wagowej $\rho$ niech

$$I(f) = \int_a^b f(x)\,\rho(x)\,dx.$$

Można podać kilka powodów, które uzasadniają, że obliczanie wartości całki $I(f)$ w oparciu o podstawowy wzór rachunku całkowego

$$I(f) = F(b) - F(a), \tag{10.1}$$

gdzie $F$ to dowolna funkcja pierwotna funkcji podcałkowej, może być ryzykowne lub wręcz niemożliwe:

- istnieją funkcje ciągłe, które nie posiadają funkcji pierwotnej elementarnej, np.

  $$x \to \frac{\sin x}{x}, \qquad x \to e^{-x^2};$$

  dla nich wzór (10.1) jest bezużyteczny;
- dla wielu funkcji posiadających funkcję pierwotną elementarną zastosowanie wzoru (10.1) może prowadzić do dużo gorszych przybliżeń rzeczywistej wartości całki $I(f)$ niż zastosowanie dobrze dobranej kwadratury (przybliżeń, ponieważ wartości $F(a)$, $F(b)$ oraz $F(b) - F(a)$ wyliczane są w arytmetyce fl, więc niedokładnie).

### 10.2. Kwadratury liniowe, rząd kwadratury

Nasze dalsze rozważania ograniczymy do kwadratur liniowych ze względu na funkcję $f$ (ma to sens, ponieważ funkcjonał $I$ jest funkcjonałem liniowym). Będziemy również chcieli, aby kwadratury wykorzystywały jedynie tę informację o funkcji całkowalnej, którą w praktyce dysponujemy; najczęściej są to wartości funkcji (czasami również jej pochodnych) w ustalonych punktach.[1] Niech zatem

$$I(f) \approx S(f) = \sum_{i=0}^{k} \sum_{j=0}^{m_i-1} A_{ij} f^{(j)}(x_i), \tag{10.2}$$

gdzie

[1] W zagadnieniach praktycznych postać analityczna funkcji to luksus, na który rzadko można sobie pozwolić.

- $x_0, \ldots, x_k \in [a, b]$ to węzły,
- $m_0, \ldots, m_k \in \mathbb{N}$ to krotności węzłów,
- $A_{ij}$, dla $i = 0, \ldots, k$; $j = 0, \ldots, m_i - 1$ to współczynniki kwadratury $S$.

Liczbę $R_S(f) = I(f) - S(f)$ nazywamy *błędem kwadratury* $S$.

### 10.2.1. Klasyczne kryterium oceny jakości kwadratury

Miarą jakości kwadratury będzie dla nas jej rząd.

**Definicja 10.1 (rząd kwadratury).** *Liczbę $r = r(S) \in \mathbb{N}$ nazywamy rzędem kwadratury $S$, jeżeli*

**a)** $\forall w \in \pi_{r-1} : S(w) = I(w)$;
**b)** $\exists w \in \pi_r : S(w) \neq I(w)$.

Zanotujmy dwie ważne obserwacje, wynikające z liniowości funkcjonałów $I$ oraz $S$.

**Uwaga 10.1.** *Jeżeli rząd kwadratury $S$ jest równy $r$, to nierówność*

$$S(w) \neq I(w)$$

*zachodzi dla wszystkich wielomianów $w \in \pi_r \backslash \pi_{r-1}$. Oznacza to, że warunek b) definicji 10.1, pomio występującego w nim kwantyfikatora szczegółowego, jest łatwy do zweryfikowania.*

**Uwaga 10.2.** *Warunki definicji 10.1 wystarczy sprawdzać dla dowolnej bazy $\{p_k\}$ przestrzeni wielomianów, której element $p_k$ jest wielomianem stopnia $k$: warunek a) sprawdzamy na wielomianach bazowych coraz wyższych stopni; rząd kwadratury $S$ to najniższy stopień wielomianu, dla którego równość $S(w) = I(w)$ nie zachodzi.*

## 10.3. Kwadratury interpolacyjne

W całej klasie kwadratur liniowych szczególne miejsce zajmują kwadratury interpolacyjne.

**Definicja 10.2 (kwadratura interpolacyjna).** *Kwadraturę $S$ postaci* (10.2) *nazywamy kwadraturą interpolacyjną, jeżeli*

$$S(f) = I(H_n),$$

*gdzie $H_n \in \pi_n$ to wielomian interpolacyjny Hermite'a interpolujący funkcję $f$ w węzłach $x_0, \ldots, x_k$ o krotnościach $m_0, \ldots, m_k$, $m_0 + \ldots + m_k = n + 1$.*

**Przykład 10.1.** *Dla pojedynczych węzłów $x_0, \ldots, x_n$, wykorzystując postać Lagrange'a* (3.2) *wielomianu interpolacyjnego $L_n$, mamy*

$$S(f) = I(L_n) = \int_a^b \sum_{i=0}^{n} f(x_i)\, \varphi_i(x)\, \rho(x)\, dx = \sum_{i=0}^{n} A_i f(x_i),$$

*gdzie*

$$A_i = \int_a^b \varphi_i(x)\, \rho(x)\, dx, \quad dla\ i = 0, \ldots, n.$$

### 10.3.1. Rząd kwadratury interpolacyjnej

Z twierdzenia 4.1 wynika, że dla funkcji $f \in \mathcal{C}^{n+1}_{[a,b]}$ zachodzi

$$R_S(f) = I(f) - S(f) = I(f) - I(H_n) = I(f - H_n) =$$
$$= \int_a^b \frac{f^{(n+1)}(\xi)}{(n+1)!} \prod_{i=0}^{k} (x - x_i)^{m_i} \rho(x) dx.$$

Otrzymujemy stąd następujący ważny wniosek.

**Wniosek 10.1.** *Rząd kwadratury interpolacyjnej opartej na węzłach o łącznej krotności $n+1$ jest nie mniejszy niż $n+1$.*

Okazuje się, że prawdziwe jest również wynikanie w drugą stronę.

**Twierdzenie 10.2 (o rzędzie kwadratury interpolacyjnej).** *Kwadratura $S$ postaci (10.2) oparta na węzłach o łącznej krotności $n+1$ jest interpolacyjna wtedy i tylko wtedy, gdy jej rząd jest nie mniejszy niż $n+1$.*

**Dowód**: Implikacja $\Rightarrow$ wynika z wniosku 10.1. Aby wykazać implikację odwrotną, przypuśćmy, że $H_n \in \pi_n$ jest wielomianem interpolacyjnym Hermite'a interpolującym funkcję $f$ w węzłach $x_0, \ldots, x_k$ o krotnościach $m_0, \ldots, m_k$, tj.

$$f^{(j)}(x_i) = H_n^{(j)}(x_i)$$

dla $i = 0, \ldots, k$ oraz $j = 0, \ldots, m_i - 1$. Stąd

$$S(f) = \sum_{i=0}^{k} \sum_{j=0}^{m_i-1} A_{ij} f^{(j)}(x_i) = \sum_{i=0}^{k} \sum_{j=0}^{m_i-1} A_{ij} H_n^{(j)}(x_i) = S(H_n) = I(H_n),$$

ostatnia równość wynika z założenia $r(S) \geqslant n+1$. To kończy dowód. ∎

Zastanowimy się teraz, jak duży może być rząd kwadratury interpolacyjnej opartej na węzłach o łącznej krotności $n+1$. W tym celu przeanalizujmy następujący przykład.

**Przykład 10.2.** *Przy powyższych oznaczeniach, niech $p_{n+1} \in \pi_{n+1}$ będzie wielomianem postaci*

$$p_{n+1}(x) = \prod_{i=0}^{k} (x - x_i)^{m_i} \tag{10.3}$$

*oraz niech*

$$\tilde{p}(x) = p_{n+1}(x) \cdot \prod_{i=0}^{k} (x - x_i)^{\varepsilon_i},$$

*gdzie, dla $i = 0, \ldots, k$,*

$$\varepsilon_i = \begin{cases} 0 & \text{gdy } m_i \in 2\mathbb{N} \\ 1 & \text{gdy } m_i \notin 2\mathbb{N} \end{cases}.$$

*Wówczas, dla kwadratury $S$ postaci (10.2) mamy*

$$0 = S(\tilde{p}) \neq I(\tilde{p}) > 0.$$

*Zauważmy również, że*

$$\deg \tilde{p} = n + 1 + \sum_{i=0}^{k} \varepsilon_i \leqslant n + k + 2 \leqslant 2n + 2$$

*oraz równość $\deg \tilde{p} = 2n+2$ będzie osiągnięta w przypadku, gdy wszystkie węzły $x_0, \ldots, x_n$ będą jednokrotne.*

Z powyższego przykładu wynika natychmiast następujący wniosek.

**Wniosek 10.3.** *Rząd kwadratury interpolacyjnej opartej na węzłach o łącznej krotności $n+1$ jest nie większy niż $2n+2$.*

### 10.3.2. Kwadratury Gaussa

Udowodnimy teraz twierdzenie mówiące o tym, że istnieje kwadratura o maksymalnym rzędzie.

**Twierdzenie 10.4 (o kwadraturze Gaussa).** *Istnieje dokładnie jedna kwadratura interpolacyjna rzędu $2n+2$ oparta na węzłach o łącznej krotności $n+1$. Jest to tzw. kwadratura Gaussa – jej węzłami są pierwiastki $(n+1)$-go wielomianu ortogonalnego w przestrzeni $\mathcal{L}_2([a,b],\rho)$.*

**Dowód**: Niech $S$ będzie kwadraturą postaci (10.2), tj.

$$S(f) = \sum_{i=0}^{k} \sum_{j=0}^{m_i-1} A_{ij} f^{(j)}(x_i), \tag{10.4}$$

gdzie $m_0 + \ldots + m_k = n+1$. Ponieważ wiemy, że jest to kwadratura interpolacyjna, zatem jej współczynniki $A_{ij}$ są już ustalone i nie mamy na nie wpływu (zob. przykład 10.1). Nie mamy wpływu również na wartość $n$. Pokażemy teraz jak ustalić wartości $k$, $x_0, \ldots, x_k$ oraz $m_0, \ldots, m_k$, aby kwadratura (10.4) miała maksymalny rząd, tzn. aby równość

$$R_S(f) = I(f) - S(f) = 0$$

zachodziła dla wszystkich wielomianów $f \in \pi_m$ możliwie wysokiego stopnia $m$. Kwadratura $S$ jest interpolacyjna, zatem jej rząd jest nie mniejszy niż $n+1$. Dowolny wielomian $f \in \pi_m$, $m \geqslant n+1$, możemy zapisać jako

$$f(x) = g(x)\, p_{n+1}(x) + r(x),$$

gdzie $p_{n+1} \in \pi_{n+1}$ jest postaci (10.3), $\deg g = m-n-1$ oraz $r \in \pi_n$ to reszta z dzielenia $f$ przez $p_{n+1}$. Stąd

$$\begin{aligned} I(f) - S(f) &= I(g \cdot p_{n+1}) + I(r) - S(g \cdot p_{n+1}) - S(r) = \\ &= I(g \cdot p_{n+1}) = \int_a^b g(x)\, p_{n+1}(x)\, \rho(x)\, dx = \langle g, p_{n+1} \rangle, \end{aligned} \tag{10.5}$$

gdzie $\langle \cdot, \cdot \rangle$ to iloczyn skalarny w przestrzeni $\mathcal{L}_2([a,b],\rho)$. Zauważmy, że przyjmując jako węzły pierwiastki $(n+1)$-go wielomianu ortogonalnego w przestrzeni $\mathcal{L}_2([a,b],\rho)$, wielomian $p_{n+1}$ jest właśnie tym wielomianem ortogonalnym. Jak wynika z ogólnych własności wielomianów ortogonalnych, pierwiastki te są pojedyncze, rzeczywiste i leżą we wnętrzu przedziału $(a,b)$. Z równości (10.5) oraz z własności wielomianów ortogonalnych wynika, że

$$I(f) - S(f) = \langle g, p_{n+1} \rangle = 0, \tag{10.6}$$

dla każdego wielomianu $g \in \pi_n$. To oznacza, że powyższa równość zachodzi dla wszystkich wielomianów $f$ stopnia co najwyżej $2n+1$. Rząd tak skonstruowanej kwadratury jest więc nie mniejszy niż $2n+2$. Z wniosku 10.3 wynika, że rząd ten nie może być większy.

Pozostało do wykazania, że skonstruowana w opisany sposób kwadratura jest jedyną kwadraturą interpolacyjną rzędu $2n+2$ opartą na $n+1$ węzłach. Aby to uzasadnić załóżmy, że kwadratura $S$ oparta na $n+1$ węzłach jest rzędu $2n+2$. Powtarzając powyższe rozumowanie otrzymujemy, że warunek (10.6) ma być spełniony dla dowolnego wielomianu $g \in \pi_n$. Stąd łatwo wywnioskować, że wielomian $p_{n+1}$ musi być $n+1$-wszym wielomianem ortogonalnym w przestrzeni $\mathcal{L}_2([a,b],\rho)$ (zob. dowód warunku W4 w stwierdzeniu 8.6). Ponowne wykorzystanie stwierdzenia 8.6 (tym razem warunku W5) kończy dowód. ∎

#### Własności kwadratury Gaussa

Kwadratura skontruowana w dowodzie twierdzenia 10.4 to tzw. *kwadratura Gaussa*. Jest to kwadratura interpolacyjna rzędu $2n+2$ oparta na pojedynczych $n+1$ węzłach będących pierwiastkami $(n+1)$-go wielomianu ortogonalnego w przestrzeni $\mathcal{L}_2([a,b],\rho)$. Przyjrzyjmy się jej nieco dokładniej.

**Stwierdzenie 10.5 (własności kwadratury Gaussa).** *Niech $S_n$ oznacza kwadraturę Gaussa opartą na węzłach $x_0, \ldots, x_n$. Wówczas:*

**W1.** $S_n(f) = \sum_{i=0}^{n} A_i f(x_i)$, *gdzie* $A_i > 0$, $i = 0, \ldots, n$;

**W2.** *jeżeli* $f \in \mathcal{C}^{2n+2}_{[a,b]}$ *to istnieje liczba* $\xi \in (a,b)$, *taka że*

$$R_{S_n}(f) = \frac{f^{(2n+2)}(\xi)}{(2n+2)!} \int_a^b p_{n+1}^2(x)\,\rho(x)\,dx,$$

*gdzie* $p_{n+1}(x) = \prod_{i=0}^{n}(x - x_i)$;

**W3.** *dla dowolnej funkcji* $f \in \mathcal{C}_{[a,b]}$ *zachodzi*

$$\lim_{n\to\infty} S_n(f) = I(f) = \int_a^b f(x)\,\rho(x)\,dx.$$

**Dowód**: Występująca w warunku W1 postać kwadratury Gaussa wynika natychmiast z przykładu 10.1. Wykażemy, że jej współczynniki $A_0, \ldots, A_n$ są dodatnie. Niech, dla $i = 0, \ldots, n$,

$$\psi_i(x) = \prod_{j\neq i} (x - x_j)^2.$$

Ponieważ $\psi_i \in \pi_{2n}$ oraz $r(S_n) = 2n+2$ zatem $S_n(\psi_i) = I(\psi_i)$, tj.

$$0 < \int_a^b \psi_i(x)\,\rho(x)\,dx = I(\psi_i) = S_n(\psi_i) = \sum_{k=0}^{n} A_k \psi_i(x_k) =$$
$$= A_i \psi_i(x_i) + \sum_{k\neq i} A_k \psi_i(x_k) = A_i \psi_i(x_i).$$

To oznacza, że $A_i > 0$.

Dla dowodu własności W2 przyjmijmy, że $H \in \pi_{2n+1}$ jest wielomianem interpolacyjnym Hermite'a spełniającym następujące warunki

$$H(x_i) = f(x_i), \quad H'(x_i) = f'(x_i), \qquad \text{dla } i = 0, \ldots, n.$$

Stąd, dla $f \in \mathcal{C}^{2n+2}_{[a,b]}$ oraz dla dowolnego $x \in [a,b]$ istnieje liczba $\xi_x \in (a,b)$ taka że

$$f(x) - H(x) = \frac{f^{(2n+2)}(\xi_x)}{(2n+2)!} p_{n+1}^2(x).$$

Możemy zatem zapisać

$$R_{S_n}(f) = I(f) - S_n(f) = I(f) - \sum_{i=0}^{n} A_i f(x_i) = I(f) - \sum_{i=0}^{n} A_i H(x_i) =$$
$$= I(f) - S_n(H) = I(f) - I(H) = \int_a^b \frac{f^{(2n+2)}(\xi_x)}{(2n+2)!} p_{n+1}^2(x)\,\rho(x)\,dx.$$

Na podstawie twierdzenia o wartości średniej dla całki[2] wnioskujemy, że istnieje niezależna od $x$ liczba $\xi \in (a,b)$, dla której

$$R_{S_n}(f) = \int_a^b \frac{f^{(2n+2)}(\xi_x)}{(2n+2)!} p_{n+1}^2(x)\,\rho(x)\,dx = \frac{f^{(2n+2)}(\xi)}{(2n+2)!} \int_a^b p_{n+1}^2(x)\,\rho(x)\,dx.$$

Dowód własności W3 pomijamy (zainteresowanych odsyłam do monografii J. i M. Jankowscy, *Przegląd metod i algorytmów numerycznych. Część 1.*, WNT, Warszawa 1981, str. 92-94). ∎

[2] **Lemat** Jeżeli $f \in \mathcal{C}_{[a,b]}$ oraz $g \geqslant 0$ jest funkcją całkowalną, to istnieje liczba $\xi \in (a,b)$ dla której

$$\int_a^b f(x)\,g(x)\,dx = f(\xi) \int_a^b g(x)\,dx.$$

### 10.3.3. Kwadratury Newtona-Cotes

Kwadratury Newtona-Cotesa to kwadratury interpolacyjne oparte na jednokrotnych węzłach $x_0, \ldots, x_n$ rozłożonych równomiernie na przedziale $[a, b]$, tj.

$$x_i = a + i\frac{b-a}{n}, \qquad \text{dla } i = 0, \ldots, n.$$

Mamy zatem

$$I(f) = \int_a^b f(x)\,dx \approx S(f) = \sum_{i=0}^{n} A_i f\left(a + i\frac{b-a}{n}\right).$$

**Kwadratura trapezów**

Kwadratura trapezów jest kwadraturą Newtona-Cotesa opartą na dwóch węzłach (czyli dla $n = 1$). Łatwo wykazać następujące jej własności (ćwiczenie):

- postać:
$$S(f) = \frac{b-a}{2}\left(f(a) + f(b)\right);$$
- rząd: $r(S) = 2$;
- błąd: jeżeli $f \in \mathcal{C}^2_{[a,b]}$, to istnieje liczba $\xi \in (a, b)$, dla której
$$R_S(f) = I(f) - S(f) = -\frac{f''(\xi)}{12}(b-a)^3.$$

**Kwadratura Simpsona**

Kwadratura Simpsona jest kwadraturą Newtona-Cotesa opartą na trzech węzłach (czyli dla $n = 2$). Łatwo wykazać (ćwiczenie), że jest ona postaci

$$S(f) = \frac{b-a}{6}\left(f(a) + 4f\left(\frac{a+b}{2}\right) + f(b)\right).$$

Ponieważ jest to kwadratura interpolacyjna, zatem $r(S) \geqslant 3$. Dla wielomianu $p_3 \in \pi_3$ postaci

$$p_3(x) = (x-a)\left(x - \frac{a+b}{2}\right)(x-b) \tag{10.7}$$

mamy

$$R_S(p_3) = I(p_3) - S(p_3) = I(p_3) = \int_a^b (x-a)\left(x - \frac{a+b}{2}\right)(x-b)\,dx = 0,$$

zatem $r(S) \geqslant 4$. Z kolei dla wielomianu $p_4 \in \pi_4$

$$p_4(x) = (x-a)\left(x - \frac{a+b}{2}\right)^2(x-b)$$

zachodzi

$$R_S(p_4) = I(p_4) - S(p_4) = I(p_4) = \int_a^b (x-a)\left(x - \frac{a+b}{2}\right)^2(x-b)\,dx =$$
$$= -\frac{(b-a)^5}{120} < 0,$$

zatem $r(S) = 4$.

Wyznaczmy błąd tej kwadratury. Na podstawie definicji mamy

$$R_S(f) = I(f) - S(f) = I(f) - I(L_2), \tag{10.8}$$

gdzie $L_2 \in \pi_2$ to wielomian interpolacyjny Lagrange'a interpolujący funkcję $f$ w węzłach $x_0 = a$, $x_1 = (a+b)/2$, $x_2 = b$. Niech

$$H(x) = L_2(x) + \alpha \cdot p_3(x), \tag{10.9}$$

gdzie $\alpha \in \mathbb{R}$ jest dowolną liczbą, $p_3 \in \pi_3$ to wielomian postaci (10.7). Jest jasne, że warunki

$$H(x_i) = f(x_i), \quad \text{dla } i = 0, 1, 2$$

są spełnione dla wszystkich wartości $\alpha \in \mathbb{R}$. Dobierzmy wartość $\alpha$ w ten sposób, aby dodatkowo

$$H'\left(\frac{a+b}{2}\right) = f'\left(\frac{a+b}{2}\right).$$

Z równości (10.9) otrzymujemy

$$H'\left(\frac{a+b}{2}\right) = L_2'\left(\frac{a+b}{2}\right) + \alpha \cdot p_3'\left(\frac{a+b}{2}\right),$$

skąd wynika, że poszukiwana wartość to

$$\alpha^* = \frac{f'\left(\frac{a+b}{2}\right) - L_2'\left(\frac{a+b}{2}\right)}{p_3'\left(\frac{a+b}{2}\right)}.$$

Uwzględniając (10.9), otrzymujemy dalej z (10.8)

$$\begin{aligned} R_S(f) = I(f) - S(f) = I(f) - I(L_2) = I(f) - I(H - \alpha^* p_3) = \\ = I(f) - I(H) + \alpha^* I(p_3) = I(f - H). \end{aligned}$$

Wielomian $H$ jest wielomianem Hermite'a dla funkcji $f$ opartym na węzłach o łącznej krotności 4. Stąd oraz z twierdzenie o błędzie interpolacji wynika, że dla funkcji $f \in \mathcal{C}^4_{[a,b]}$ mamy równość

$$R_S(f) = \int_a^b \frac{f^{(4)}(\xi_x)}{24}(x-a)\left(x - \frac{a+b}{2}\right)^2 (x-b)\,dx.$$

Wykorzystując twierdzenie o wartości średniej dla całki otrzymujemy ostatecznie postać błędu:

$$R_S(f) = \frac{f^{(4)}(\xi)}{24}\int_a^b (x-a)\left(x - \frac{a+b}{2}\right)^2 (x-b)\,dx = -\frac{f^{(4)}(\xi)}{2880}(b-a)^5,$$

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

**Twierdzenie o rzędzie dla kwadratur Newtona-Cotesa**

Widzimy, że kwadratura Newtona-Cotes oparta na dwóch węzłach jest rzędu 2, a oparta na trzech węzłach ma rząd 4. W przypadku dowolnej liczby węzłów zachodzi następujące twierdzenie, które podamy bez dowodu.

**Twierdzenie 10.6.** *Kwadratura Newtona-Cotesa oparta na $n+1$ węzłach jest rzędu $r$, gdzie*

$$r = \begin{cases} n+1 & \text{dla } n \notin 2\mathbb{N} \\ n+2 & \text{dla } n \in 2\mathbb{N} \end{cases}.$$

## 10.4. Kwadratury złożone

Rozważane w poprzednim rozdziale kwadratury interpolacyjne to przykłady kwadratur prostych. Przy ich pomocy będziemy konstruować tzw. kwadratury złożone.

### 10.4.1. Sposób konstrukcji kwadratur złożonych

Przypuśćmy, że chcemy wyznaczyć przybliżoną wartość całki

$$I(f) = \int_a^b f(x)\,dx.$$

Przedział całkowania $[a,b]$ podzielmy na podprzedziały

$$[a,b] = \bigcup_{i=0}^{n-1} [x_i, x_{i+1}],$$

gdzie $x_i = a + i\dfrac{b-a}{n}$, $i = 0, \ldots, n$. Całkę

$$I_i(f) = \int_{x_i}^{x_{i+1}} f(x)\,dx$$

przybliżamy dowolnie wybraną kwadraturą prostą $S_i(f)$ (np. kwadraturą Gaussa, kwadraturą Newtona-Cotesa)

$$I_i(f) \approx S_i(f), \qquad i = 0, \ldots, n-1.$$

Kwadraturę

$$S(f) := \sum_{i=0}^{n-1} S_i(f)$$

przybliżającą całkę $I(f)$ nazywamy kwadraturą złożoną.

Błąd kwadratury złożonej $S$ to

$$R_S(f) = I(f) - \sum_{i=0}^{n-1} S_i(f) = \sum_{i=0}^{n-1} (I_i(f) - S_i(f)) = \sum_{i=0}^{n-1} R_i(f), \tag{10.10}$$

jest on zatem sumą błędów wszystkich kwadratur prostych tworzących daną kwadraturę złożoną. Z przedstawionej konstrukcji kwadratur złożonych wynika, że na ogół (tj. dla $n \geqslant 2$) nie są one kwadraturami interpolacyjnymi – nie mają więc do nich zastosowania twierdzenia z poprzedniego rozdziału.

### 10.4.2. Przykłady kwadratur złożonych

Ze względu na dużą swobodę podczas wyboru kwadratur prostych tworzących kwadraturę złożoną (każdą z całek $I_i(f)$ możemy przecież przybliżać inną kwadraturą $S_i(f)$) nie będziemy formułować ogólnych twierdzeń dotyczących rzędu oraz błędu kwadratur złożonych. Zamiast tego przeanalizujemy szczegółowo kilka najważniejszych przykładów kwadratur złożonych.

#### Kwadratury złożone prostokątów*

Rozpocznijmy od wyznaczenia kwadratur prostych prostokątów.

Zazwyczaj rozważa się trzy rodzaje kwadratur prostokątów:

$$I(f) = \int_a^b f(x)\,dx \approx S^{(i)}(f) = \begin{cases} f(a)(b-a) & \text{dla } i = 0 \\ f\left(\dfrac{a+b}{2}\right)(b-a) & \text{dla } i = 1/2 \\ f(b)(b-a) & \text{dla } i = 1 \end{cases}.$$

Łatwo stwierdzić[3], że rzędy tych kwadratur to odpowiednio

$$r\left(S^{(0)}\right) = r\left(S^{(1)}\right) = 1$$

oraz

$$r\left(S^{(1/2)}\right) = 2.$$

[3] W tym przypadku (wyjątkowo!) wystarczy dobry rysunek, który może być nawet bardziej pouczający niż zupełnie elementarne obliczenia

Ponadto, dla funkcji $f \in \mathcal{C}^1_{[a,b]}$

$$\begin{aligned} R_{S^{(0)}}(f) = I(f) - S^{(0)}(f) &= \int_a^b f(x)\,dx - f(a)(b-a) = \\ &= \int_a^b f(a) + f'(\xi_x)(x-a)\,dx - f(a)(b-a) = \int_a^b f'(\xi_x)(x-a)\,dx = \\ &= f'(\xi)\int_a^b (x-a)\,dx = \frac{f'(\xi)}{2}(b-a)^2, \quad \text{dla pewnego } \xi \in (a,b). \end{aligned} \tag{10.11}$$

Podobnie pokazujemy (ćwiczenie), że

$$R_{S^{(1)}}(f) = -\frac{f'(\xi)}{2}(b-a)^2$$

oraz dla funkcji $f \in \mathcal{C}^2_{[a,b]}$

$$R_{S^{(1/2)}}(f) = \frac{f''(\xi)}{24}(b-a)^3.$$

Z kwadratur prostych prostokątów możemy utworzyć trzy kwadratury złożone prostokątów: dwie rzędu pierwszego

$$\widetilde{S}_0(f) = \sum_{i=0}^{n-1} S_i^{(0)}(f) = \sum_{i=0}^{n-1} f(x_i)(x_{i+1}-x_i) = h\sum_{i=0}^{n-1} f(x_i)$$

oraz

$$\widetilde{S}_1(f) = \sum_{i=0}^{n-1} S_i^{(1)}(f) = \sum_{i=0}^{n-1} f(x_{i+1})(x_{i+1}-x_i) = h\sum_{i=1}^{n} f(x_i)$$

i jedną rzędu drugiego

$$\widetilde{S}_{1/2}(f) = \sum_{i=0}^{n-1} S_i^{(1/2)}(f) = h\sum_{i=0}^{n-1} f\left(\frac{x_{i+1}+x_i}{2}\right).$$

Wyznaczmy błędy tych kwadratur. Na podstawie zależności (10.10) oraz postaci błędu (10.11) kwadratury prostej otrzymujemy

$$R_{\widetilde{S}_0}(f) = \sum_{i=0}^{n-1} \frac{f'(\xi_i)}{2}\left(\frac{b-a}{n}\right)^2 = \frac{(b-a)^2}{2n}\underbrace{\sum_{i=0}^{n-1}\frac{f'(\xi_i)}{n}}_{f'(\xi)} = \frac{f'(\xi)(b-a)^2}{2n}. \tag{10.12}$$

Podobnie pokazujemy, że

$$R_{\widetilde{S}_1}(f) = -\frac{f'(\xi)(b-a)^2}{2n} \tag{10.13}$$

oraz

$$R_{\widetilde{S}_{1/2}}(f) = \frac{f''(\xi)(b-a)^3}{24n^2}. \tag{10.14}$$

**Kwadratura złożona trapezów**

Przypomnijmy, że kwadratura prosta trapezów ma postać

$$S(f) = \frac{b-a}{2}(f(a)+f(b)),$$

jej błąd to

$$R_S(f) = -\frac{f''(\xi)}{12}(b-a)^3,$$

jest ona rzędu 2. Utworzona z niej kwadratura złożona to

$$\begin{aligned}\widetilde{S}(f) &= \sum_{i=0}^{n-1} \frac{b-a}{2n}\left(f(x_i) + f(x_{i+1})\right) = \\ &= \frac{b-a}{n}\left(\frac{1}{2}f(x_0) + f(x_1) + \ldots + f(x_{n-1}) + \frac{1}{2}f(x_n)\right),\end{aligned}$$

jej błąd ma postać

$$\begin{aligned}R_{\widetilde{S}}(f) &= -\sum_{i=0}^{n-1} \frac{f''(\xi_i)}{12}\left(\frac{b-a}{n}\right)^3 = -\frac{(b-a)^3}{12n^2}\sum_{i=0}^{n-1}\frac{f''(\xi_i)}{n} = \\ &= -\frac{f''(\xi)(b-a)^3}{12n^2}.\end{aligned} \tag{10.15}$$

**Podsumowanie**

Kwadratury złożone nie są kwadraturami interpolacyjnymi, a ich rzędy nie są większe niż rzędy kwadratur prostych, które je tworzą. Posiadają one jednak istotną zaletę: zagęszczanie węzłów (czyli zwiększanie $n$) skutkuje coraz dokładniejszym przybliżaniem wartości całki, i to przy bardzo rozsądnych założeniach dotyczących regularności całkowanej funkcji. Jak wynika ze wzorów (10.12) oraz (10.13), kwadratury złożone prostokątów $\widetilde{S}_0$ oraz $\widetilde{S}_1$ są zbieżne do całki $I(f)$ wraz z $n \to \infty$ już dla funkcji o regularności $\mathcal{C}^1$. W przypadku kwadratur prostokątów $\widetilde{S}_{1/2}$ oraz trapezów, ze wzorów (10.14) oraz (10.15) wynika, że zbieżność ta jest zagwarantowana w klasie $\mathcal{C}^2$.

## Rozdział 11

# Rozwiązywanie układów równań liniowych – metody skończone

Zadanie obliczeniowe, z jakim będziemy się mierzyć w tym rozdziale, polega na znalezieniu, dla zadanych macierzy $A \in \mathbb{R}^{n\times n}$ oraz wektora $b \in \mathbb{R}^n$, rozwiązania $x \in \mathbb{R}^n$ równania

$$Ax = b. \tag{11.1}$$

Zakładamy, że macierz $A$ jest nieosobliwa, zatem rozwiązanie to jest jedyne.

Metody skończone (czyli takie, których realizacja w arytmetyce dokładnej zwraca wynik dokładny w skończonej liczbie kroków) rozwiązywania układów liniowych polegają na rozkładzie (faktoryzacji) macierzy $A$ na iloczyn macierzy, dla których znalezienie rozwiązania wyjściowego zadania staje się łatwiejsze. Najczęściej stosuje się jeden z poniższych typów rozkładów:

- rozkład trójkątno-trójkątny: $A = LU$, gdzie $L, U \in \mathbb{R}^{n\times n}$ to macierze trójkątne,
- rozkład ortogonalno-trójkątny: $A = QR$, gdzie $Q \in \mathbb{R}^{n\times n}$ to macierz ortogonalna, $R \in \mathbb{R}^{n\times n}$ to macierz trójkątna.

Dysponując powyższymi rozkładami macierzy $A$, zadanie rozwiązania równania (11.1) zastępujemy dwoma prostszymi do rozwiązania:
dla rozkładu $A = LU$

$$Ax = b \quad \Leftrightarrow \quad \begin{cases} Ly = b \\ Ux = y \end{cases},$$

dla rozkładu $A = QR$

$$Ax = b \quad \Leftrightarrow \quad \begin{cases} Qy = b \\ Rx = y \end{cases}.$$

Podamy teraz numerycznie poprawne algorytmy rozwiązywania układów równań liniowych o macierzach trójkątnych oraz ortogonalnych.

## 11.1. Rozwiązywanie układów równań o macierzach o szczególnej postaci

### 11.1.1. Rozwiązywanie układów o macierzy trójkątnej

Niech $U = [u_{ij}] \in \mathbb{R}^{n\times n}$ będzie nieosobliwą macierzą trójkątną górną, tj. $u_{ij} = 0$, dla $i > j$ oraz $u_{ii} \neq 0$ $(i = 1, \ldots, n)$; niech $b = (b_1, \ldots, b_n)^T \in \mathbb{R}^n$.

Układ równań $Ux = b$ ze względu na niewiadomą $x = (x_1, \ldots, x_n)^T \in \mathbb{R}^n$ możemy zapisać jako

$$\left.\begin{aligned} u_{11}x_1 + u_{12}x_2 + \cdots + u_{1n}x_n &= b_1 \\ &\vdots \\ u_{ii}x_i + \cdots + u_{in}x_n &= b_i \\ &\vdots \\ u_{nn}x_n &= b_n \end{aligned}\right\}. \tag{11.2}$$

**Algorytm**

Z postaci (11.2) układu wynika następujący bardzo naturalny algorytm rozwiązywania układów liniowych o macierzy trójkątnej:

$$x_n = b_n/u_{nn};$$
$$\texttt{for } i = n-1, \ldots, 1: \quad x_i = \left(b_i - \sum_{k=i+1}^{n} u_{ik}x_k\right)/u_{ii}.$$

**Koszt**

Wyliczenie $x_i$ to $n - i + 1$ iloczynów (łącznie z jednym dzieleniem) oraz $n - i$ dodawań: łącznie $2n - 2i + 1$ operacji arytmetycznych. Koszt wyliczenia rozwiązania $x$ to zatem

$$\sum_{i=1}^{n} (2n - 2i + 1) = n^2$$

operacji arytmetycznych.

**Właściwości numeryczne**

Niech $\widetilde{x} = (\widetilde{x}_1, \ldots, \widetilde{x}_n) = \mathrm{fl}\,(x)$. Mamy

$$\widetilde{x}_n = \mathrm{fl}\left(\frac{b_n}{u_{nn}}\right) = \frac{b_n}{u_{nn}\,(1 + \varepsilon_{nn})},$$

gdzie $|\varepsilon_{nn}| \leqslant 2^{-t}$ oraz

$$\widetilde{x}_i = \mathrm{fl}\left(\frac{b_i - \sum_{k=i+1}^{n} u_{ik}\widetilde{x}_k}{u_{ii}}\right) = \frac{b_i - \sum_{k=i+1}^{n} u_{ik}\widetilde{x}_k\,(1 + \varepsilon_{ik})}{u_{ii}\,(1 + \varepsilon_{ii})},$$

gdzie $\varepsilon_{ik}$ to błędy powstałe przy wyliczaniu iloczynu skalarnego (zob. przykład 1.12), $|\varepsilon_{ik}| \leqslant_1 (n - k + 2)\,2^{-t}$ oraz $\varepsilon_{ii}$ to błąd odjęcia $b_i$ oraz podzielenia przez $u_{ii}$, $|\varepsilon_{ii}| \leqslant_1 2 \cdot 2^{-t}$, dla $i = 1, \ldots, n-1$, $k = i+1, \ldots, n$.

Oznacza to, że

$$\widetilde{x}_i = \frac{b_i - \sum_{k=i+1}^{n} (u_{ik} + u_{ik}\varepsilon_{ik})\,\widetilde{x}_k}{u_{ii} + u_{ii}\varepsilon_{ii}},$$

czyli otrzymane w arytmetyce fl rozwiązanie $\widetilde{x}$ spełnia układ równań

$$(U + \triangle U)\,\widetilde{x} = b,$$

gdzie $\triangle U = [u_{ik}\varepsilon_{ik}] \in \mathbb{R}^{n\times n}$ jest macierzą zaburzeń oraz

$$\|\triangle U\|_1 = \max_j \sum_i |u_{ij}\varepsilon_{ij}| \leqslant_1 K(n)\,2^{-t} \max_j \sum_i |u_{ij}| = K(n)\,2^{-t}\,\|U\|_1,$$

gdzie $K(n)$ jest stałą rzędu $n$. To oznacza, że uzyskane w arytmetyce fl rozwiązanie $\widetilde{x}$ jest rozwiązaniem dokładnym zadania o nieco zaburzonej macierzy $U$. Zaproponowany algorytm jest więc numerycznie poprawny.

### 11.1.2. Rozwiązywanie układów o macierzy ortogonalnej

Niech $Q = [q_{ij}] \in \mathbb{R}^{n\times n}$ będzie macierzą ortogonalną, tj. $Q^T Q = I_n$. Rozwiązanie układ równań $Qx = b$ możemy wówczas zapisać jako

$$x = Q^T b.$$

**Algorytm**

Wynika stąd następujący naturalny algorytm rozwiązywania układów liniowych o macierzy ortogonalnej:

$$\texttt{for } i = 1, \ldots, n: \quad x_i = \sum_{k=1}^{n} q_{ki} b_k.$$

**Koszt**

Wyliczenie $x_i$ to $n$ iloczynów oraz $n-1$ dodawań: łącznie $2n-1$ operacji arytmetycznych. Całkowity koszt wyliczenia rozwiązania $x$ nie przekracza zatem $2n^2 - n$ operacji arytmetycznych.

**Właściwości numeryczne**

Niech $\widetilde{x} = (\widetilde{x}_1, \ldots, \widetilde{x}_n) = \mathrm{fl}\,(x)$. Mamy

$$\widetilde{x}_i = \mathrm{fl}\left(\sum_{k=1}^{n} q_{ki} b_k\right) = \sum_{k=1}^{n} q_{ki} b_k \,(1 + \varepsilon_{ki}) = \sum_{k=1}^{n} \left(q_{ki} + q_{ki}\varepsilon_{ki}\right) b_k,$$

gdzie $|\varepsilon_{ik}| \leqslant_1 (n-k+2)\, 2^{-t}$, dla $i, k = 1, \ldots, n$.

Oznacza to, że otrzymane w arytmetyce fl rozwiązanie $\widetilde{x}$ spełnia układ równań

$$(Q + \triangle Q)\, \widetilde{x} = b,$$

gdzie $\triangle Q = [q_{ik}\varepsilon_{ik}] \in \mathbb{R}^{n\times n}$ jest macierzą zaburzeń oraz

$$\|\triangle Q\|_1 = \max_j \sum_i |q_{ij}\varepsilon_{ij}| \leqslant_1 K(n)\, 2^{-t} \max_j \sum_i |q_{ij}| = K(n)\, 2^{-t} \|Q\|_1,$$

gdzie $K(n)$ jest stałą rzędu $n$. To oznacza, że uzyskane w arytmetyce fl rozwiązanie $\widetilde{x}$ jest rozwiązaniem dokładnym zadania o nieco zaburzonej macierzy $Q$. Zaproponowany algorytm jest więc numerycznie poprawny.

## 11.2. Metoda eliminacji Gaussa

Niech $A = [a_{ij}] \in \mathbb{R}^{n\times n}$ będzie macierzą nieosobliwą. Naszym celem będzie sprowadzenie macierzy $A$ do postaci trójkątnej górnej przy pomocy tzw. operacji elementarnych:

- zmiany kolejności wierszy lub kolumn,
- dodawania do dowolnie wybranego wiersza (kolumny) kombinacji liniowej pozostałych wierszy (kolumn).

### 11.2.1. Realizacja metody eliminacji

Niech $A^{(1)} := A$ oraz niech $w_l^{(k)}$ będzie $l$-tym wierszem macierzy $A^{(k)} = [a_{ij}^{(k)}]$ powstałej w $k$-tym kroku metody eliminacji, tj.

$$A^{(k)} = \begin{pmatrix} w_1^{(k)} \\ \vdots \\ w_n^{(k)} \end{pmatrix}.$$

Mamy kolejno:

$$A = A^{(1)} = \begin{pmatrix} a_{11}^{(1)} & a_{12}^{(1)} & \cdots & a_{1n}^{(1)} \\ a_{21}^{(1)} & a_{22}^{(1)} & \cdots & a_{2n}^{(1)} \\ \vdots & \vdots & & \vdots \\ a_{n1}^{(1)} & a_{n2}^{(1)} & \cdots & a_{nn}^{(1)} \end{pmatrix} \rightsquigarrow \left| \begin{array}{l} w_1^{(2)} = w_1^{(1)}; \\ \text{dla } i = 2, \ldots, n \\ w_i^{(2)} = w_i^{(1)} - \frac{a_{i1}^{(1)}}{a_{11}^{(1)}} w_1^{(1)} \end{array} \right| \rightsquigarrow$$

$$\rightsquigarrow A^{(2)} = \begin{pmatrix} a_{11}^{(1)} & a_{12}^{(1)} & \cdots & a_{1n}^{(1)} \\ 0 & a_{22}^{(2)} & \cdots & a_{2n}^{(2)} \\ \vdots & \vdots & & \vdots \\ 0 & a_{n2}^{(2)} & \cdots & a_{nn}^{(2)} \end{pmatrix} \rightsquigarrow \left| \begin{array}{l} \text{dla } i = 1, 2 : w_i^{(3)} = w_i^{(2)}; \\ \text{dla } i = 3, \ldots, n \\ w_i^{(3)} = w_i^{(2)} - \frac{a_{i2}^{(2)}}{a_{22}^{(2)}} w_2^{(2)} \end{array} \right| \rightsquigarrow$$

$$\rightsquigarrow A^{(3)} = \begin{pmatrix} a_{11}^{(1)} & a_{12}^{(1)} & a_{13}^{(1)} & \cdots & a_{1n}^{(1)} \\ 0 & a_{22}^{(2)} & a_{23}^{(2)} & \cdots & a_{2n}^{(2)} \\ 0 & 0 & a_{33}^{(3)} & \cdots & a_{3n}^{(3)} \\ \vdots & \vdots & \vdots & & \vdots \\ 0 & 0 & a_{n3}^{(3)} & \cdots & a_{nn}^{(3)} \end{pmatrix} \rightsquigarrow \ldots \rightsquigarrow$$

$$\rightsquigarrow A^{(n)} = \begin{pmatrix} a_{11}^{(1)} & a_{12}^{(1)} & a_{13}^{(1)} & \cdots & a_{1n}^{(1)} \\ 0 & a_{22}^{(2)} & a_{23}^{(2)} & \cdots & a_{2n}^{(2)} \\ 0 & 0 & \ddots & \ddots & a_{3n}^{(3)} \\ \vdots & \vdots & \ddots & \ddots & \vdots \\ 0 & 0 & \cdots & 0 & a_{nn}^{(n)} \end{pmatrix}$$

Krok $k$-ty metody eliminacji ($k = 1, \ldots, n-1$) możemy zapisać w następujący sposób

$$A^{(k)} \rightsquigarrow \left| \begin{array}{l} \text{dla } i = 1, \ldots, k : w_i^{(k+1)} = w_i^{(k)}; \\ \text{dla } i = k+1, \ldots, n \\ w_i^{(k+1)} = w_i^{(k)} - \frac{a_{ik}^{(k)}}{a_{kk}^{(k)}} w_k^{(k)} \end{array} \right| \rightsquigarrow A^{(k+1)}.$$

### 11.2.2. Zapis macierzowy metody eliminacji

Zauważmy, że krok pierwszy, tj. przejście $A^{(1)} \rightsquigarrow A^{(2)}$ możemy zapisać jako iloczyn

$$A^{(2)} = L_1 A^{(1)},$$

gdzie $L_1 \in \mathbb{R}^{n \times n}$ jest macierzą trójkątną dolną postaci

$$L_1 = \begin{pmatrix} 1 & & & \\ m_{21} & \ddots & & \\ \vdots & & \ddots & \\ m_{n1} & & & 1 \end{pmatrix}, \qquad m_{i1} = -\frac{a_{i1}^{(1)}}{a_{11}^{(1)}}; \tag{11.3}$$

wszystkie niewypisane elementy macierzy $L_1$ są zerami. Podobnie, $k$-ty krok metody eliminacji czyli przejście $A^{(k)} \rightsquigarrow A^{(k+1)}$ możemy zapisać jako

$$A^{(k+1)} = L_k A^{(k)},$$

gdzie

$$A^{(k)} = \begin{pmatrix} a_{11}^{(1)} & \cdots & \cdots & \cdots & a_{1n}^{(1)} \\ & \ddots & & & \vdots \\ & & a_{kk}^{(k)} & \cdots & a_{kn}^{(k)} \\ & & \vdots & & \vdots \\ & & a_{nk}^{(k)} & \cdots & a_{nn}^{(k)} \end{pmatrix},$$

a $L_k \in \mathbb{R}^{n\times n}$ jest macierzą trójkątną dolną

$$L_k = \begin{pmatrix} 1 & & & & \\ & \ddots & & & \\ & & 1 & & \\ & & m_{k+1,k} & \ddots & \\ & & \vdots & & \ddots \\ & & m_{n,k} & & 1 \end{pmatrix}, \qquad m_{ik} = -\frac{a_{ik}^{(k)}}{a_{kk}^{(k)}}; \tag{11.4}$$

wszystkie niewypisane elementy macierzy $L_k$ są zerami. Cały proces eliminacji w zapisie macierzowym wygląda więc tak

$$U = A^{(n)} = L_{n-1}A^{(n-1)} = L_{n-1}L_{n-2}A^{(n-2)} = \cdots = L_{n-1}\cdot\ldots\cdot L_1 A, \tag{11.5}$$

gdzie $L_1, \ldots, L_{n-1} \in \mathbb{R}^{n\times n}$ są macierzami postaci (11.4): są to macierze trójkątne dolne z jedynkami na przekątnej.

**Stwierdzenie 11.1 (własności macierzy $L_k$).** *Przy powyższych oznaczeniach*
**W1.** $L_k = I_n + m_k e_k^T$, *gdzie* $m_k \in \mathbb{R}^n$ *jest wektorem postaci*

$$m_k = (\underbrace{0, \ldots, 0}_{\times k}, m_{k+1,k}, \ldots, m_{n,k})^T,$$

*a* $e_k \in \mathbb{R}^n$ *to* $k$*-ty wektor bazy kanonicznej przestrzeni* $\mathbb{R}^n$.
**W2.** $L_k^{-1} = I_n - m_k e_k^T$
**W3.** $L_1^{-1}L_2^{-1}\cdots L_{n-1}^{-1} = L$, *gdzie*

$$L = \begin{pmatrix} 1 & & & \\ -m_{21} & \ddots & & \\ \vdots & \ddots & \ddots & \\ -m_{n1} & \cdots & -m_{n,n-1} & 1 \end{pmatrix}, \qquad m_{ik} = -\frac{a_{ik}^{(k)}}{a_{kk}^{(k)}}.$$

**Dowód**: Własność W1 wynika z faktu, że dla dowolnego wektora $v \in \mathbb{R}^n$ macierz $ve_k^T$ jest macierzą wymiaru $n \times n$, której $k$-tą kolumną są współrzędne wektora $v$, pozostałe kolumny są wektorami zerowymi.

Własność W2 wynika z prostych przekształceń:

$$L_k\left(I_n - m_k e_k^T\right) = \left(I_n + m_k e_k^T\right)\left(I_n - m_k e_k^T\right) = I_n - m_k \underbrace{e_k^T m_k}_{=0} e_k^T = I_n.$$

Podobnie uzasadniamy własność W3:

$$\begin{aligned} L_1^{-1}L_2^{-1}\cdots L_{n-1}^{-1} &= \left(I_n - m_1 e_1^T\right)\left(I_n - m_2 e_2^T\right)\cdots\left(I_n - m_{n-1}e_{n-1}^T\right) = \\ &= I_n - m_1 e_1^T - m_2 e_2^T - \ldots - m_{n-1}e_{n-1}^T + \\ &\quad + \sum_{i<j} m_i e_i^T m_j e_j^T + \ldots + (-1)^{n-1} m_1 e_1^T \cdot\ldots\cdot m_{n-1}e_{n-1}^T = \\ &= I_n - m_1 e_1^T - m_2 e_2^T - \ldots - m_{n-1}e_{n-1}^T = L. \end{aligned}$$

■

Z równości (11.5), uwzględniając oznaczenia ze stwierdzenia 11.1,

$$(L_{n-1}L_{n-2}\cdots L_1)^{-1} = L_1^{-1}L_2^{-1}\cdots L_{n-1}^{-1} = L,$$

otrzymujemy poszukiwany rozkład trójkątno-trójkątny macierzy $A$

$$A = LU.$$

**Przykład 11.1.** *Dla macierzy*

$$A = \begin{pmatrix} 1 & 0 & 1 \\ -2 & 1 & 0 \\ 2 & 3 & 0 \end{pmatrix},$$

*stosując metodę eliminacji Gaussa, wyznaczymy jej rozkład LU.*

*W pierwszym kroku wyznaczamy macierz $L_1$ odpowiedzialną za wyzerowanie elementów pierwszej kolumny macierzy $A^{(1)} = A$ stojących pod jej przekątną. Na podstawie wzoru (11.3) mamy*

$$L_1 = \begin{pmatrix} 1 & 0 & 0 \\ 2 & 1 & 0 \\ -2 & 0 & 1 \end{pmatrix},$$

*a stąd*

$$A^{(2)} = L_1 A^{(1)} = \begin{pmatrix} 1 & 0 & 0 \\ 2 & 1 & 0 \\ -2 & 0 & 1 \end{pmatrix} \begin{pmatrix} 1 & 0 & 1 \\ -2 & 1 & 0 \\ 2 & 3 & 0 \end{pmatrix} = \begin{pmatrix} 1 & 0 & 1 \\ 0 & 1 & 2 \\ 0 & 3 & -2 \end{pmatrix}.$$

*W kolejnym kroku wyznaczamy macierz $L_2$ odpowiedzialną za wyzerowanie elementu stojącego pod przekątną w drugiej kolumnie macierzy $A^{(2)}$. Na podstawie (11.4) mamy*

$$L_2 = \begin{pmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & -3 & 1 \end{pmatrix}$$

*oraz*

$$A^{(3)} = L_2 A^{(2)} = \begin{pmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & -3 & 1 \end{pmatrix} \begin{pmatrix} 1 & 0 & 1 \\ 0 & 1 & 2 \\ 0 & 3 & -2 \end{pmatrix} = \begin{pmatrix} 1 & 0 & 1 \\ 0 & 1 & 2 \\ 0 & 0 & -8 \end{pmatrix} = U.$$

*Na podstawie stwierdzenia 11.1 wnioskujemy, że poszukiwana macierz L to*

$$L = L_1^{-1} L_2^{-1} = \begin{pmatrix} 1 & 0 & 0 \\ -2 & 1 & 0 \\ 2 & 3 & 1 \end{pmatrix}.$$

*Ostatecznie,*

$$A = \begin{pmatrix} 1 & 0 & 1 \\ -2 & 1 & 0 \\ 2 & 3 & 0 \end{pmatrix} = \begin{pmatrix} 1 & 0 & 0 \\ -2 & 1 & 0 \\ 2 & 3 & 1 \end{pmatrix} \begin{pmatrix} 1 & 0 & 1 \\ 0 & 1 & 2 \\ 0 & 0 & -8 \end{pmatrix} = LU.$$

### 11.2.3. Koszt i właściwości numeryczne

#### Koszt

Wyliczenie wszystkich mnożników $m_{ik}$ wykorzystywanych w kolejnych krokach metody to koszt $(n^2 - n)/2$ operacji arytmetycznych (jest ich właśnie tyle, a koszt wyliczenia każdego z nich to jedno dzielenie). Koszt pierwszego kroku to dodatkowo $2(n-1)^2$ operacji arytmetycznych: mamy do wyznaczenia $(n-1)^2$ elementów, każdy kosztem jednego mnożenia i jednego odejmowania. Ogólnie, koszt $k$-tego kroku (dla $k = 1, \ldots, n-1$) to $2(n-k)^2$ operacje arytmetyczne: wyznaczanie $(n-k)^2$ elementów, każdy kosztem jednego mnożenia i jednego odejmowania. Całkowity koszt metody eliminacji Gaussa nie przekracza zatem[1]

$$\frac{n^2 - n}{2} + 2\sum_{k=1}^{n-1} k^2 = \frac{2}{3}n^3 + \mathcal{O}\left(n^2\right)$$

operacji arytmetycznych.

[1] Przypomnijmy, że dla $n \in \mathbb{N}$:

$$1^2 + 2^2 + \ldots + n^2 = \frac{n(n+1)(2n+1)}{6}.$$

**Właściwości numeryczne**

W przedstawionej wersji metoda eliminacji Gaussa posiada niestety istotne wady:

- wymaga założenia gwarantującego, że $a_{kk}^{(k)} \neq 0$ dla $k = 1, \ldots, n-1$; warunek ten nie wynika z założonej nieosobliwości macierzy $A$,
- współczynniki $a_{ij}^{(k)}$ (czyli elementy macierzy $U$) oraz mnożniki $m_{ik}$ (czyli elementy macierzy $L$) mogą bardzo wzrastać w kolejnych krokach potęgując błędy wynikające z prowadzenia obliczeń w arytmetyce fl,
- w podanej wersji metoda eliminacji Gaussa nie jest akceptowalna numerycznie w całej klasie macierzy nieosobliwych; jest natomiast numerycznie poprawna (!) w klasie macierzy symetrycznych określonych (dodatnio lub ujemnie) oraz w klasie macierzy o silnie dominującej przekątnej; zob. twierdzenia 5.2.8 oraz 5.2.9 w monografii A. Kiełbasiński, H. Schwetlick, *Numeryczna algebra liniowa. Wprowadzenie do obliczeń zautomatyzowanych*, WNT, Warszawa 1992.

W celu zapewnienia numerycznej poprawności metody eliminacji Gaussa w całej klasie macierzy nieosobliwych ustalonego wymiaru $n \times n$ wprowadza się pewną modyfikację gwarantującą, że $|m_{ik}| \leqslant 1$ (w konsekwencji również $|\mathrm{fl}\,(m_{ik})| \leqslant_1 1$). Modyfikacja ta polega na odpowiednim *wyborze elementu głównego* (WEG) w każdym kroku eliminacji. Metoda eliminacji Gaussa z WEG pozbawiona będzie wszystkich wspomnianych wad, a jej koszt będzie taki sam, jak koszt zwykłej metody eliminacji (wybór elementu głównego nie wymaga wykonania żadnej dodatkowej operacji arytmetycznej!).

### 11.2.4. Wybór elementu głównego

Wybór elementu głównego w metodzie eliminacji Gaussa polega na takim przestawianiu wierszy w macierzach otrzymywanych w kolejnych krokach eliminacji, aby za każdym razem element $a_{kk}^{(k)}$ był elementem o największym module spośród wszystkim elementów kolumny $w_k^{(k)}$; na podstawie założonej nieosobliwości macierzy $A$, wszystkie tak wybrane elementy są różne od zera. Przypomnijmy, że w zapisie macierzowym operacja zamiany kolejności wierszy to przemnożenie lewostronne macierzy przez odpowiednio dobraną macierz permutacji.

Metodę eliminacji Gaussa z wyborem elementu głównego możemy schematycznie zapisać jako

$$A = A^{(1)} \to P_1 A^{(1)} \to L_1 P_1 A^{(1)} = A^{(2)} \to P_2 A^{(2)} \to L_2 P_2 A^{(2)} = A^{(3)} \to \ldots,$$

gdzie, dla $k = 1, \ldots, n-1$,

- $P_k \in \mathbb{R}^{n \times n}$ oznacza macierz permutacji otrzymaną z macierzy jednostkowej $I_n$ przez zamianę miejscami jej dwóch wierszy: wiersza o indeksie $k$ z pewnym wierszem o indeksie $i_k \geqslant k$,
- $L_k$ to macierz postaci (11.4) odpowiedzialna za zerowanie elementów w $k$-tej kolumnie macierzy $P_k A^{(k)}$.

Macierzowy zapis opisanego powyżej procesu eliminacji Gaussa z WEG prowadzi do rozkładu

$$U = L_{n-1} P_{n-1} L_{n-2} P_{n-2} \cdot \ldots \cdot L_2 P_2 L_1 P_1 A.$$

Zauważmy, że

$$\begin{aligned}
U &= L_{n-1} P_{n-1} L_{n-2} P_{n-2} \cdot \ldots \cdot L_2 P_2 L_1 P_1 A = \\
&= L_{n-1} P_{n-1} L_{n-2} P_{n-2} \cdot \ldots \cdot L_2 P_2 L_1 \Big( \prod_{i=2}^{n-1} P_i \Big) \Big( \prod_{i=n-1}^{2} P_i \Big) P_1 A = \\
&= L_{n-1} P_{n-1} L_{n-2} P_{n-2} \cdot \ldots \cdot L_2 \Big( \prod_{i=3}^{n-1} P_i \Big) \Big( \prod_{i=n-1}^{3} P_i \Big) P_2 L_1 \Big( \prod_{i=2}^{n-1} P_i \Big) \Big( \prod_{i=n-1}^{1} P_i \Big) A = \\
&= L_{n-1} P_{n-1} L_{n-2} P_{n-2} \cdot \ldots \cdot L_2 \Big( \prod_{i=3}^{n-1} P_i \Big) \Big( \prod_{i=n-1}^{2} P_i \Big) L_1 \Big( \prod_{i=2}^{n-1} P_i \Big) \Big( \prod_{i=n-1}^{1} P_i \Big) A.
\end{aligned}$$

Kontynuując powyższe rozumowanie dochodzimy do rozkładu

$$U = L_{n-1}L^{\star}_{n-2} \cdot \cdots \cdot L^{\star}_1 P^T A, \tag{11.6}$$

gdzie $P^T = P_{n-1}P_{n-2}\cdots P_1$ oraz dla $k = 1, \ldots, n-2$

$$L^{\star}_k = \Big(\prod_{i=n-1}^{k+1} P_i\Big) L_k \Big(\prod_{i=k+1}^{n-1} P_i\Big).$$

Na podstawie stwierdzenia 11.1 możemy zapisać

$$L^{\star}_k = \Big(\prod_{i=n-1}^{k+1} P_i\Big)(I_n + m_k e_k^T)\Big(\prod_{i=k+1}^{n-1} P_i\Big) = I_n + \Big(\prod_{i=n-1}^{k+1} P_i\Big) m_k e_k^T \Big(\prod_{i=k+1}^{n-1} P_i\Big),$$

a ponieważ macierze

$$\prod_{i=n-1}^{k+1} P_i \quad \text{oraz} \quad \prod_{i=k+1}^{n-1} P_i$$

to macierze permutacji, które zamieniają kolejnością wybrane wiersze (przy mnożeniu przez nie z lewej strony) oraz kolumny (przy mnożeniu z prawej strony) o indeksach $i_{p_1}, \ldots, i_{p_{n-k-1}} \in \{k+1, \ldots, n\}$, zatem

$$e_k^T \Big(\prod_{i=k+1}^{n-1} P_i\Big) = e_k^T;$$

z kolei wektor

$$\Big(\prod_{i=n-1}^{k+1} P_i\Big) m_k$$

powstaje z wektora $m_k$ poprzez przestawienie pewnych jego współrzędnych, ale początkowe $k$ jego zerowych współczynników pozostaje bez zmiany. W efekcie każda z macierzy $L^{\star}_k$ ma dokładnie taką samą strukturę jak macierz $L_k$. Stąd, z (11.6) oraz na podstawie stwierdzenia 11.1 otrzymujemy ostatecznie rozkład ortogonalno-trójkątno-trójkątny macierzy $A$:

$$A = PLU, \tag{11.7}$$

gdzie $P = P_1P_2\cdots P_{n-1}$ jest macierzą ortogonalną, $L = (L^{\star}_1)^{-1}(L^{\star}_2)^{-1}\cdots(L^{\star}_{n-2})^{-1}L^{-1}_{n-1}$ jest macierzą trójkątną dolną z jedynkami na przekątnej, $U$ jest macierzą trójkątną górną.

**Przykład 11.2.** *Rozważmy ponownie macierz $A$ z przykładu 11.1, tj.*

$$A = \begin{pmatrix} 1 & 0 & 1 \\ -2 & 1 & 0 \\ 2 & 3 & 0 \end{pmatrix}.$$

*Wyznaczymy jej rozkład $PLU$.*

*Rozpoczynamy od przemnożenia lewostronnie macierzy $A$ przez macierz permutacji $P_1$,*

$$P_1 = \begin{pmatrix} 0 & 1 & 0 \\ 1 & 0 & 0 \\ 0 & 0 & 1 \end{pmatrix},$$

*odpowiedzialną za zamianę miejscami dwóch pierwszych wierszy macierzy $A$. Otrzymujemy*

$$P_1 A = \begin{pmatrix} 0 & 1 & 0 \\ 1 & 0 & 0 \\ 0 & 0 & 1 \end{pmatrix} \begin{pmatrix} 1 & 0 & 1 \\ -2 & 1 & 0 \\ 2 & 3 & 0 \end{pmatrix} = \begin{pmatrix} -2 & 1 & 0 \\ 1 & 0 & 1 \\ 2 & 3 & 0 \end{pmatrix}.$$

*Wyznaczamy teraz macierz $L_1$ odpowiedzialną za wyzerowanie elementów stojących pod przekątną w pierwszej kolumnie macierzy $P_1A$. Na podstawie wzoru (11.3) mamy*

$$L_1 = \begin{pmatrix} 1 & 0 & 0 \\ 1/2 & 1 & 0 \\ 1 & 0 & 1 \end{pmatrix},$$

*a stąd*

$$A^{(2)} = L_1P_1A = \begin{pmatrix} 1 & 0 & 0 \\ 1/2 & 1 & 0 \\ 1 & 0 & 1 \end{pmatrix} \begin{pmatrix} -2 & 1 & 0 \\ 1 & 0 & 1 \\ 2 & 3 & 0 \end{pmatrix} = \begin{pmatrix} -2 & 1 & 0 \\ 0 & 1/2 & 1 \\ 0 & 4 & 0 \end{pmatrix}.$$

*Macierz $A^{(2)}$ mnożymy lewostronnie przez macierz permutacji $P_2$,*

$$P_2 = \begin{pmatrix} 1 & 0 & 0 \\ 0 & 0 & 1 \\ 0 & 1 & 0 \end{pmatrix},$$

*otrzymując*

$$P_2A^{(2)} = \begin{pmatrix} 1 & 0 & 0 \\ 0 & 0 & 1 \\ 0 & 1 & 0 \end{pmatrix} \begin{pmatrix} -2 & 1 & 0 \\ 0 & 1/2 & 1 \\ 0 & 4 & 0 \end{pmatrix} = \begin{pmatrix} -2 & 1 & 0 \\ 0 & 4 & 0 \\ 0 & 1/2 & 1 \end{pmatrix}.$$

*Ostatni krok, to wyznaczenie macierzy $L_2$ zerującej element drugiej kolumny macierzy $P_2A^{(2)}$ stojący pod jej przekątną. Zgodnie z* (11.4) *mamy*

$$L_2 = \begin{pmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & -1/8 & 1 \end{pmatrix},$$

*a stąd*

$$A^{(3)} = L_2P_2A^{(2)} = \begin{pmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & -1/8 & 1 \end{pmatrix} \begin{pmatrix} -2 & 1 & 0 \\ 0 & 4 & 0 \\ 0 & 1/2 & 1 \end{pmatrix} = \begin{pmatrix} -2 & 1 & 0 \\ 0 & 4 & 0 \\ 0 & 0 & 1 \end{pmatrix}.$$

*Podsumowując,*

$$U = A^{(3)} = L_2P_2L_1P_1A. \tag{11.8}$$

*Zauważmy, że*

$$L_2P_2L_1P_1 = L_2P_2L_1P_2P_2P_1 = L_2L_1^{\star}P_2P_1,$$

*gdzie macierz $L_1^{\star} = P_2L_1P_2$ to macierz trójkątna dolna powstała z macierzy $L_1$ przez zamianę miejscami jej dwóch ostatnich wierszy (lewostronne przemnożenie macierzy $L_1$ przez macierz permutacji $P_2$) oraz przez zamianę miejscami dwóch ostatnich kolumn macierzy $P_2L_1$ (prawostronne przemnożenie macierzy $P_2L_1$ przez macierz permutacji $P_2$) tj.*

$$L_1^{\star} = P_2L_1P_2 = \begin{pmatrix} 1 & 0 & 0 \\ 0 & 0 & 1 \\ 0 & 1 & 0 \end{pmatrix} \begin{pmatrix} 1 & 0 & 0 \\ 1/2 & 1 & 0 \\ 1 & 0 & 1 \end{pmatrix} \begin{pmatrix} 1 & 0 & 0 \\ 0 & 0 & 1 \\ 0 & 1 & 0 \end{pmatrix} =$$
$$= \begin{pmatrix} 1 & 0 & 0 \\ 1 & 0 & 1 \\ 1/2 & 1 & 0 \end{pmatrix} \begin{pmatrix} 1 & 0 & 0 \\ 0 & 0 & 1 \\ 0 & 1 & 0 \end{pmatrix} = \begin{pmatrix} 1 & 0 & 0 \\ 1 & 1 & 0 \\ 1/2 & 0 & 1 \end{pmatrix}.$$

*Z otrzymanej równości (11.8) wynika, że*

$$A = PLU,$$

*gdzie*

$$P = (P_2P_1)^{-1} = P_1^{-1}P_2^{-1} = P_1P_2 = \begin{pmatrix} 0 & 1 & 0 \\ 1 & 0 & 0 \\ 0 & 0 & 1 \end{pmatrix} \begin{pmatrix} 1 & 0 & 0 \\ 0 & 0 & 1 \\ 0 & 1 & 0 \end{pmatrix} = \begin{pmatrix} 0 & 0 & 1 \\ 1 & 0 & 0 \\ 0 & 1 & 0 \end{pmatrix}$$

*oraz, na podstawie stwierdzenia 11.1,*

$$L = (L_2L_1^\star)^{-1} = (L_1^\star)^{-1}L_2^{-1} = \begin{pmatrix} 1 & 0 & 0 \\ -1 & 1 & 0 \\ -1/2 & 1/8 & 1 \end{pmatrix}.$$

*Ostatecznie, poszukiwane macierze tworzące rozkład $A = PLU$ to*

$$P = \begin{pmatrix} 0 & 0 & 1 \\ 1 & 0 & 0 \\ 0 & 1 & 0 \end{pmatrix}, \quad L = \begin{pmatrix} 1 & 0 & 0 \\ -1 & 1 & 0 \\ -1/2 & 1/8 & 1 \end{pmatrix}, \quad U = \begin{pmatrix} -2 & 1 & 0 \\ 0 & 4 & 0 \\ 0 & 0 & 1 \end{pmatrix}.$$

**Koszt i właściwości numeryczne**

Ponieważ wybór elementu głównego nie wymaga wykonania żadnego dodatkowego działania arytmetycznego[2], zatem koszt metody eliminacji Gaussa z WEG to $2n^3/3 + \mathcal{O}\left(n^2\right)$ operacji arytmetycznych. Można wykazać, że opisany algorytm jest numerycznie poprawny w klasie macierzy nieosobliwych ustalonego wymiaru $n \times n$ (zob. podrozdział 6.6.2 w monografii M. Dryja, J. i M. Jankowscy, *Przegląd metod i algorytmów numerycznych. Część 2*, WNT Warszawa 1982).

Warto podkreślić, że algorytm eliminacji Gaussa jest w pewnym sensie optymalnym algorytmem rozwiązywania dowolnego nieosobliwego układu równań liniowych: wybór elementu głównego zapewnia jego numeryczną poprawność, a żaden inny numerycznie poprawny algorytm rozwiązywania nieosobliwego układu liniowego nie może być istotnie tańszy od eliminacji Gaussa (zob. W. Miller, *Computational Complexity and Numerical Stability*, SIAM Journal on Computing, vol. 4(2):97–107, 1975). Znane tańsze algorytmy rozwiązujące układy równań liniowych kosztem $\mathcal{O}\left(n^{2,79}\right)$ czy $\mathcal{O}\left(n^{2,53}\right)$ operacji arytmetycznych nie są numerycznie poprawne i okazują się bardziej efektywne dopiero dla układów o bardzo dużych wymiarach.

### 11.2.5. Zastosowanie metody eliminacji Gaussa

Metoda eliminacji Gaussa znajduje zastosowanie podczas:

- rozwiązywania układów równań liniowych,
- obliczania wyznaczników macierzy: z rozkładu (11.7) wynika, że

  $$\det A = \det P \det U = (-1)^s \, a_{11}^{(1)} a_{22}^{(2)} \cdot \ldots \cdot a_{nn}^{(n)},$$

  gdzie $s \in \{0, \ldots, n-1\}$ oznacza faktyczną liczbę permutacji wierszy wykonanych w całym procesie eliminacji,
- wyznaczania macierzy $A^{-1}$: oznaczając przez $x_i$ $i$-tą kolumnę macierz $A^{-1}$, tj.

  $$A^{-1} = [x_1, \ldots, x_n]$$

  zależność $AA^{-1} = I_n$ możemy zapisać w postaci $n$ układów równań liniowych

  $$Ax_i = e_i \qquad (i = 1, \ldots, n), \tag{11.9}$$

  gdzie $e_i$ oznacza $i$-ty wektor bazy kanonicznej przestrzeni $\mathbb{R}^n$. Każda kolumna macierzy $A^{-1}$ może być zatem wyznaczona algorytmem numerycznie poprawnym.[3] Zauważmy, że wyznaczając

[2] Porównywanie elementów oraz zmiana indeksów wierszy macierzy to nie są operacje arytmetyczne wykonywane w arytmetyce fl.

[3] Uzyskiwane oszacowania błędów zależą jednak od indeksu wyznaczanej kolumny – algorytm ten jest więc prawie numerycznie poprawnym algorytmem wyznaczania macierzy odwrotnej.

w podany sposób macierz $A^{-1}$ musimy rozwiązać $n$ układów równań (11.9) o tej samej macierzy $A$. Oznacza to, że algorytm eliminacji Gaussa wykonujemy tylko raz (!) i dysponując macierzami $P$, $L$ oraz $U$, które zwraca, rozwiązujemy odpowiednie układy o macierzach trójkątnych:

$$Ly_i = P^T e_i, \quad Ux_i = y_i \qquad (i = 1, \ldots, n). \tag{11.10}$$

Na koszt wyznaczenia macierzy $A^{-1}$ składa się zatem koszt uzyskania rozkładu $A = PLU$ równy $2n^3/3 + \mathcal{O}\left(n^2\right)$ oraz koszt rozwiązania $2n$ układów równań (11.10) o macierzach trójkątnych, czyli $2n^3$; całkowity koszt nie przekroczy zatem $8n^3/3 + \mathcal{O}\left(n^2\right)$ operacji arytmetycznych.

**Uwaga 11.1.** *W praktyce obliczeniowej konieczność wyznaczania macierzy $A^{-1}$ pojawia się stosunkowo rzadko. W celu wyznaczenia wektora $A^{-1}v$ należy rozwiązać układ równań $Ax = v$ metodą eliminacji Gaussa kosztem $2n^3/3 + \mathcal{O}\left(n^2\right)$ operacji arytmetycznych.*

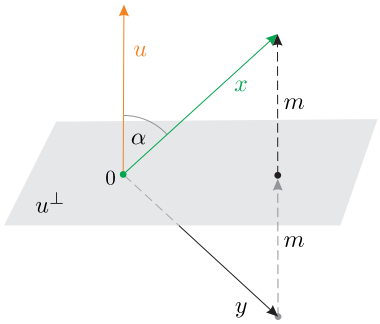
## 11.3. Metoda Householdera

### 11.3.1. Elementarne przekształcenie Householdera

Wektor $u \in \mathbb{R}^n$, $u \neq 0$, jest wektorem normalnym dla hiperpłaszczyzny

$$u^\perp = \left\{v \in \mathbb{R}^n : u^T v = 0\right\}.$$

Niech $y \in \mathbb{R}^n$ będzie tzw. *odbiciem zwierciadlanym* wektora $x \in \mathbb{R}^n$ względem hiperpłaszczyzny $u^\perp$ (zob. rysunek 12.1). Jest jasne, że $y = x - 2m$, gdzie $m$ jest odpowiednio przeskalowanym wektorem normalnym dla podprzestrzeni $u^\perp$.




Rysunek 11.1.

Ponieważ

$$m = \frac{u}{\|u\|_2} \|m\|_2$$

oraz

$$\cos \measuredangle (u, x) = \frac{\|m\|_2}{\|x\|_2}$$

zatem

$$m = \frac{u}{\|u\|_2} \|m\|_2 = \frac{u}{\|u\|_2} \|x\|_2 \cos \measuredangle (u, x) = \frac{u}{\|u\|_2^2} \|u\|_2 \|x\|_2 \cos \measuredangle (u, x) = \frac{uu^T x}{\|u\|_2^2}.$$

Stąd

$$y = x - 2m = x - 2\frac{uu^T x}{\|u\|_2^2} = \left(I_n - \frac{uu^T}{R}\right) x,$$

gdzie $R = \|u\|_2^2 / 2$. Macierz

$$P = I_n - \frac{uu^T}{R}, \qquad R = \frac{\|u\|_2^2}{2}$$

nazywamy macierzą Householdera.

**Stwierdzenie 11.2 (własności macierzy Householdera).** *Przy powyższych oznaczeniach*

**W1.** $P = P^T$ *oraz* $P^2 = I_n$*;*
**W2.** $\det P = -1$*.*

**Dowód**: Symetryczność macierzy $P$ jest oczywista. Dodatkowo

$$P^2 = \left(I_n - \frac{uu^T}{R}\right)\left(I_n - \frac{uu^T}{R}\right) = I_n - 2\frac{uu^T}{R} + \frac{uu^T uu^T}{R^2} =$$
$$= I_n - 2\frac{uu^T}{R} + 2\frac{u\left(u^T u\right)u^T}{Ru^T u} = I_n.$$

Aby wykazać własność W2 zauważmy, że

$$Pu = u - \frac{uu^T}{R}u = u - 2u = -u$$

co oznacza, że $\lambda_1 = -1$ jest wartością własną macierz $P$. Niech wektory $v_2, \ldots, v_n$ stanowią bazę przestrzeni $u^{\perp}$. Wówczas, dla $i = 2, \ldots, n$

$$Pv_i = v_i - \frac{uu^T v_i}{R} = v_i.$$

Wynika stąd, że wektory $v_2, \ldots, v_n$ są liniowo niezależnymi wektorami własnymi macierzy $P$ odpowiadającymi jej $(n-1)$-krotnej wartości własnej $\lambda_2 = \ldots = \lambda_n = 1$, zatem

$$\det P = \lambda_1 \cdot \ldots \cdot \lambda_n = -1.$$

■

Macierz Householdera zapamiętujemy jako wektor $(u, R)^T \in \mathbb{R}^{n+1}$ i nie wyznaczamy jej bez potrzeby.

**Uwaga 11.2.** *Dla dowolnego wektora* $x \in \mathbb{R}^n$ *wartość*

$$Px = \left(I_n - \frac{uu^T}{R}\right)x = x - \frac{uu^T x}{R}$$

*wyliczamy w następującej kolejności*

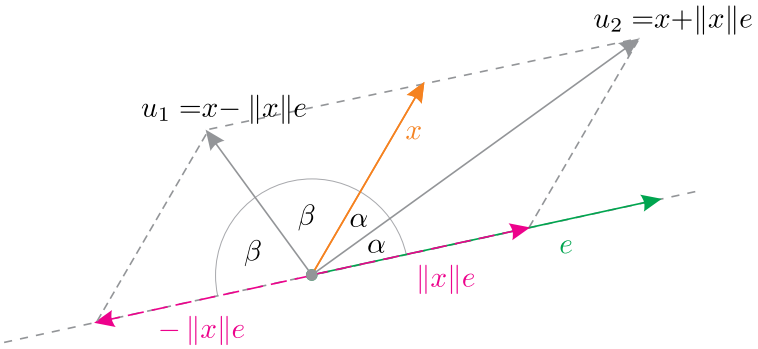
$$\begin{aligned}
u^T x &\rightsquigarrow 2n-1 \text{ operacji arytmetycznych} \\
\frac{u^T x}{R} &\rightsquigarrow 1 \text{ dzielenie} \\
u\frac{u^T x}{R} &\rightsquigarrow n \text{ operacji arytmetycznych} \\
x - u\frac{u^T x}{R} &\rightsquigarrow n \text{ operacji arytmetycznych}
\end{aligned}$$

*wykonując co najwyżej* $4n$ *operacje arytmetyczne. Samo wyznaczenie macierzy* $P$ *wymagałoby wykonania* $\mathcal{O}\left(n^2\right)$ *operacji arytmetycznych, obliczenie następnie wektora* $Px$ *to kolejne* $2n^2 + \mathcal{O}(n)$ *działania.*

### Przekształcanie wektora na kierunek innego wektora

Niech $x \in \mathbb{R}^n$ będzie dowolnym niezerowym wektorem oraz niech $e \in \mathbb{R}^n$ będzie wektorem jednostkowym, tj. $\|e\|_2 = 1$. W celu przekształcenia wektora $x$ na kierunek wektora $e$ możemy wykorzystać odpowiednio zdefiniowaną macierz Householdera. Łatwo zauważyć (zob. rysunek 12.2), że macierze Householdera $P_1$ oraz $P_2$ wyznaczone odpowiednio przez wektory $u_1$ oraz $u_2$, gdzie

$$u_1 = x - \|x\|_2 e \qquad \text{oraz} \qquad u_2 = x + \|x\|_2 e$$




Rysunek 11.2.

mają tę własność, że

$$P_1 x = \|x\|_2 e \qquad \text{oraz} \qquad P_2 x = -\|x\|_2 e.$$

W celu zapewnienia numerycznej stabilności metody, spośród dwóch wektorów $u_1$, $u_2$ definiujących macierze Householdera należy wybrać ten o dłuższej normie. Ponieważ

$$\begin{aligned}\|x \pm \|x\|_2 e\|_2^2 &= (x \pm \|x\|_2 e)^T (x \pm \|x\|_2 e) = x^T x \pm 2\|x\|_2 x^T e + \|x\|_2^2 e^T e = \\ &= 2\|x\|_2^2 \pm 2\|x\|_2 x^T e\end{aligned}$$

zatem wybieramy wektor $u_1$, gdy $x^T e < 0$ oraz wektor $u_2$, gdy $x^T e \geqslant 0$.

### 11.3.2. Algorytm Householdera

Niech $A \in \mathbb{R}^{m\times n}$ będzie macierzą kolumnami regularną (tj. $\operatorname{rank} A = n$). Wyznaczymy rozkład ortogonalno-trójkątny

$$A = QR,$$

w którym $Q \in \mathbb{R}^{m\times m}$ będzie macierzą ortogonalną, a $R = [r_{ij}] \in \mathbb{R}^{m\times n}$ będzie uogólnioną macierzą trójkątną górną, tzn. $r_{ij} = 0$ dla $i > j$ $(i = 1, \ldots, m;\ j = 1, \ldots, n)$.

Niech $A^{(1)} := A$ oraz niech $A_i^{(k)} \in \mathbb{R}^m$ $(i = 1, \ldots, n)$ oznacza $i$-tą kolumnę macierzy $A^{(k)} = [a_{ij}^{(k)}]$ powstałej w $k$-tym kroku algorytmu, tj.

$$A^{(k)} = \begin{pmatrix} A_1^{(k)} & \ldots & A_n^{(k)} \end{pmatrix}.$$

Wówczas, $A^{(2)} = P_1 A^{(1)} = \begin{pmatrix} P_1 A_1^{(1)} & \ldots & P_1 A_n^{(1)} \end{pmatrix}$, gdzie $P_1 \in \mathbb{R}^{m\times m}$ to macierz Householdera przekształcająca kolumnę $A_1^{(1)}$ na kierunek wektora $e_{1,m} = (1, 0, \ldots, 0)^T \in \mathbb{R}^m$. Stąd

$$A^{(2)} = \begin{pmatrix} a_{11}^{(2)} & a_{12}^{(2)} & \cdots & a_{1n}^{(2)} \\ 0 & a_{22}^{(2)} & \cdots & a_{2n}^{(2)} \\ \vdots & \vdots & & \vdots \\ 0 & a_{m2}^{(2)} & \cdots & a_{mn}^{(2)} \end{pmatrix} = \left(\begin{array}{c|c} a_{11}^{(2)} & \\ \hline \mathbf{0}_{(m-1)\times 1} & \widetilde{A}^{(2)} \end{array}\right).$$

W drugim kroku mamy $A^{(3)} = P_2 A^{(2)}$, gdzie

$$P_2 = \left(\begin{array}{c|ccc} 1 & 0 & \cdots & 0 \\ \hline 0 & & & \\ \vdots & & \widetilde{P}_2 & \\ 0 & & & \end{array}\right),$$

a $\widetilde{P}_2 \in \mathbb{R}^{(m-1)\times(m-1)}$ to macierz Householdera przekształacająca kolumnę $\widetilde{A}_1^{(2)}$ na kierunek wektora $e_{1,m-1} = (1, 0, \ldots, 0)^T \in \mathbb{R}^{m-1}$.

W $k$-tym kroku mamy

$$A^{(k)} = \left(\begin{array}{ccc|ccc} a_{11}^{(2)} & & & & & \\ & \ddots & & & & \\ & & a_{k-1,k-1}^{(k)} & & & \\ \hline & & & a_{kk}^{(k)} & \cdots & a_{kn}^{(k)} \\ & \mathbf{0}_{(m-k+1)\times(k-1)} & & \vdots & & \vdots \\ & & & a_{mk}^{(k)} & \cdots & a_{mn}^{(k)} \end{array}\right)$$

oraz $A^{(k+1)} = P_k A^{(k)}$, gdzie

$$P_k = \begin{pmatrix} I_{k-1} & \mathbf{0}_{(k-1)\times(n-k+1)} \\ \mathbf{0}_{(m-k+1)\times(k-1)} & \widetilde{P}_k \end{pmatrix},$$

a $\widetilde{P}_k \in \mathbb{R}^{(m-k+1)\times(m-k+1)}$ to macierz Householdera przekształcająca pierwszą kolumnę macierzy

$$\widetilde{A}_k = \begin{pmatrix} a_{kk}^{(k)} & \cdots & a_{kn}^{(k)} \\ \vdots & & \vdots \\ a_{mk}^{(k)} & \cdots & a_{mn}^{(k)} \end{pmatrix}$$

na kierunek wektora $e_{1,m-k+1} = (1, 0, \ldots, 0)^T \in \mathbb{R}^{m-k+1}$.

Po $n$ krokach otrzymujemy uogólnioną macierz trójkątną górną

$$R = A^{(n+1)} = P_n A^{(n)} = P_n P_{n-1} A^{(n-1)} = \ldots = P_n \cdot \ldots \cdot P_1 A. \tag{11.11}$$

W przypadku, gdy $m = n$ macierz $A^{(n)}$ jest macierzą trójkątną górną, czyli poszukiwany rozkład otrzymujemy po $n-1$ krokach. Przyjmując $Q^T = P_n \cdot \ldots \cdot P_1$ uzyskujemy poszukiwany rozkład ortogonalno-trójkątny macierzy $A$:

$$A = QR. \tag{11.12}$$

**Koszt i właściwości numeryczne:**[4] algorytm Householdera wymaga wykonania co najwyżej $\mathcal{O}\left((m - n/3)\, n^2\right)$ operacji arytmetycznych oraz wyliczenia co najwyżej $n$ pierwiastków. W podanej wersji (tj. z odpowiednim wyborem wektorów definiujących macierze Householdera) algorytm ten jest numerycznie poprawny.

**Przykład 11.3.** *Wykorzystując algorytm Householdera wyznaczymy rozkład ortogonalno-trójkątny macierzy*

$$A = \begin{pmatrix} 1 & -1 & 0 \\ 2 & 1 & -3 \\ 2 & 4 & 3 \end{pmatrix}.$$

*Niech* $a_1^{(1)} = (1, 2, 2)^T$, $a_2^{(1)} = (-1, 1, 4)^T$, $a_3^{(1)} = (0, -3, 3)^T$ *oraz* $e_1 = (1, 0, 0)^T$. *Ponieważ* $e_1^T a_1^{(1)} = 1 > 0$ *zatem przyjmujemy*

$$u_1 = a_1^{(1)} + \|a_1^{(1)}\| e_1 = (1, 2, 2)^T + 3(1, 0, 0)^T = (4, 2, 2)^T.$$

*Stąd* $R_1 = u_1^T u_1 / 2 = 12$ *oraz kolejno*

$$\begin{aligned} P_1 a_1^{(1)} &= a_1^{(1)} - u_1 (u_1^T a_1^{(1)})/R_1 = (1, 2, 2)^T - (4, 2, 2)^T = (-3, 0, 0)^T, \\ P_1 a_2^{(1)} &= a_2^{(1)} - u_1 (u_1^T a_2^{(1)})/R_1 = (-1, 1, 4)^T - (2, 1, 1)^T = (-3, 0, 3)^T, \\ P_1 a_3^{(1)} &= a_3^{(1)} - u_1 (u_1^T a_3^{(1)})/R_1 = (0, -3, 3)^T. \end{aligned}$$

[4] Osoby zainteresowane szczegółową analizą algorytmu Householdera w arytmetyce fl oraz wyznaczaniem jego kosztu odsyłam do podrozdziału 10.2 monografii A. Kiełbasiński, H. Schwetlick, *Numeryczna algebra liniowa. Wprowadzenie do obliczeń zautomatyzowanych*, WNT, Warszawa 1992.

*Mamy zatem*

$$P_1 A = (I_3 - u_1 u_1^T / R_1) \begin{pmatrix} 1 & -1 & 0 \\ 2 & 1 & -3 \\ 2 & 4 & 3 \end{pmatrix} = \begin{pmatrix} -3 & -3 & 0 \\ 0 & 0 & -3 \\ 0 & 3 & 3 \end{pmatrix}.$$

*Niech teraz* $a_1^{(2)} = (0,3)^T$, $a_2^{(2)} = (-3,3)^T$ *oraz* $e_2 = (1,0)^T$. *Ponieważ* $e_2^T a_1^{(2)} \geqslant 0$, *zatem przyjmujemy*

$$u_2 = a_1^{(2)} + \|a_1^{(2)}\| e_2 = (0,3)^T + (3,0)^T = (3,3)^T.$$

*Stąd* $R_2 = u_2^T u_2 / 2 = 9$ *oraz*

$$\widetilde{P}_2 a_1^{(2)} = a_1^{(2)} - u_2 (u_2^T a_1^{(2)}) / R_2 = (0,3)^T - (3,3)^T = (-3,0)^T,$$
$$\widetilde{P}_2 a_2^{(2)} = a_2^{(2)} - u_2 (u_2^T a_2^{(2)}) / R_2 = (-3,3)^T,$$

*gdzie*

$$\widetilde{P}_2 = I_2 - u_2 u_2^T / R_2 = \begin{pmatrix} 0 & -1 \\ -1 & 0 \end{pmatrix}.$$

*Ostatecznie*

$$P_2 P_1 A = \begin{pmatrix} -3 & -3 & 0 \\ 0 & -3 & -3 \\ 0 & 0 & 3 \end{pmatrix},$$

*gdzie*

$$P_1 = I_3 - u_1 u_1^T / R_1 = \frac{1}{3} \begin{pmatrix} -1 & -2 & -2 \\ -2 & 2 & -1 \\ -2 & -1 & 2 \end{pmatrix}, \; P_2 = \begin{pmatrix} 1 & 0 & 0 \\ 0 & 0 & -1 \\ 0 & -1 & 0 \end{pmatrix}.$$

*Poszukiwane macierze* $Q, R$ *tworzące rozkład* $A = QR$ *to*

$$Q = P_1 P_2 = \frac{1}{3} \begin{pmatrix} -1 & 2 & 2 \\ -2 & 1 & -2 \\ -2 & -2 & 1 \end{pmatrix} \text{ oraz } R = \begin{pmatrix} -3 & -3 & 0 \\ 0 & -3 & -3 \\ 0 & 0 & 3 \end{pmatrix}.$$

### 11.3.3. Zastosowanie algorytmu Householdera

Algorytm ortogonalizacyjny Householdera znajduje zastosowanie między innymi podczas:

- rozwiązywania układów równań liniowych,
- obliczania wyznaczników macierzy: dla macierzy kwadratowej z rozkładu (11.12) wynika, że

  $$\det A = \det Q \det R$$

  gdzie, na podstawie (11.11) oraz uwagi dotyczącej przypadku $m = n$,

  $$\det Q = \prod_{i=1}^{n-1} \det P_i = \pm 1$$

  oraz

  $$\det R = \det A^{(n)} = a_{11}^{(2)} a_{22}^{(3)} \cdot \ldots \cdot a_{n-1,n-1}^{(n)} a_{nn}^{(n)},$$

- rozwiązywania tzw. *liniowego zadania najmniejszych kwadratów.*

## 11.4. Metoda Grama-Schmidta

Niech $A \in \mathbb{R}^{m\times n}$ będzie macierzą kolumnami regularną (tj. $\operatorname{rank} A = n$) oraz niech $A_i \in \mathbb{R}^m$ $(i = 1, \ldots, n)$ oznacza $i$-tą kolumnę macierzy $A$. Wyznaczymy rozkład ortogonal- no-trójkątny

$$A = QR,$$

gdzie $Q \in \mathbb{R}^{m\times n}$ jest macierzą spełniającą warunek $Q^TQ = I_n$, a $R \in \mathbb{R}^{n\times n}$ to macierz trójkątna górna. Przedstawiona metoda oparta będzie na algorytmie ortogonalizacyjnym Grama-Schmidta.[5]

### 11.4.1. Algorytm ortogonalizacyjny

Kolumny $A_1, \ldots, A_n$ macierzy $A$ stanowią bazę pewnej $n$-wymiarowej podprzestrzeni liniowej przestrzeni $\mathbb{R}^m$. Wykorzystując algorytm Grama-Schmidta wyznaczymy bazę ortonormalną $Q_1, \ldots Q_n$ tej podprzestrzeni (ortonormalną w sensie naturalnego iloczynu skalarnego przestrzeni $\mathbb{R}^m$, czyli $\langle x, y\rangle = x^Ty$).

Mamy kolejno

$$Q_1 = \frac{A_1}{\|A_1\|} \tag{11.13}$$

oraz, dla $k \geqslant 2$,

$$Q_k = \frac{b_k}{\|b_k\|}, \tag{11.14}$$

gdzie

$$b_k = A_k - \sum_{i=1}^{k-1} \left(A_k^TQ_i\right) Q_i. \tag{11.15}$$

Zależności (11.13)–(11.15) możemy przepisać w postaci równoważnej jako

$$A_1 = \|A_1\|\, Q_1$$

oraz

$$A_k = \|b_k\|\, Q_k + \sum_{i=1}^{k-1} \left(A_k^TQ_i\right) Q_i.$$

W zapisie macierzowym mamy więc równość

$$(A_1, A_2, \ldots, A_n) = (Q_1, Q_2, \ldots, Q_n) \begin{pmatrix} \|A_1\| & A_2^TQ_1 & A_3^TQ_1 & \cdots & A_n^TQ_1 \\ 0 & \|b_2\| & A_3^TQ_2 & \cdots & A_n^TQ_2 \\ \vdots & 0 & \|b_3\| & & \vdots \\ & \vdots & 0 & & \vdots \\ & & \vdots & \ddots & A_n^TQ_{n-1} \\ & & & & \|b_n\| \end{pmatrix}.$$

[5] **Algorytm ortogonalizacyjny Grama-Schmidta:** jeżeli wektory $v_1, \ldots, v_n$ stanowią bazę przestrzeni $X$ wyposażonej w iloczyn skalarny $\langle\cdot,\cdot\rangle$, to wektory $u_1, \ldots, u_n$ postaci

$$u_1 = \frac{v_1}{\|v_1\|}; \quad u_k = \frac{b_k}{\|b_k\|} \quad \text{dla} \quad b_k = v_k - \sum_{i=1}^{k-1} \langle v_k, u_i\rangle\, u_i,\ k \geqslant 2$$

gdzie $\|v\| = \sqrt{\langle v, v\rangle}$ dla $v \in X$, są bazą ortonormalną przestrzeni $X$.

Oznaczając

$$Q = (Q_1, Q_2, \ldots, Q_n) \quad \text{oraz} \quad R = \begin{pmatrix} \|A_1\| & A_2^T Q_1 & A_3^T Q_1 & \cdots & A_n^T Q_1 \\ 0 & \|b_2\| & A_3^T Q_2 & \cdots & A_n^T Q_2 \\ \vdots & 0 & \|b_3\| & & \vdots \\ & \vdots & 0 & & \vdots \\ & & \vdots & \ddots & A_n^T Q_{n-1} \\ & & & & \|b_n\| \end{pmatrix}$$

otrzymujemy poszukiwany rozkład ortogonalno-trójkątny macierzy $A$.

### 11.4.2. Koszt i właściwości numeryczne

Koszt wyliczenia wektora $b_k$ zgodnie z formułą (11.15) to

$$\underbrace{(2m-1+m)}_{k_1}(k-1) + \underbrace{(k-2)\,m}_{k_2} + m = (4m-1)(k-1)$$

operacji arytmetycznych, gdzie $k_1$ to koszt wyliczenia każdego z $k-1$ składników $\left(A_k^T Q_i\right) Q_i$ a $k_2$ to koszt ich sumowania; koszt wyliczenia wektora $Q_k$ zgodnie z formułą (11.14) to dodatkowe $3m-1$ operacje arytmetyczne oraz jeden pierwiastek. Algorytm wymaga zatem wykonania nie więcej niż

$$\sum_{k=1}^{n} ((4m-1)(k-1) + 3m - 1) = \sum_{k=1}^{n} ((4m-1)k - m) = 2mn^2 + \mathcal{O}(mn)$$

operacji arytmetycznych oraz obliczenia $n$ pierwiastków.

Przekonamy się teraz, że przedstawiony powyżej algorytm ortogonalizacyjny jest katastrofalnie niestabilny.

**Przykład 11.4.** *Rozważmy macierz $A \in \mathbb{R}^{4\times 3}$ postaci*

$$A = \begin{pmatrix} 1 & 1 & 1 \\ \varepsilon & 0 & 0 \\ 0 & \varepsilon & 0 \\ 0 & 0 & \varepsilon \end{pmatrix},$$

*gdzie $\varepsilon > 0$. Przypuśćmy, że $\mathrm{fl}\left(1+\varepsilon^2\right) = 1$ a wszystkie pozostałe operacje wykonujemy dokładnie. Stosując do kolumn macierzy $A$ algorytm ortogonalizacyjny opisany formułami* (11.13)–(11.15) *otrzymujemy*

$$\mathrm{fl}\left(\|A_1\|\right) = \mathrm{fl}\left(\sqrt{1+\varepsilon^2}\right) = 1$$

*czyli $Q_1 = A_1$. Z kolei*

$$b_2 = A_2 - \left(A_2^T Q_1\right) Q_1 = A_2 - A_1 = (0, -\varepsilon, \varepsilon, 0)^T$$

*czyli $\|b_2\| = \varepsilon 2^{1/2}$ oraz*

$$Q_2 = \left(0, -2^{-1/2}, 2^{-1/2}, 0\right)^T.$$

*Podobnie,*

$$b_3 = A_3 - \left(A_3^T Q_1\right) Q_1 - \left(A_3^T Q_2\right) Q_2 = A_3 - A_1 = (0, -\varepsilon, 0, \varepsilon)^T,$$

*czyli $\|b_3\| = \varepsilon 2^{1/2}$ oraz*

$$Q_3 = \left(0, -2^{-1/2}, 0, 2^{-1/2}\right)^T.$$

*Stąd*

$$Q = \begin{pmatrix} 1 & 0 & 0 \\ \varepsilon & -2^{-1/2} & -2^{-1/2} \\ 0 & 2^{-1/2} & 0 \\ 0 & 0 & 2^{-1/2} \end{pmatrix} \quad \textit{oraz} \quad R = \begin{pmatrix} 1 & 1 & 1 \\ 0 & \varepsilon 2^{1/2} & 0 \\ 0 & 0 & \varepsilon 2^{1/2} \end{pmatrix}.$$

*Łatwo sprawdzić, że w arytmetyce dokładnej zachodzi*

$$QR = \begin{pmatrix} 1 & 0 & 0 \\ \varepsilon & -2^{-1/2} & -2^{-1/2} \\ 0 & 2^{-1/2} & 0 \\ 0 & 0 & 2^{-1/2} \end{pmatrix} \begin{pmatrix} 1 & 1 & 1 \\ 0 & \varepsilon 2^{1/2} & 0 \\ 0 & 0 & \varepsilon 2^{1/2} \end{pmatrix} = \begin{pmatrix} 1 & 1 & 1 \\ \varepsilon & 0 & 0 \\ 0 & \varepsilon & 0 \\ 0 & 0 & \varepsilon \end{pmatrix} = A,$$

*lecz macierz $Q$ nie jest ortogonalna: kolumny macierzy ortogonalnej tworzą układ ortonormalny, a w naszym przypadku*

$$\measuredangle (Q_2, Q_3) = \arccos \frac{Q_2^T Q_3}{\|Q_2\| \, \|Q_3\|} = \arccos \frac{1}{2} = \frac{\pi}{3}.$$

W celu poprawienia właściwości numerycznych przedstawionego algorytmu wystarczy wyznaczane zgodnie z formułą (11.15) wektory $b_2, \ldots, b_n$ ponownie poddać ortogonalizacji otrzymując wektory $b'_2, \ldots, b'_n$, gdzie dla $k = 2, \ldots, n$:

$$b'_k = b_k - \sum_{i=1}^{k-1} (b_k^T Q_i) Q_i.$$

Proces ten nazywy *reortogonalizacją*. Koszt takiej modyfikacji jest niemal taki sam jak koszt wyjściowego algorytmu; w szczególności warto podkreślić, że reortogonalizacja nie wymaga obliczania dodatkowych pierwiastków arytmetycznych (w całym procesie wyznaczamy ich co najwyżej $n$), ale jej zastosowanie poprawia właściwości numeryczne algorytmu. Można wykazać, że algorytm Grama-Schmidta z reortogonalizacją jest numerycznie poprawny w klasie macierzy ustalonego wymiaru o liniowo niezależnych kolumnach.

Kończąc rozważania na temat konstrukcji rozkładów ortogonalno-trójkątnych macierzy zauważmy, że w przypadku gdy $m \neq n$, przedstawione w tym rozdziale algorytmy Householdera oraz Grama-Schmidta zastosowane do macierzy $A \in \mathbb{R}^{m \times n}$ o kolumnach liniowo niezależnych zwracają dwa jakościowo różne jej rozkłady $QR$. Pierwszy z algorytmów zwraca macierz ortogonalną[6] $Q$ wymiaru $m \times m$ oraz uogólnioną macierz trójkątną górną $R$ wymiaru $m \times n$. Z kolei algorytm Grama-Schmidta zwraca macierz $Q$ wymiaru $m \times n$ o kolumnach wzajemnie ortonormalnych, tj. taką że $Q^T Q = I_n$, oraz macierz trójkątną górną $R$ wymiaru $n \times n$. Można wykazać, że dla każdej nieosobliwej macierzy kwadratowej istnieje dokładnie jeden rozkład ortogonalno-trójkątny $QR$, w którym macierz $Q$ jest macierzą ortogonalną a $R$ to macierz trójkątna górna z dodatnimi elementami na przekątnej.

[6] Przypomnijmy, że macierz $Q$ jest macierzą ortogonalną, jeżeli jest macierzą kwadratową wymiaru $m \times m$ spełniającą warunek $Q^T Q = Q Q^T = I_m$.

## Rozdział 12

# Metody iteracyjne rozwiązywania układów równań liniowych

W rozdziale tym skupimy się na metodach przybliżonych rozwiązywania układu równań liniowych

$$Ax = b \tag{12.1}$$

o macierzy nieosobliwej $A \in \mathbb{R}^{n\times n}$. Metody przybliżone, stanowiące alternatywę dla metod dokładnych wykorzystujących rozkłady trójkątno–trójkątne czy ortogonalno–trójkątne przedstawione w poprzednim rozdziale, znajdują zastosowanie w sytuacjach, w których metody dokładne zawodzą. Dla układów o bardzo dużej liczbie niewiadomych koszt metod skończonych może okazać się zbyt duży, mogą one wymagać olbrzymich zasobów pamięci potrzebnej to zapisania czynników rozkładu, a same metody mogą być niewygodne w użyciu.

Macierze o wielkich wymiarach występujące w zagadnieniach praktycznych najczęściej są macierzami *rzadkimi*, tzn. większość ich elementów ma wartość zero. Macierze rzadkie pojawiają się m.in. podczas wyznaczania funkcji sklejanych (zob. układ równań (5.12) na str. 34) czy w trakcie numerycznego rozwiązywania równań różniczkowych cząstkowych metodą różnic skończonych lub elementu skończonego.[1] Przykładem problemu obliczeniowego, w którym występują macierze o ekstremalnie wielkich rozmiarach jest zadanie polegające na wyznaczeniu wektora PageRank wartościującego strony internetowe w wyszukiwarce Google – wymaga ono pracy z macierzami o wymiarach równych liczbie indeksowanych stron www, czyli około $10^{10}$. Czas potrzebny na uzyskanie rozkładu $LU$ takich macierzy przy pomocy algorytmu Gaussa na superkomputerze wykonującym $10^{15}$ operacji zmiennopozycyjnych na sekundę to ponad $10^7\ldots$ lat!

Zaletą metod iteracyjnych, których przykłady zostaną omówione w dalszej części tego rozdziału, jest ich prostota powodująca, że są one bardzo przyjemne w implementacji. Często stanowią one również bardzo ważny składnik innych dużo bardziej efektywnych metod rozwiązywania *trudnych* układów równań.

### 12.1. Stacjonarne metody iteracyjne

Macierz $A \in \mathbb{R}^{n\times n}$ zapiszmy w postaci

$$A = N + R,$$

[1] Na stronie www.cise.ufl.edu/research/sparse/matrices/Boeing/ zamieszczono przykłady macierzy, które uzyskano podczas prac konstrukcyjnych w zakładach Boeinga oraz NASA. Wiele spośród nich to macierze rzadkie o olbrzymich wymiarach.

gdzie $N \in \mathbb{R}^{n\times n}$ jest macierzą nieosobliwą (łatwą do odwrócenia), a $R \in \mathbb{R}^{n\times n}$ to reszta. Układ równań (12.1) możemy wówczas zapisać w postaci równoważnej

$$Nx = -Rx + b,$$

lub jako

$$x = -N^{-1}Rx + N^{-1}b. \tag{12.2}$$

Wprowadzając oznaczenia

$$B = -N^{-1}R, \qquad c = N^{-1}b$$

zależność (12.2) przyjmuje postać

$$x = Bx + c. \tag{12.3}$$

Poszukiwane rozwiązanie $x$ jest zatem punktem stałym odwzorowania

$$\Phi : \mathbb{R}^n \ni x \to Bx + c \in \mathbb{R}^n, \tag{12.4}$$

który będzie mógł być wyznaczony przy pomocy, wynikającej z twierdzenia Banacha o punkcie stałym, iteracji

$$x_{k+1} = Bx_k + c. \tag{12.5}$$

Metody takiej postaci nazywamy *stacjonarnymi metodami iteracyjnymi*, macierz $B$ nazywamy *macierzą iterującą*.

**Uwaga 12.1.** *Zauważmy, że z zależności* (12.2) *wynika wygodna do implementacji postać metody iteracyjnej*

$$x_{k+1} = -N^{-1}\left(Rx_k - b\right) = -N^{-1}\left(\left(A - N\right)x_k - b\right) = x_k - N^{-1}r_k,$$

*w której wektor residualny $r_k = Ax_k - b$ może być wykorzystywany do określenia tzw. warunku stopu, czyli momentu zatrzymania iteracji.*

## 12.2. Zbieżność stacjonarnych metod iteracyjnych

Łącząc zależności (12.3) oraz (12.5) otrzymujemy równość

$$x_{k+1} - x = Bx_k + c - Bx - c = B\left(x_k - x\right),$$

z której wynika, że

$$x_{k+1} - x = B^{k+1}\left(x_0 - x\right).$$

Stąd, dla dowolnej normy wektorowej oraz zgodnej z nią normy macierzowej,

$$\|x_{k+1} - x\| \leqslant \|B\|^{k+1}\|x_0 - x\|, \qquad \text{dla } k \in \mathbb{N}_0. \tag{12.6}$$

**Stwierdzenie 12.1.** *Jeżeli $\|B\| < 1$, gdzie $\|\cdot\|$ oznacza dowolną normę macierzową zgodną z pewną normą wektorową, to metoda iteracyjna* (12.5) *jest zbieżna, niezależnie od wyboru przybliżenia początkowego $x_0$, do rozwiązania równania $Ax = b$.*

**Dowód**: Dla dowolnego przybliżenia początkowego $x_0 \in \mathbb{R}^n$, z zależności (12.6) otrzymujemy[2]

$$\|x_{k+1} - x\| \leqslant \|B\|^{k+1}\|x_0 - x\| \overset{k\to+\infty}{\longrightarrow} 0.$$

To kończy dowód. ∎

Dla wielu norm macierzowych sprawdzenie podanego w stwierdzeniu 12.1. warunku wystarczającego zbieżności metody iteracyjnej jest łatwe; tak jest na przykład w przypadku norm $\|\cdot\|_1$, $\|\cdot\|_\infty$ oraz normy Frobeniusa $\|\cdot\|_F$ (zob. dodatek A). Na ogół dużo trudniej stwierdzić, czy spełniony jest warunek konieczny i wystarczający.

[2] Zauważmy, że do tego samego wniosku można dojść wykorzystując twierdzenie Banacha o punkcie stałym. W tym celu wystarczy uzasadnić, że odwzorowanie $\Phi$ postaci (12.4) jest kontrakcją ze stałą $\|B\| < 1$ przekształcającą kulę $\{z \in \mathbb{R}^n : \|z - x\| \leqslant \|x_0 - x\|\}$ w siebie.

**Twierdzenie 12.2.** *Metoda iteracyjna* (12.5) *jest zbieżna, niezależnie od wyboru przybliżenia początkowego $x_0$, do rozwiązania równania $Ax = b$ wtedy i tylko wtedy, gdy*

$$\rho(B) = \max\{|\lambda| : \lambda \in \sigma(B)\} < 1.$$

Dowód twierdzenia opiera się na dwóch ważnych faktach.

**Lemat 12.3.** *Dla dowolnej macierzy $B \in \mathbb{R}^{n\times n}$ istnieją macierze nieosobliwa $P$ oraz trójkątna górna $T$ takie, że*

$$B = PTP^{-1};$$

*dodatkowo, macierz $T$ może być tak wybrana, aby jej elementy pozadiagonalne były dowolnie małe co do modułu.*

**Dowód**: Dla macierzy $B$ istnieje macierz nieosobliwa $Q$, dla której

$$Q^{-1}BQ = J_B,$$

gdzie $J_B = [r_{ij}]$ jest macierzą Jordana macierzy $B$. Stąd, dla dowolnego $\varepsilon > 0$ oraz macierzy diagonalnej $D_\varepsilon$ postaci

$$D_\varepsilon = \operatorname{diag}\left(\varepsilon, \varepsilon^2, \ldots, \varepsilon^n\right)$$

otrzymujemy

$$D_\varepsilon^{-1} J_B D_\varepsilon = \left[r_{ij}\varepsilon^{j-i}\right].$$

Poszukiwanymi macierzami są zatem $P = QD_\varepsilon$ oraz $T = D_\varepsilon^{-1} J_B D_\varepsilon$. ∎

**Lemat 12.4.** *Dla dowolnej macierzy $B \in \mathbb{R}^{n\times n}$ zachodzi*

$$\rho(B) = \inf_{\|\cdot\|} \|B\|,$$

*gdzie infimum jest brane po wszystkich normach macierzowych $\|\cdot\|$ zgodnych z jakąć normą wektorową.*

**Dowód**: Dla dowolnej wartości własnej $\lambda \in \sigma(B)$ istnieje wektor $v \neq 0$ taki, że $Bv = \lambda v$. Stąd

$$|\lambda|\,\|v\| = \|\lambda v\| = \|Bv\| \leqslant \|B\|\,\|v\|,$$

czyli

$$|\lambda| \leqslant \|B\|.$$

Ponieważ lewa strona nie zależy od normy macierzy, zatem biorąc infimum po wszystkich normach macierzowych, wobec dowolności wyboru wartości własnej $\lambda$, otrzymujemy

$$\rho(B) \leqslant \inf_{\|\cdot\|} \|B\|.$$

Aby wykazać nierówność w drugą stronę ustalmy dowolne $\varepsilon > 0$. Na podstawie lematu 12.3 wnioskujemy, że istnieją macierze nieosobliwa $P$ oraz trójkątna górna $T$, dla których

$$P^{-1}BP = T$$

oraz elementy pozadiagonalne macierzy $T$ mogą być dowolnie małe. Macierz $T$ możemy zatem zapisać w postaci

$$T = \operatorname{diag}(\lambda_1, \ldots, \lambda_n) + R,$$

gdzie $\lambda_1, \ldots, \lambda_n$ to wartości własne macierzy $B$, a $R = [r_{ij}]$ jest macierzą trójkątną górną, taką że

$$r_{ii} = 0 \ \ (i = 1, \ldots, n) \quad \text{oraz} \quad \|R\|_\infty \leqslant \varepsilon.$$

Stąd

$$\left\|P^{-1}BP\right\|_\infty = \left\|\operatorname{diag}(\lambda_1,\ldots,\lambda_n)+R\right\|_\infty \leqslant \left\|\operatorname{diag}(\lambda_1,\ldots,\lambda_n)\right\|_\infty + \|R\|_\infty \leqslant$$
$$\leqslant \rho(B) + \|R\|_\infty \leqslant \rho(B)+\varepsilon$$

i ostatecznie

$$\inf_{\|\cdot\|} \|B\| \leqslant \left\|P^{-1}BP\right\|_\infty \leqslant \rho(B)+\varepsilon.$$

Ponieważ ani lewa strona, ani $\rho(B)$ nie zależy od $\varepsilon$, zatem

$$\inf_{\|\cdot\|} \|B\| \leqslant \rho(B).$$

Nierówność ta kończy dowód. ∎

*Dowód twierdzenia 12.2*: Przypuśćmy, że $\rho(B)<1$. Stąd, na podstawie lematu 12.4, wnioskujemy, że istnieje norma macierzowa $\|\cdot\|$ zgodna z pewną normą wektorową, dla której

$$\|B\|<1.$$

Stwierdzenie 12.1 kończy dowód.

Implikację w drugą stronę wykażemy przy dodatkowym założeniu, że największa co do modułu wartość własna macierzy $B$ jest rzeczywista.

Przypuśćmy, że $\rho(B)\geqslant 1$. Oznacza to, że istnieje wartość własna $\lambda$ oraz wektor $v\neq 0$, takie że

$$|\lambda|\geqslant 1 \quad \text{oraz} \quad Bv=\lambda v.$$

Dla przybliżenia początkowego $x_0=v+x$ mamy wówczas

$$x_{k+1}-x = B^{k+1}(x_0-x) = B^{k+1}v = \lambda^{k+1}v,$$

skąd wynika, że dla dowolnej normy wektorowej oraz dla dowolnego $k\geqslant 0$

$$\|x_{k+1}-x\| = \|\lambda^{k+1}v\| = |\lambda|^{k+1}\|v\| \geqslant \|v\| > 0.$$

Podsumowując: istnieje przybliżenie początkowe $x_0$, dla którego metoda iteracyjna (12.5) nie jest zbieżna do rozwiązania równania $Ax=b$. To kończy dowód twierdzenia. ∎

Okazuje się, że wielkość promienia spektralnego macierzy iterującej $B$ decyduje nie tylko o samej zbieżności, lecz również o szybkości tej zbieżności. Aby to zaobserwować przypuśćmy, że macierz $B$ jest diagonalizowalna, tj.

$$B = P\operatorname{diag}(\lambda_1,\ldots,\lambda_n)P^{-1},$$

dla pewnej macierzy nieosobliwej $P$. Wówczas

$$\|x_k-x\| \leqslant \|B^k\|\|x_0-x\| = \|P\operatorname{diag}(\lambda_1^k,\ldots,\lambda_n^k)P^{-1}\|\,\|x_0-x\| \leqslant$$
$$\leqslant \|P\|\|P^{-1}\|\|\operatorname{diag}(\lambda_1^k,\ldots,\lambda_n^k)\|\,\|x_0-x\| = \operatorname{cond}(P)\,\rho^k(B)\,\|x_0-x\|,$$

gdzie $\|\cdot\|$ to norma macierzowa zgodna z jakąkolwiek normą wektorową spełniająca warunek

$$\|\operatorname{diag}(x_1,\ldots,x_n)\| = \max_{k=1}^{n}|x_k|;$$

może to być zatem norma $\|\cdot\|_1, \|\cdot\|_2, \|\cdot\|_\infty$, ale nie $\|\cdot\|_F$.

W przypadku dowolnej macierzy $B$ (również niediagonalizowalnej) można pokazać, że dla dowolnej wartości $\varepsilon>0$ i dla dostatecznie dużych wartości $k$

$$\|x_k-x\|_2 \leqslant (\rho(B)+\varepsilon)^k\|x_0-x\|_2.$$

## 12.3. Przykłady stacjonarnych metod iteracyjnych

Macierz $A = [a_{ij}]$ zapiszmy w postaci sumy trzech macierzy: ściśle trójkątnej dolnej $L = [l_{ij}]$, diagonalnej $D = \operatorname{diag}(a_{11}, \ldots, a_{nn})$ oraz ściśle trójkątnej górnej $U = [u_{ij}]$

$$A = L + D + U, \tag{12.7}$$

oczywiście

$$l_{ij} = \begin{cases} a_{ij} & i > j \\ 0 & i \leqslant j \end{cases} \quad \text{oraz} \quad u_{ij} = \begin{cases} a_{ij} & i < j \\ 0 & i \geqslant j \end{cases}.$$

### 12.3.1. Metoda Jacobiego

Wykorzystując rozkład (12.7) układ równań $Ax = b$ możemy zapisać w postaci

$$Dx = -(L + U)x + b,$$

lub równoważnie

$$x = -D^{-1}(L + U)x + D^{-1}b.$$

Stąd wynika postać metody iteracyjnej Jacobiego

$$\begin{aligned} x_{k+1} &= -D^{-1}(L + U)x_k + D^{-1}b = \\ &= D^{-1}(b - (L + U)x_k) \end{aligned} \tag{12.8}$$

elementy macierzy iterującej $B_J = -D^{-1}(L + U)$ są postaci

$$b_{ij} = \begin{cases} -\dfrac{a_{ij}}{a_{ii}} & i \neq j \\ 0 & i = j \end{cases}. \tag{12.9}$$

Przyjmując $x_k = (x_{k,1}, \ldots, x_{k,n})^T$ oraz rozpisując równość (12.8) na współrzędne otrzymujemy

$$x_{k+1,i} = \frac{1}{a_{ii}} \left( b_i - \sum_{j \neq i} a_{ij} x_{k,j} \right), \qquad \text{dla } i = 1, \ldots, n. \tag{12.10}$$

Przepis (12.10) potwierdza prostotę implementacji metody Jacobiego oraz jej istotnie mniejszy koszt, w porównaniu do kosztu metod skończonych przedstawionych w poprzednim rozdziale: zgodnie z (12.10), koszt wykonania $k$–tego kroku iteracji to w najgorszym przypadku $2n^2 + \mathcal{O}(n)$ operacji arytmetycznych, ale w przypadku macierzy rzadkich koszt ten będzie istotnie mniejszy.[3] Musimy również pamiętać, że na całkowity koszt metody składa się wielokrotne wywołanie iteracji (12.10), ale liczba tych wywołań będzie najczęściej istotnie mniejsza niż $n$ (przypomnijmy, że metody iteracyjne stosujemy dla macierzy o bardzo duży wymiarach). W efekcie, całkowity koszt jest co najmniej o rząd wielkości mniejszy w porównaniu do kosztu metod opartych na rozkładach $LU$ lub $QR$ macierzy $A$. Z (12.10) wynika również, że każda współrzędna wektora $x_k$ może być wyznaczana niezależenie od pozostałych; obliczenia mogą więc być realizowane równolegle, np. na procesorach graficznych GPU.

Z postaci (12.9) macierzy $B_J$ wynika równość

$$\|B_J\|_\infty = \left\|D^{-1}(L + U)\right\|_\infty = \max_{i=1}^{n} \sum_{j \neq i} \frac{|a_{ij}|}{|a_{ii}|}, \tag{12.11}$$

a z niej prosty warunek wystarczający zbieżności metody Jacobiego.

[3] Zakładając, że macierz $A$ posiada dokładnie $c(n)$ niezerowych elementów, liczba działań arytmetycznych potrzebnych do wykonania w $k$-tym kroku iteracji (12.10) nie przekroczy $2c(n) + \mathcal{O}(n)$.

**Stwierdzenie 12.5 (zbieżność metody Jacobiego).** *Jeżeli macierz $A$ ma silnie dominującą przekątną, tj.*

$$|a_{ii}| > \sum_{j \neq i} |a_{ij}|, \qquad dla\ i = 1, \ldots, n,$$

*to metoda Jacobiego jest zbieżna, niezależnie od wyboru przybliżenia początkowego $x_0$, do rozwiązania równania $Ax = b$.*

**Dowód**: Na podstawie równości (12.11) oraz założeń wnioskujemy, że

$$\|B_J\|_\infty < 1;$$

teza wynika ze stwierdzenia 12.1. ∎

### 12.3.2. Metoda Gaussa-Seidela

Wykorzystując rozkład (12.7) układ równań $Ax = b$ możemy zapisać jako

$$(L + D)\, x = -Ux + b,$$

lub równoważnie

$$x = -(L + D)^{-1} Ux + (L + D)^{-1} b.$$

Stąd wynika zależność rekurencyjna określająca metodę iteracyjną Gaussa-Seidela:

$$x_{k+1} = -(L + D)^{-1} Ux_k + (L + D)^{-1} b,$$

którą równoważnie możemy zapisać w postaci niewymagającej wyznaczania macierzy odwrotnej

$$(L + D)\, x_{k+1} = b - Ux_k. \tag{12.12}$$

Po rozpisaniu równości (12.12) na współrzędne otrzymujemy łatwą do zaimplementowania postać metody Gaussa-Seidela:

$$x_{k+1,i} = \frac{1}{a_{ii}} \left( b_i - \sum_{j<i} a_{ij} x_{k+1,j} - \sum_{j>i} a_{ij} x_{k,j} \right), \qquad \text{dla } i = 1, \ldots, n.$$

Aby uzyskać warunek gwarantujący zbieżność metody Gaussa-Seidela przypuśćmy, że $\lambda$ jest dowolną wartością własną macierzy iterującej $B_{GS} = -(L + D)^{-1}U$, a $v \neq 0$ odpowiadającym jej unormowanym wektorem własnym, tj.

$$-(L + D)^{-1} Uv = \lambda v \qquad \text{oraz} \qquad \|v\|_\infty = 1.$$

Stąd

$$-Uv = \lambda (L + D)\, v,$$

co, po rozpisaniu na współrzędne, prowadzi do równości

$$-\sum_{j=i+1}^{n} a_{ij} v_j = \lambda \sum_{j=1}^{i} a_{ij} v_j = \lambda \sum_{j=1}^{i-1} a_{ij} v_j + \lambda a_{ii} v_i, \tag{12.13}$$

dla $i = 1, \ldots, n$. W szczególności, dla każdego indeksu $i$ spełniającego warunek

$$1 = \|v\|_\infty = |v_i|$$

z równości (12.13) otrzymujemy

$$|\lambda|\, |a_{ii}| \leqslant |\lambda| \sum_{j=1}^{i-1} |a_{ij}| + \sum_{j=i+1}^{n} |a_{ij}|,$$

lub równoważnie

$$|\lambda| \left( |a_{ii}| - \sum_{j=1}^{i-1} |a_{ij}| \right) \leqslant \sum_{j=i+1}^{n} |a_{ij}| \,. \tag{12.14}$$

Wynika stąd prosty warunek zbieżności metody Gaussa-Seidela.

**Stwierdzenie 12.6 (zbieżność metody Gaussa-Seidela).** *Jeżeli macierz $A$ ma silnie dominującą przekątną, to metoda Gaussa-Seidela jest zbieżna do rozwiązania równania $Ax = b$, niezależnie od wyboru przybliżenia początkowego $x_0$.*

**Dowód**: Z założenia dotyczącego macierzy $A$ otrzymujemy nierówność

$$\frac{\sum\limits_{j=i+1}^{n} |a_{ij}|}{|a_{ii}| - \sum\limits_{j=1}^{i-1} |a_{ij}|} < 1,$$

która w połączeniu z (12.14), wobec dowolności wyboru wartości własnej $\lambda$, prowadzi do warunku

$$\rho\left(B_{GS}\right) < 1.$$

Na podstawie twierdzenia 12.2 dowód jest zakończony. ■

## Rozdział 13

# Metody iteracyjne rozwiązywania równań nieliniowych

Niech $f : [a, b] \to \mathbb{R}$ będzie zadaną funkcją oraz niech

$$f(\alpha) = 0, \tag{13.1}$$

dla pewnego $\alpha \in (a, b)$. W tym rozdziale naszym zadaniem jest wyznaczenie $\alpha$. W przypadku ogólnym efektywne metody skończone (tj. takie, które po wykonaniu skończonej liczby operacji zwracają wynik dokładny) umożliwiające rozwiązanie równania (13.1) nie istnieją; jesteśmy zatem skazani na metody iteracyjne (przybliżone).

## 13.1. Metody iteracyjne

Każda metoda iteracyjna rozwiązująca równanie (13.1) zwraca pewien ciąg przybliżeń $x_0, x_1, \ldots$ poszukiwanej wartości $\alpha$. Ogólna postać takiej metody to

$$\begin{cases} x_{k+1} = \varphi_k(x_k, \ldots, x_{k-n}; f), \quad k \geqslant n \\ x_0, \ldots, x_n - \text{wartości początkowe} \end{cases}. \tag{13.2}$$

Metody iteracyjne konstruowane są w ten sposób, aby ciąg $\{x_k\}$ jaki zwracają był zbieżny do poszukiwanej wartości $\alpha$.

### 13.1.1. Rodzaje metod iteracyjnych

Metodę (13.2) nazywamy

- metodą *bez pamięci*, jeżeli $n = 0$, tj. $x_{k+1} = \varphi_k(x_k; f)$,
- metodą *z pamięcią*, jeżeli $n \geqslant 1$,
- metodą *stacjonarną*, jeżeli $\varphi_k = \varphi$ dla $k \in \mathbb{N}$,
- metodą *niestacjonarną*, jeżeli postać funkcji $\varphi_k$ zależy od $k$,
- metodą *zbieżną lokalnie* w klasie funkcji $\mathcal{F}$, jeżeli

$$\forall f \in \mathcal{F} \quad \exists \delta > 0 \quad \forall x_0, \ldots, x_n : \quad |x_i - \alpha| < \delta \Rightarrow \lim_{k \to \infty} x_k = \alpha,$$

- metodą *zbieżną globalnie* w klasie funkcji $\mathcal{F}$, jeżeli

$$\forall f \in \mathcal{F} \quad \forall x_0, \ldots, x_n : \quad \lim_{k \to \infty} x_k = \alpha.$$

Liczbę $e_k = x_k - \alpha$ nazywamy błędem przybliżenia wartości $\alpha$. Występująca w powyższych definicjach klasa funkcji $\mathcal{F}$ będzie zwykle klasą funkcji o odpowiedniej regularności posiadających miejsce zerowe $\alpha$ o ustalonej krotności $k \in \mathbb{N}$. [1]

Jak wynika z powyższych definicji, metody zbieżne lokalnie zwracają ciąg przybliżeń zbieżny do poszukiwanej wartości $\alpha$, o ile przybliżenia początkowe znajdują się dostatecznie blisko wartości $\alpha$. Z kolei metody zbieżne globalnie zwracają ciąg zbieżny do wartości $\alpha$, niezależnie od wyboru przybliżeń początkowych. Z oczywistych powodów może wydawać się, że metody zbieżne globalnie są *lepsze* (w jakim sensie?) od ich lokalnie zbieżnych alternatyw. Okazuje się jednak, że ceną, jaką najczęściej płacimy za globalną zbieżność metody, jest wolne tempo tej zbieżności. Innymi słowy, w przypadku metod zbieżnych globalnie, w celu uzyskania przybliżenia poszukiwanej wartości $\alpha$ z zadaną dokładnością $d > 0$, będziemy musieli wykonać znacznie więcej iteracji, w porównaniu do metod zbieżnych lokalnie. Zdarzają się sytuacje, w których metoda zbieżna jedynie lokalnie w klasie funkcji $\mathcal{F}$, jest metodą zbieżną globalnie w pewnej podklasie $\mathcal{F}_0 \subset \mathcal{F}$.

### 13.1.2. Wykładnik zbieżności

Bardzo ważną własnością charakteryzującą każdą zbieżną metodę iteracyjną jest szybkość tej zbieżności. Jej miarą jest tzw. *wykładnik zbieżności.*

**Definicja 13.1 (wykładnik zbieżności).** *Liczbę $p \geqslant 1$ nazywamy wykładnikiem zbieżności metody iteracyjnej w klasie funkcji $\mathcal{F}$, jeżeli dla każdej funkcji $f \in \mathcal{F}$ istnieją stała $C > 0$ ($dla\ p = 1$ wymagamy, aby $C \in (0,1)$) oraz $n_0 \in \mathbb{N}$ dla których*

$$\|e_{k+1}\| \leqslant C \|e_k\|^p, \qquad dla\ k \geqslant n_0 \tag{13.3}$$

*oraz dla każdej wartości $q > p$ istnieje funkcja $f \in \mathcal{F}$ dla której nie zachodzi*

$$\|e_{k+1}\| \leqslant C \|e_k\|^q. \tag{13.4}$$

Zauważmy, że warunki występujące w powyższej definicji będą spełnione, jeżeli dla każdej funkcji $f \in \mathcal{F}$ ciąg $\left\{\|e_{k+1}\|/\|e_k\|^p\right\}_{k\in\mathbb{N}}$ będzie miał granicę skończoną $C_f$ (dla $p = 1$ dodatkowo $C_f < 1$). Istotnie, z warunku

$$\lim_{k\to+\infty} \frac{\|e_{k+1}\|}{\|e_k\|^p} = C_f$$

wynika, że dla dowolnie wybranej wartości $\varepsilon > 0$ znajdziemy wartość $n_0 \in \mathbb{N}$, dla której

$$\frac{\|e_{k+1}\|}{\|e_k\|^p} \leqslant C_f + \varepsilon, \qquad \text{dla } k \geqslant n_0.$$

Stąd wynika warunek (13.3), a ponieważ w przypadku, gdy $C_f > 0$, dla $q > p$ zachodzi

$$\lim_{k\to+\infty} \frac{\|e_{k+1}\|}{\|e_k\|^q} = +\infty,$$

zatem warunek (13.4) nie może być spełniony dla żadnej funkcji $f \in \mathcal{F}$, dla której $C_f > 0$.

## 13.2. Przykłady metod iteracyjnych

### 13.2.1. Metoda bisekcji

Przypuśćmy, że $f : [a,b] \to \mathbb{R}$ jest funkcją ciągłą dla której $f(a) f(b) < 0$. Na podstawie własności Darboux, warunek ten gwarantuje, że funkcja $f$ posiada w przedziale $(a,b)$ przynajmniej jedno miejsce zerowe. Metoda bisekcji (połowienia) polega na konstrukcji coraz mniejszych przedziałów zawierających miejsce zerowe funkcji $f$.

[1] Przypomnijmy, że liczba $\alpha$ jest miejscem zerowym o krotności $k$ funkcji $f$, jeżeli

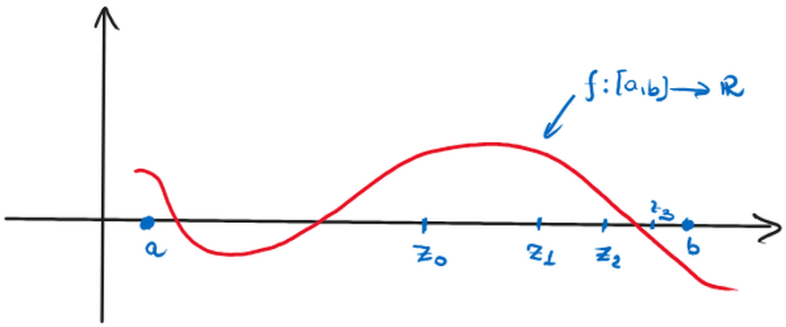
$$f(\alpha) = \ldots = f^{(k-1)}(\alpha) = 0 \neq f^{(k)}(\alpha).$$

**Konstrukcja metody**

Niech $[x_0, y_0] := [a, b]$ oraz przypuśćmy, że wyznaczyliśmy już przedział $[x_k, y_k]$, dla którego $f(x_k) f(y_k) < 0$. Kolejny przedział wyznaczamy w następujący sposób:

dla $z_k = (x_k + y_k)/2$ przyjmujemy

$$[x_{k+1}, y_{k+1}] := \begin{cases} [x_k, z_k] & \text{jeżeli } f(x_k) f(z_k) \leqslant 0 \\ [z_k, y_k] & \text{jeżeli } f(x_k) f(z_k) > 0 \end{cases}.$$




Rysunek 13.1. Metoda bisekcji.

Z powyższej konstrukcji ciągu przedziałów $\{[x_k, y_k]\}_{k\in\mathbb{N}}$ wynika, że

$$\frac{y_{k+1} - x_{k+1}}{y_k - x_k} = \frac{1}{2}.$$

Stąd

$$y_k - x_k = \frac{1}{2^k}(y_0 - x_0), \quad \text{dla } k \in \mathbb{N}. \tag{13.5}$$

Ciągi $\{x_k\}$ oraz $\{y_k\}$, jako ciągi monotoniczne i ograniczone, są zbieżne. Na podstawie zależności (13.5) widzimy, że

$$\lim_{k\to\infty} x_k = \lim_{k\to\infty} y_k = \alpha.$$

**Własności metody**

Metoda bisekcji jest metodą z pamięcią, która w klasie funkcji

$$\mathcal{F} = \{f : [a, b] \to \mathbb{R} : \ f - \text{ciągła}, \ f(a)f(b) < 0\}$$

jest zbieżna globalnie w tempie liniowym (wykładnik zbieżności $p = 1$) ze stałą $C = 1/2$. Jej zaletą są słabe założenia dotyczące funkcji $f$ gwarantujące zbieżność. Istnieją (nietrywialne!) uogólnienia metody bisekcji na przypadek wielowymiarowy.

### 13.2.2. Metoda stycznych (Newtona)

Przypuśćmy, że funkcja $f : (a, b) \to \mathbb{R}$ posiada pierwiastek jednokrotny $\alpha \in (a, b)$, tj. $f(\alpha) = 0$ oraz $f'(\alpha) \neq 0$.

**Konstrukcja metody**

Dysponując przybliżeniem $x_k$ kolejne przybliżenie $x_{k+1}$ wyznaczamy jako miejsce zerowe stycznej do funkcji $f$ w punkcie $x_k$. Zauważmy, że styczna jest wielomianem interpolacyjnym Hermite'a $H \in \pi_1$ dla funkcji $f$ spełniającym warunki interpolacji

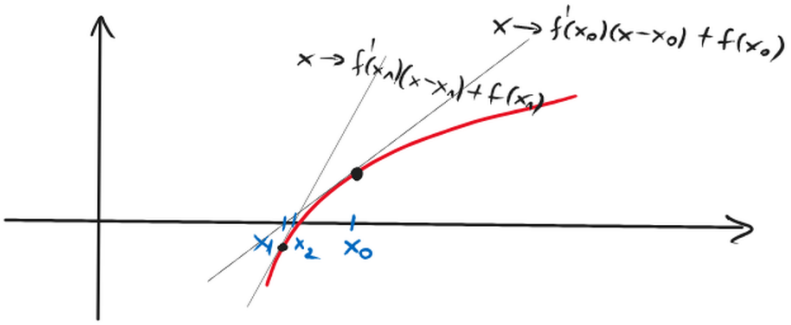
$$f(x_k) = H(x_k), \quad f'(x_k) = H'(x_k).$$

Wiadomo, że

$$H(x) = f(x_k) + f'(x_k)(x - x_k),$$

a z warunku $H(x_{k+1}) = 0$ wynika postać metody Newtona:

$$x_{k+1} = x_k - \frac{f(x_k)}{f'(x_k)}.$$




Rysunek 13.2. Metoda Newtona.

**Własności metody**

Metoda stycznych jest stacjonarną metodą bez pamięci. Wykażemy, że w klasie funkcji

$$\mathcal{F} = \{f : (a,b) \to \mathbb{R} : \ f \in \mathcal{C}^2_{(a,b)}, \ f \text{ posiada miejsce zerowe o krotności jeden}\}$$

metoda ta jest zbieżna lokalnie.

**Twierdzenie 13.1 (o lokalnej zbieżności metody Newtona).** *Jeżeli dla funkcji $f \in \mathcal{C}^2_{(a,b)}$ istnieje punkt $\alpha \in (a,b)$, taki że $f(\alpha) = 0$ oraz $f'(\alpha) \neq 0$, to istnieje $\varepsilon > 0$ o tej własności, że ciąg*

$$x_{k+1} = x_k - \frac{f(x_k)}{f'(x_k)}$$

*jest zbieżny do $\alpha$, o ile $|x_0 - \alpha| < \varepsilon$.*

**Dowód**: Niech $F(x) = x - f(x)/f'(x)$. Zauważmy, że $F(\alpha) = \alpha$ oraz

$$F'(\alpha) = 1 - \left.\frac{(f'(x))^2 - f(x) f''(x)}{(f'(x))^2}\right|_{x=\alpha} = 0.$$

Ponieważ funkcja $F'$ jest ciągła w otoczeniu punktu $\alpha$, zatem dla dowolnej stałej $\lambda \in (0,1)$ istnieje $\varepsilon > 0$, dla którego

$$\left|F'(x)\right| < \lambda < 1 \tag{13.6}$$

dla wszystkich $x \in U = \{x \in (a,b) : |x - \alpha| \leqslant \varepsilon\}$. Wykażemy teraz, że funkcja $F$ jest kontrakcją przekształcającą zbiór $U$ w niego samego.

Dla $x \in U$ mamy

$$|F(x) - \alpha| = |F(x) - F(\alpha)| = \left|F'(\xi)\right| |x - \alpha|$$

dla pewnego $\xi \in U$. Stąd oraz z (13.6) wynika, że

$$|F(x) - \alpha| \leqslant \varepsilon,$$

czyli $F : U \to U$. Ponadto, dla dowolnych $x, y \in U$ istnieje $\eta \in U$ dla którego

$$|F(x) - F(y)| = \left|F'(\eta)\right| |x - y| < \lambda |x - y|.$$

Oznacza to, że $F$ jest kontrakcją. Z twierdzenia Banacha o punkcie stałym (zob. Dodatek B) wnioskujemy, że funkcja $F$ posiada dokładnie jeden punkt stały – jak wiemy jest nim punkt $\alpha$. Ponadto

$$\alpha = \lim_{k\to\infty} x_k,$$

gdzie $x_{k+1} = F(x_k)$ oraz $x_0 \in U$ jest dowolnym punktem startowym. To kończy dowód. ∎

Wykażemy teraz, że metoda Newtona jest zbieżna kwadratowo, tj. jej wykładnik zbieżności to $p = 2$. Dla funkcji $f \in \mathcal{C}^2_{(a,b)}$ mamy

$$e_{k+1} = x_{k+1} - \alpha = x_k - \frac{f(x_k)}{f'(x_k)} - \alpha = e_k - \frac{f(\alpha) + f'(\alpha)\, e_k + \frac{1}{2} f''(\xi_k)\, e_k^2}{f'(\alpha) + f''(\eta_k)\, e_k},$$

gdzie $\xi_k, \eta_k \in \operatorname{conv}(x_k, \alpha)$. Stąd

$$e_{k+1} = \frac{f''(\eta_k) - \frac{1}{2} f''(\xi_k)}{f'(\alpha) + f''(\eta_k)\, e_k} e_k^2$$

i ostatecznie, uwzględniając założenia,

$$\frac{e_{k+1}}{e_k^2} = \frac{f''(\eta_k) - \frac{1}{2} f''(\xi_k)}{f'(\alpha) + f''(\eta_k)\, e_k} \overset{k\to\infty}{\longrightarrow} \frac{f''(\alpha)}{2f'(\alpha)}.$$

Wykazana zbieżność kwadratowa metody Newtona oznacza, że jeżeli na przykład $e_k \approx 10^{-2}$, to $e_{k+1} \approx 10^{-4}$, a $e_{k+2} \approx 10^{-8}$. W praktyce, najczęściej wystarczy wykonanie kilku iteracji metody, aby uzyskać założoną dokładność; ich dokładna liczba zależna jest oczywiście od wymaganej dokładności, od przybliżenia początkowego $x_0$ oraz od wartości $f''(\alpha)/(2f'(\alpha))$.

Istnieje naturalne uogólnienie metody Newtona na przypadek wielowymiarowy: w miejsce odwrotności pochodnej $f'(x_k)$ należy wstawić $(D_f(x_k))^{-1}$, gdzie $D_f(x_k)$ to macierz Jacobiego funkcji $f$ w punkcie $x_k$.[2]

### 13.2.3. Metoda siecznych

Konstrukcja metody siecznych jest bardzo podobna do konstrukcji metody stycznych – jest ona również metodą opartą na wielomianie interpolacyjnym.

Dysponując przybliżeniami $x_{k-1}$ oraz $x_k$, kolejnym przybliżeniem $x_{k+1}$ będzie miejsce zerowe wielomianu interpolacyjnego $L \in \pi_1$ spełniającego warunki interpolacji

$$L(x_{k-1}) = f(x_{k-1}), \quad L(x_k) = f(x_k).$$

Wielomian $L$ jest oczywiście sieczną dla funkcji $f$. Wiemy, że

$$L(x) = f(x_k) + f[x_k, x_{k-1}](x - x_k) = f(x_k) + \frac{f(x_k) - f(x_{k-1})}{x_k - x_{k-1}}(x - x_k),$$

z warunku $L(x_{k+1}) = 0$ wynika postać metody siecznych:

$$x_{k+1} = x_k - f(x_k)\frac{x_k - x_{k-1}}{f(x_k) - f(x_{k-1})}.$$

**Własności metody**

Metoda siecznych jest stacjonarną metodą z pamięcią. Można wykazać, że – przy założeniach podobnych jak w przypadku metody Newtona – jest to metoda zbieżna lokalnie, jej wykładnik zbieżności to $p = (1 + \sqrt{5})/2 \approx 1,618$. Dużą zaletą metody siecznych jest to, że nie wymaga ona znajomości pochodnej funkcji $f$.

[2] Więcej na temat możliwych uogólnień metody Newtona, w tym na przypadek, gdy funkcja $f$ posiada miejsce zerowe o krotności $m > 1$, można przeczytać w rozdziale 5. monografii J. Stoer, R. Bulirsch, *Introduction to Numerical Analysis*, Springer, 2002.

## Rozdział 14

# Algebraiczny problem własny

Skalar $\lambda \in \mathbb{C}$ nazywamy wartością własną macierzy $A \in \mathbb{C}^{n\times n}$, jeżeli istnieje niezerowy wektor $v \in \mathbb{C}^n$, taki że

$$Av = \lambda v;$$

wektor $v$ nazywamy wektorem własnym macierzy $A$ odpowiadającym wartości własnej $\lambda$.

Każda macierz posiada dokładnie $n$ (zespolonych) wartości własnych (liczonych z krotnościami), które równoważnie mogą być definiowane jako pierwiastki wielomianu charakterystycznego

$$\varphi_A : \lambda \to \det(A - \lambda I)$$

macierzy $A$. Zbiór wszystkich wartości własnych macierzy $A$ oznaczamy jako $\sigma(A)$ i nazywamy widmem macierzy $A$.

### 14.1. Sformułowanie zadania

Dla zadanej macierzy $A \in \mathbb{C}^{n\times n}$ poszukujemy jej wartości własnej (jednej, kilku, wszystkich) oraz odpowiadającego jej (im) wektora własnego. Rozważania ograniczymy do macierzy diagonalizowalnych, które są gęstym podzbiorem zbioru wszystkich macierzy.

Przypomnijmy, że macierz $A$ jest diagonalizowalna, jeżeli jest podobna do macierzy diagonalnej, tzn. istnieją macierze nieosobliwa $P$ oraz diagonalna $D = \operatorname{diag}(\lambda_1, \ldots, \lambda_n)$, takie że

$$A = P \operatorname{diag}(\lambda_1, \ldots, \lambda_n) P^{-1}. \tag{14.1}$$

Macierze podobne mają te same wartości własne, zatem elementy diagonalne $\lambda_1, \ldots, \lambda_n$ macierzy $D$ to wartości własne macierzy $A$. Oznaczając kolumny macierzy $P$ jako $v_1, \ldots, v_n$, z równości (14.1) otrzymujemy

$$Av_i = \lambda_i v_i \qquad (i = 1, \ldots, n),$$

co oznacza, że współrzędne $i-$tej kolumny macierzy $P$ tworzą wektor własny odpowiadający wartości własnej $\lambda_i$ macierzy $A$.

Przypomnijmy również, że każda rzeczywista macierz symetryczna (zespolona macierz hermitowska) ma jedynie rzeczywiste wartości własne oraz istnieje baza ortonomalna przestrzeni $\mathbb{R}^n$ (odp. $\mathbb{C}^n$) złożona z jej wektorów własnych, a macierz podobieństwa $P$ występującą w równości (14.1) można wówczas wybrać jako macierz ortogonalną (odp. unitarną, tj. $P^{-1} = P^*$).

## 14.2. Uwarunkowanie ☠

Przykład Wilkinsona (zob. przykład 2.1 na str. 11) pokazuje, że zadanie obliczania pierwiastków wielomianu może być bardzo źle uwarunkowane. Wyznaczając wartości własne macierzy powinniśmy zatem unikać przechodzenia na jawną postać wielomianu charakterystycznego. Często okazuje się, że zadanie obliczania wartości własnych jest bardzo źle uwarunkowane, gdy danymi są współczynniki wielomianu charakterystycznego, podczas, gdy to samo zadanie jest bardzo dobrze uwarunkowane ze względu na pierwotne dane, tzn. współczynniki macierzy, której wartości własne wyznaczamy.

Zajmiemy się teraz zbadaniem wrażliwości wartości własnych macierzy jako funkcji jej współczynników. Przypuśćmy, że $A$ jest macierzą diagonalizowalną postaci (14.1) oraz niech $\lambda$ oznacza dowolną wartość własną macierzy $A+E$, gdzie $E \in \mathbb{C}^{n\times n}$ jest macierzą zaburzeń. Zbadamy jak duża może być wartość

$$\min_{i=1}^{n} |\lambda - \lambda_i| .$$

Niech $x \neq 0$ będzie wektorem własnym macierzy $A+E$ odpowiadającym wartości własnej $\lambda$, tzn.

$$(A+E)\, x = \lambda x,$$

lub równoważnie

$$Ex = (\lambda I - A)\, x.$$

Stąd

$$P^{-1}EPP^{-1}x = P^{-1}\,(\lambda I - A)\, PP^{-1}x.$$

Przyjmując $y = P^{-1}x$, otrzymujemy

$$P^{-1}EPy = P^{-1}\,(\lambda I - A)\, Py = \left(\lambda I - P^{-1}AP\right) y,$$

a uwzględniając (14.1)

$$P^{-1}EPy = (\lambda I - \operatorname{diag}(\lambda_1, \ldots, \lambda_n))\, y = \operatorname{diag}(\lambda - \lambda_1, \ldots, \lambda - \lambda_n)\, y.$$

Dla $\lambda \neq \lambda_i$ wynika stąd, że

$$y = \operatorname{diag}(\lambda - \lambda_1, \ldots, \lambda - \lambda_n)^{-1}\, P^{-1}EPy,$$

zatem

$$\|y\|_2 \leqslant \|\operatorname{diag}(\lambda - \lambda_1, \ldots, \lambda - \lambda_n)^{-1}\|_2 \|P^{-1}\|_2 \|P\|_2 \|E\|_2 \|y\|_2 . \tag{14.2}$$

Ponieważ $y \neq 0$ oraz

$$\|\operatorname{diag}(\lambda - \lambda_1, \ldots, \lambda - \lambda_n)^{-1}\|_2 = \frac{1}{\min_{i=1}^{n} |\lambda - \lambda_i|}$$

zatem z nierówności (14.2) wynika oszacowanie

$$\min_{i=1}^{n} |\lambda - \lambda_i| \leqslant \|P^{-1}\|_2 \|P\|_2 \|E\|_2 . \tag{14.3}$$

Zauważmy, że rozumowanie to możemy powtórzyć dla każdej macierzy nieosobliwej $P$, dla której $P^{-1}AP = \operatorname{diag}(\lambda_1, \ldots, \lambda_n)$. [1] Stąd oraz z (14.3) wynika ostatecznie, że

$$\min_{i=1}^{n} |\lambda - \lambda_i| \leqslant \psi(A) \|E\|_2 ,$$

[1] Zauważmy na przykład, że dla każdej diagonalnej macierzy nieosobliwej $G$ mamy

$$P^{-1}AP = \operatorname{diag}(\lambda_1, \ldots, \lambda_n) \Leftrightarrow (PG)^{-1} A\, (PG) = \operatorname{diag}(\lambda_1, \ldots, \lambda_n) .$$

gdzie

$$\psi(A) = \inf\left\{\|P^{-1}\|_2 \|P\|_2 : P^{-1}AP = \operatorname{diag}(\lambda_1, \ldots, \lambda_n)\right\}.$$

Wartość $\psi(A)$ to maksymalny wpływ macierzy zaburzeń $E$ macierzy $A$ na wartości własne macierzy $A$. Wartość $\psi(A)$ możemy zatem interpretować jako wskaźnik uwarunkowania zadania wyznaczania wartości własnych. Ponieważ

$$1 = \|I\|_2 = \|P^{-1}P\|_2 \leqslant \|P^{-1}\|_2 \|P\|_2,$$

zatem $\psi(A) \geqslant 1$. W przypadku dowolnej macierzy jej wartości własne macierzy mogą być bardzo wrażliwe ze względu na jej zaburzenie. Z kolei, jak zauważyliśmy na wstępie, w przypadku, gdy macierz $A$ jest hermitowska, wśród macierzy podobieństwa $P$ znajdziemy macierz unitarną, dla której

$$\begin{aligned}\|P^{-1}\|_2 \|P\|_2 &= \sqrt{\lambda_{\max}((P^{-1})^*P^{-1})}\sqrt{\lambda_{\max}(P^*P)} = \\ &= \sqrt{\lambda_{\max}(PP^*)}\sqrt{\lambda_{\max}(P^*P)} = \sqrt{\lambda_{\max}(I_n)}\sqrt{\lambda_{\max}(I_n)} = 1,\end{aligned}$$

a w konsekwencji $\psi(A) = 1$. Wynika stąd nierówność

$$\min_{i=1}^{n} |\lambda - \lambda_i| \leqslant \|E\|_2,$$

która pokazuje, że zadanie wyznaczania wartości własnych macierzy hermitowskich jest bardzo dobrze uwarunkowane.

## 14.3. Lokalizacja wartości własnych, twierdzenie Gerszgorina

W wielu zagadnieniach nie potrzebujemy dokładnej znajomości wartości własnych, lecz jedynie pewnej informacji dotyczącej ich lokalizacji. Dla przykładu przypomnijmy, że warunkiem koniecznym i wystarczającym zbieżności dowolnej stacjonarnej metody iteracyjnej rozwiązywania układu równań liniowych jest to, aby wszystkie wartości własne macierzy iterującej definiującej tę metodę leżały w kole jednostkowym (zob. twierdzenie 12.2 na str. 91); ich dokładne wartości nie są potrzebne.

**Stwierdzenie 14.1.** *Dla dowolnej macierzy $A \in \mathbb{C}^{n\times n}$ zachodzi*

$$\sigma(A) \subset \{s \in \mathbb{C} : |s| \leqslant \|A\|\},$$

*gdzie $\|\cdot\|$ to dowolna norma macierzowa zgodna z jakąkolwiek normą wektorową.*

**Dowód**: Dowód pozostawiam jako ćwiczenie (zob. początek dowodu lematu 12.4 na str. 91). ∎

**Przykład 14.1.** *Dla macierzy*

$$A = \begin{pmatrix} 0 & 0,2 & -0,1 \\ -0,1 & 0,18 & -0,12 \\ 0,1 & 0 & 0,7 \end{pmatrix}$$

*mamy*

$$\|A\|_1 = \max_{j=1}^{3} \sum_{i=1}^{3} |a_{ij}| = \max\{0,2; 0,38; 0,92\} = 0,92$$

$$\|A\|_\infty = \max_{i=1}^{3} \sum_{j=1}^{3} |a_{ij}| = \max\{0,3; 0,4; 0,8\} = 0,8$$

$$\|A\|_F = \sqrt{\sum_{i,j} a_{ij}^2} = \sqrt{0,6068} \approx 0,7790$$

*skąd wynika, że*

$$\sigma(A) \subset \{s \in \mathbb{C} : |s| \leqslant 0,7790\}.$$

Bardziej subtelne oszacowanie wartości własnych macierzy wynika z poniższego twierdzenia Gerszgorina. Występujące w nim zbiory $K_1(A), \ldots, K_n(A)$ to tzw. koła Gerszgorina.

**Twierdzenie 14.2 (Gerszgorin 1931).** *Niech* $A = [a_{ij}] \in \mathbb{C}^{n\times n}$ *oraz niech*

$$K_i(A) = \{s \in \mathbb{C} : |s - a_{ii}| \leqslant r_i(A) := \sum_{j\neq i} |a_{ij}|\} \qquad (i = 1, \ldots, n).$$

*Wówczas:*

**(a)** $\sigma(A) \subset K_A$, *gdzie*

$$K_A = \bigcup_{i=1}^{n} K_i(A);$$

**(b)** *jeżeli suma dowolnych* $k$ *kół Gerszgorina tworzy zbiór* $\Omega_1$ *rozłączny z sumą* $\Omega_2$ *pozostałych* $n-k$ *kół, to w zbiorze tym znajduje się dokładnie* $k$ *wartości własnych macierzy* $A$.

**Dowód**: Niech $\lambda \in \sigma(A)$ będzie dowolną wartością własną macierzy $A$, a $v = (v_1, \ldots, v_n)^T$ odpowiadającym jej unormowanym wektorem własnym, tj.

$$Av = \lambda v \quad \text{oraz} \quad \|v\|_\infty = 1.$$

Mamy więc dla $i = 1, \ldots, n$

$$\lambda v_i = \sum_{j=1}^{n} a_{ij} v_j = a_{ii} v_i + \sum_{j\neq i} a_{ij} v_j,$$

a stąd

$$(\lambda - a_{ii}) v_i = \sum_{j\neq i} a_{ij} v_j.$$

Dla wszystkich indeksów $i$ spełniających warunek $|v_i| = 1 = \|v\|_\infty$ zachodzi nierówność

$$|\lambda - a_{ii}| = |\sum_{j\neq i} a_{ij} v_j| \leqslant \sum_{j\neq i} |a_{ij}|\,|v_j| \leqslant \sum_{j\neq i} |a_{ij}| = r_i(A),$$

która kończy dowód pierwszej cześci tezy.

Aby zakończyć dowód rozważmy macierz $D = \operatorname{diag}(a_{11}, \ldots, a_{nn})$ oraz niech

$$B(t) = D + t \cdot (A - D), \qquad 0 \leqslant t \leqslant 1.$$

Na podstawie wykazanej już pierwszej części twierdzenia,

$$\sigma(B(t)) \subset K_{B(t)}, \qquad \text{dla} \quad 0 \leqslant t \leqslant 1.$$

Zauważmy, że zbiór $K_{B(t)}$ jest sumą kół $K_1(B(t)), \ldots, K_n(B(t))$ o środkach w punktach $a_{11}, \ldots, a_{nn}$ i o promieniach $r_1(B(t)), \ldots, r_n(B(t))$, gdzie dla $t \in [0, 1]$

$$r_i(B(t)) = t \cdot r_i(B(1)) = t \cdot r_i(A) \leqslant r_i(A).$$

To oznacza, że

$$K_i(B(t)) \subset K_i(A),$$

dla wszystkich $i \in \{1, \ldots, n\}$ oraz $t \in [0, 1]$. Przypuśćmy, że

$$\Omega_1 = K_{i_1}(A) \cup \ldots \cup K_{i_k}(A),$$

dla $i_1, \ldots, i_k \in \{1, \ldots, n\}$. Zauważmy, że dokładnie $k$ wartości własnych macierzy $B(0) = D$ leży w zbiorze $\Omega_1$ – tymi wartościami własnymi są $a_{i_1,i_1}, \ldots, a_{i_k,i_k}$, pozostałe $n-k$ wartości własnych leży w rozłącznym z $\Omega_1$ zbiorze $\Omega_2$. Ponieważ wartości własne macierzy są ciągłymi funkcjami jej elementów, wynika stąd, że zbiór $\Omega_1$ zawiera dokładnie $k$ wartości własnych macierzy $B(t)$ dla wszystkich wartości $t \in [0, 1]$ – nie jest możliwe *przejście* ze zwartego zbioru $\Omega_1$ do rozłącznego z nim zwartego zbioru $\Omega_2$ w sposób ciągły. Dla $t = 1$ otrzymujemy tezę. ∎

**Uwaga 14.1.** *Ponieważ macierze $A$ oraz $A^T$ mają te same wartości własne, zatem stosując twierdzenie Gerszgorina jednocześnie do obydwu tych macierzy otrzymujemy*

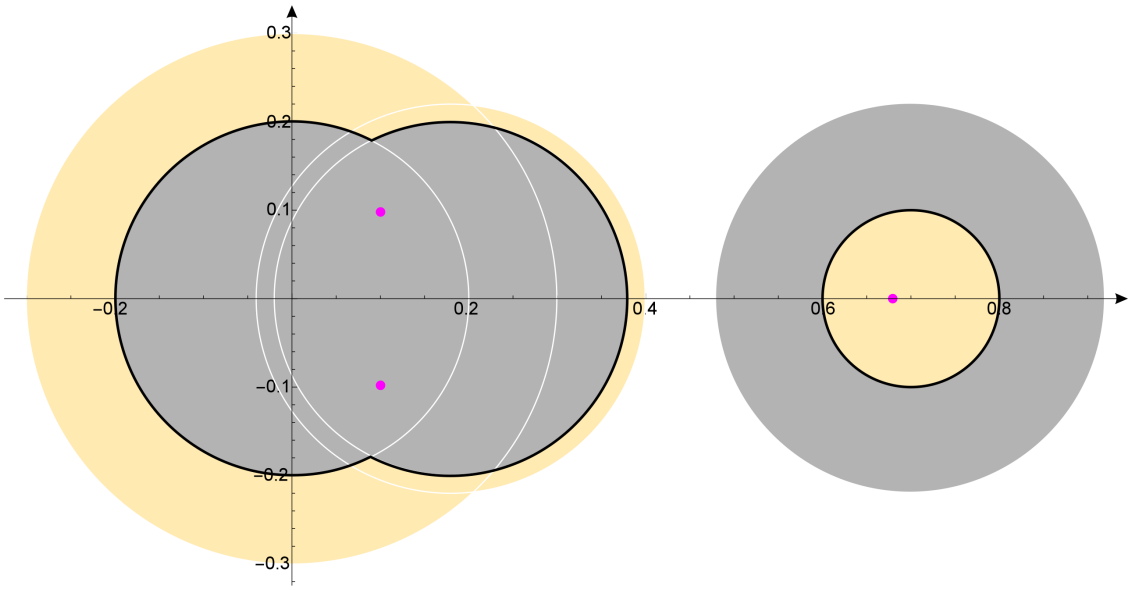
$$\sigma(A) \subset K_A \cap K_{A^T}.$$

**Uwaga 14.2.** *Przypuśćmy, że koło Gerszgorina $K_i(A)$ macierzy rzeczywistej $A \in \mathbb{R}^{n\times n}$ rozłączne jest z jej pozostałymi $n-1$ kołami. Wiemy, że wówczas koło to zawiera dokładnie jedną wartość własną macierzy $A$. Ponieważ nierzeczywiste wartości własne macierzy rzeczywistej występują zawsze jako para $\lambda$ oraz $\overline{\lambda}$, zatem ta jedyna wartość własna zawarta w kole $K_i(A)$ musi być rzeczywista (zob. rysunek 14.1).*

**Przykład 14.2.** *Dla macierzy z przykładu 14.1 mamy*

$$\begin{aligned} K_1(A) &= \{s \in \mathbb{C} : |s| \leqslant 0,3\}, \\ K_2(A) &= \{s \in \mathbb{C} : |s - 0,18| \leqslant 0,22\}, \\ K_3(A) &= \{s \in \mathbb{C} : |s - 0,7| \leqslant 0,1\}. \end{aligned}$$

*Ponieważ koło $K_3(A)$ jest rozłączne z pozostałymi dwoma kołami, zatem zawiera ono jedną wartość własną macierzy $A$, pozostałe dwie wartości własne zawarte są w sumie kół $K_1(A)$ oraz $K_2(A)$.*




Rysunek 14.1. Wartości własne $\lambda_1 = \overline{\lambda_2} \approx -0,10029 + 0,0983i$; $\lambda_3 \approx 0,6794$ (różowe) oraz koła Gerszgorina $K_1(A), K_2(A), K_3(A)$ (żółte) i $K_1(A^T), K_2(A^T), K_3(A^T)$ (szare) dla macierzy $A$ z przykładu 14.1; czarna linia to brzeg obszaru $K_A \cap K_{A^T}$ zawierającego jej wartości własne.

## 14.4. Metoda potęgowa oraz ilorazy Rayleigha

Przypuśćmy, że $A \in \mathbb{C}^{n\times n}$ jest macierzą diagonalizowalną o wartościach własnych $\lambda_1, \ldots, \lambda_n$, takich że

$$|\lambda_1| > |\lambda_2| \geqslant \ldots \geqslant |\lambda_n|, \tag{14.4}$$

którym odpowiadają liniowo niezależne wektory własne $v_1, \ldots, v_n$. Przedstawimy teraz iteracyjną metodę potęgową wyznaczania wartości własnej $\lambda_1$ oraz odpowiadającego jej wektora własnego.

Niech $z_0 \in \mathbb{C}^n$ będzie dowolnym wektorem startowym. Istnieją zatem stałe $c_1, \ldots, c_n$ dla których

$$z_0 = c_1 v_1 + \ldots + c_n v_n. \tag{14.5}$$

W dalszym ciągu zakładamy, że $c_1 \neq 0$.[2] Kolejne elementy ciągu $z_1, z_2, \ldots$ definiujemy rekurencyjnie: przypuśćmy, że znamy już $z_k$; wówczas

$$z_{k+1} = \frac{Az_k}{\|Az_k\|}.$$

Łatwo zauważyć, że

$$z_{k+1} = \frac{Az_k}{\|Az_k\|} = \frac{A\frac{Az_{k-1}}{\|Az_{k-1}\|}}{\left\|A\frac{Az_{k-1}}{\|Az_{k-1}\|}\right\|} = \frac{A^2 z_{k-1}}{\|A^2 z_{k-1}\|},$$

a w konsekwencji

$$z_k = \frac{A^k z_0}{\|A^k z_0\|}, \qquad \text{dla } k \in \mathbb{N}.$$

Dodatkowo, z warunku (14.5) otrzymujemy

$$A^k z_0 = \sum_{i=1}^{n} c_i A^k v_i = \sum_{i=1}^{n} c_i \lambda_i^k v_i.$$

Stąd wynikają równości

$$z_k = \frac{A^k z_0}{\|A^k z_0\|} = \frac{\sum\limits_{i=1}^{n} c_i \lambda_i^k v_i}{\|\sum\limits_{i=1}^{n} c_i \lambda_i^k v_i\|} = \frac{\lambda_1^k}{|\lambda_1^k|} \frac{c_1 v_1 + \sum\limits_{i=2}^{n} c_i \left(\lambda_i/\lambda_1\right)^k v_i}{\|c_1 v_1 + \sum\limits_{i=2}^{n} c_i \left(\lambda_i/\lambda_1\right)^k v_i\|}. \tag{14.6}$$

Na podstawie założenia (14.4) mamy

$$\sum_{i=2}^{n} c_i \left(\frac{\lambda_i}{\lambda_1}\right)^k v_i \overset{k\to+\infty}{\longrightarrow} 0, \tag{14.7}$$

zatem z zależności (14.6) wynika, że ciąg $\{z_k\}$ *zbiega* do wektora własnego odpowiadającego wartości własnej $\lambda_1$ macierzy $A$. Warto w tym miejscu wyjaśnić powód, dla którego użyte tu słowo *zbiega* zostało zapisane kursywą. Z zależności (14.7) wynika, że

$$\frac{c_1 v_1 + \sum\limits_{i=2}^{n} c_i \left(\lambda_i/\lambda_1\right)^k v_i}{\|c_1 v_1 + \sum\limits_{i=2}^{n} c_i \left(\lambda_i/\lambda_1\right)^k v_i\|} \overset{k\to+\infty}{\longrightarrow} c_1 v_1/\|c_1 v_1\|;$$

wektor $c_1 v_1/\|c_1 v_1\|$ to oczywiście wektor własny macierzy $A$ odpowiadający jej wartości własnej $\lambda_1$. Zauważmy jednak, że występujący po prawej stronie wyrażania (14.6) ciąg $(\lambda_1^k/|\lambda_1^k|)_{k\in\mathbb{N}}$ na ogół nie jest zbieżny. W efekcie ciąg $(z_k)_{k\in\mathbb{N}}$ nie jest zbieżny w zwykłym sensie. Z drugiej strony łatwo stwierdzić, że jeżeli wektor $v \in \mathbb{C}^n$ jest wektorem własnym macierzy $A$ odpowiadającym jej wartości własnej $\lambda$, to dla dowolnego ciągu skalarów niezerowych $(\alpha_k)_{k\in\mathbb{N}} \subset \mathbb{C}$, wektory $\alpha_k v$, dla wszystkich $k \in \mathbb{N}$, również są wektorami własnym macierzy $A$ odpowiadającymi jej wartości własnej $\lambda$.

[2] Warunek $c_1 = 0$ oznaczałby, że punkt $z_0$ należy do podprzestrzeni rozpiętej przez wektory $v_2, \ldots, v_n$; przy losowym wyborze wartości $z_0$ warunek $c_1 \neq 0$ najczęściej będzie spełniony.

Wyznaczymy teraz tzw. ilorazy Rayleigha $\langle Az_k, z_k\rangle$. Rozumując podobnie jak powyżej otrzymujemy kolejno

$$\langle Az_k, z_k\rangle = \left\langle \frac{A^{k+1}z_0}{\|A^k z_0\|}, \frac{A^k z_0}{\|A^k z_0\|} \right\rangle = \frac{\langle A^{k+1}z_0, A^k z_0\rangle}{\|A^k z_0\|^2} = \frac{\left\langle \sum_{i=1}^{n} c_i\lambda_i^{k+1}v_i, \sum_{i=1}^{n} c_i\lambda_i^k v_i \right\rangle}{\left\langle \sum_{i=1}^{n} c_i\lambda_i^k v_i, \sum_{i=1}^{n} c_i\lambda_i^k v_i \right\rangle} =$$

$$= \frac{\lambda_1 |\lambda_1|^{2k}}{|\lambda_1|^{2k}} \frac{\left\langle c_1v_1 + \sum_{i=2}^{n} c_i (\lambda_i/\lambda_1)^{k+1} v_i, c_1v_1 + \sum_{i=2}^{n} c_i (\lambda_i/\lambda_1)^{k} v_i \right\rangle}{\left\langle c_1v_1 + \sum_{i=2}^{n} c_i (\lambda_i/\lambda_1)^{k} v_i, c_1v_1 + \sum_{i=2}^{n} c_i (\lambda_i/\lambda_1)^{k} v_i \right\rangle}.$$

Stąd oraz z (14.7) wynika, że

$$\langle Az_k, z_k\rangle \overset{k\to+\infty}{\longrightarrow} \lambda_1.$$

**Uwaga 14.3.** *Wartościami własnymi macierzy $A^{-1}$ są odwrotności wartości własnych macierzy $A$, zatem stosując metodę potęgową oraz ilorazy Rayleigha do macierzy $A^{-1}$ możemy wyznaczyć najmniejszą co do modułu wartość własną macierzy $A$ oraz odpowiadający jej wektor własny. Z kolei wartości własne macierzy $A + \alpha I$ są postaci $\lambda + \alpha$, gdzie $\lambda \in \sigma(A)$, zatem stosując przedstawione techniki do macierzy $A + \alpha I$ możemy wyznaczyć wartości własne oraz odpowiadające im wektory własne macierzy $A$ znajdujące się najdalej oraz najbliżej dowolnej wartości $\alpha \in \mathbb{C}$.*

Potrafimy już wyznaczać największą co do modułu wartość własną oraz odpowiadający jej wektor własny. Na zakończenie przedstawimy jeden z wariantów tzw. *deflacji* umożliwiającej wyznaczenie kolejnej pary własnej.

## 14.5. Deflacja

Przypuśćmy, że znamy (wystarczą dobre przybliżenia) parę własną $(\lambda_1, v_1)$ macierzy $A$, tj.

$$Av_1 = \lambda_1 v_1. \tag{14.8}$$

Niech $P$ będzie macierzą przekształcającą wektor $v_1$ w wektor $e_1 = (1, 0, \ldots, 0)^T \in \mathbb{R}^n$, tj.

$$Pv_1 = e_1; \tag{14.9}$$

macierz taką można łatwo skonstruować odpowiednio skalując macierz Householdera przekształcającą wektor $v_1$ na kierunek wektora $e_1$ (zob. podrozdział 11.3). Z równości (14.8) otrzymujemy

$$\left(PAP^{-1}\right) Pv_1 = \lambda_1 Pv_1,$$

a stąd, na podstawie (14.9), wynika, że

$$\left(PAP^{-1}\right) e_1 = \lambda_1 e_1.$$

Zauważmy, że wektor $\left(PAP^{-1}\right) e_1$ to pierwsza kolumna macierzy $PAP^{-1}$, zatem macierz $PAP^{-1}$ jest postaci

$$PAP^{-1} = \left(\begin{array}{c|c} \lambda_1 & \\ \hline 0 & \\ \vdots & \widetilde{A} \\ 0 & \end{array}\right).$$

Na podstawie podobieństwa macierzy $A$ oraz $PAP^{-1}$ wnioskujemy, że wartościami własnymi macierzy $\widetilde{A}$ są wartości własne $\lambda_2, \ldots, \lambda_n$ macierzy $A$. Przejście od problemu własnego dla macierzy $A$ do problemu własnego dla macierzy $\widetilde{A}$ nazywamy *deflacją*. Jeżeli spełniony jest warunek

$$|\lambda_2| > |\lambda_3| \geqslant \ldots \geqslant |\lambda_n|,$$

stosując do macierzy $\widetilde{A}$ metodę potęgową oraz wykorzystując ilorazy Rayleigha możem wyznaczyć kolejną parę własną $(\lambda_2, v_2)$ macierzy $A$.

# Dodatek A

# Norma macierzowa indukowana przez normę wektorową

Niech $\|\cdot\|_p$ oznacza $p$–tą normę wektorową Höldera, tj. dla $x = (x_1, \ldots, x_n) \in \mathbb{R}^n$

$$\|x\|_p = \begin{cases} \left( \sum\limits_{i=1}^{n} |x_i|^p \right)^{1/p} & \text{dla } 1 \leqslant p < \infty \\ \max\limits_{i=1}^{n} |x_i| & \text{dla } p = \infty \end{cases} .$$

Można łatwo wykazać, że funkcja $\mathbb{R}^{m\times n} \ni A \to \|A\|_{p,q} \in \mathbb{R}$ określona wzorem

$$\|A\|_{p,q} = \sup \left\{ \frac{\|Ax\|_p}{\|x\|_q} : x \neq 0 \right\} = \max \left\{ \|Ax\|_p : \|x\|_q = 1 \right\}$$

jest normą w przestrzeni $\mathbb{R}^{m\times n}$. Jest to tzw. *norma macierzowa indukowana przez normy wektorowe* $\|\cdot\|_p$, $\|\cdot\|_q$. Dla uproszczenia zapisu przyjmujemy, że $\|\cdot\|_{p,p} = \|\cdot\|_p$.

Z powyższej definicji wynikają następujące własności normy macierzowej (ich dowody pozostawiam jako proste ćwiczenie):

- $\|I_n\|_p = 1$;
- dla dowolnych macierzy $A \in \mathbb{R}^{m\times n}$ oraz wektora $x \in \mathbb{R}^n$

$$\|Ax\|_p \leqslant \|A\|_{p,q} \|x\|_q ; \tag{A.1}$$

- dla dowolnych macierzy $A, B \in \mathbb{R}^{n\times n}$

$$\|AB\|_p \leqslant \|A\|_p \|B\|_p ;$$

- dla dowolnej macierzy $A \in \mathbb{R}^{n\times n}$

$$\|A\|_p \geqslant \rho(A) = \max_{\lambda\in\sigma(A)} |\lambda| .$$

Nierówność (A.1) to tzw. *warunek zgodności normy macierzowej z normą wektorową.*

Ze względów praktycznych szczególnie istotną rolę odgrywają normy macierzowe $\|\cdot\|_1$, $\|\cdot\|_2$ oraz $\|\cdot\|_\infty$.

**Stwierdzenie A.1.** *Dla dowolnej macierzy $A \in \mathbb{R}^{n\times n}$ zachodzi*

- $\|A\|_1 = \max\limits_j \sum\limits_i |a_{ij}|$,
- $\|A\|_\infty = \max\limits_i \sum\limits_j |a_{ij}|$,
- $\|A\|_2 = \sqrt{\lambda_{\max}(A^TA)}$.

**Dowód**: Niech $y = Ax$, gdzie $\|x\|_1 = 1$. Wówczas

$$\begin{aligned}\|Ax\|_1 = \|y\|_1 = \sum_i |y_i| = \sum_i |\sum_j a_{ij}x_j| \leqslant \sum_i \sum_j |a_{ij}x_j| = \\ = \sum_j \sum_i |a_{ij}||x_j| = \sum_j \left( |x_j| \sum_i |a_{ij}| \right) \leqslant \sum_j \left( |x_j| \max_j \sum_i |a_{ij}| \right) = \\ = \max_j \sum_i |a_{ij}| \cdot \sum_j |x_j| = \max_j \sum_i |a_{ij}| \cdot \|x\|_1 = \max_j \sum_i |a_{ij}|.\end{aligned}$$

To oznacza, że

$$\|A\|_1 \leqslant \max_j \sum_i |a_{ij}|.$$

Aby wykazać nierówność w drugą stronę wystarczy obliczyć wartość $\|Ae_k\|_1$, gdzie $e_i$ to $i$-ty wektor bazy kanonicznej, a $k$ to indeks dla którego

$$\max_j \sum_i |a_{ij}| = \sum_i |a_{ik}|.$$

Wówczas

$$\|Ae_k\|_1 = \sum_i |a_{ik}|,$$

skąd wynika nierówność

$$\|A\|_1 = \max\{\|Ax\|_1 : \|x\|_1 = 1\} \geqslant \|Ae_k\|_1 = \max_j \sum_i |a_{ij}|,$$

która kończy dowód punktu pierwszego. Punkt drugi dowodzimy analogicznie.

W celu udowodnienia ostatniej równości, przypomnijmy, że macierz $A^TA$, jako macierz symetryczna nieujemnie określona, posiada jedynie nieujemny wartości własne $\lambda_1, \ldots, \lambda_n$, którym odpowiadają wektory własne $v_1, \ldots, v_n$ (tj. $Av_i = \lambda_i v_i$). Wektory te można wybrać w ten sposób, aby stanowiły bazę ortonormalną (w sensie naturalnego iloczny skalarnego) przestrzeni $\mathbb{R}^n$ (tj. $v_i^T v_k = \delta_{ik}$). Dowolny wektor $x \in \mathbb{R}^n$ jest zatem kombinacją liniową wektorów $v_1, \ldots, v_n$

$$x = \sum_{i=1}^n c_i v_i.$$

Stąd

$$\begin{aligned}\|x\|_2^2 = x^Tx = \left(\sum_{i=1}^n c_i v_i\right)^T \left(\sum_{i=1}^n c_i v_i\right) = \left(\sum_{i=1}^n c_i v_i^T\right) \left(\sum_{i=1}^n c_i v_i\right) = \\ = \sum_{i,k} c_i c_k v_i^T v_k = \sum_i c_i^2.\end{aligned}$$

Podobnie,

$$\|Ax\|_2^2 = x^T A^T Ax = \left(\sum_{i=1}^n c_i v_i\right)^T A^T A \left(\sum_{i=1}^n c_i v_i\right) = \left(\sum_{i=1}^n c_i v_i^T\right) A^T A \left(\sum_{i=1}^n c_i v_i\right) =$$
$$= \left(\sum_{i=1}^n c_i v_i^T\right) \left(\sum_{i=1}^n c_i A^T A v_i\right) = \left(\sum_{i=1}^n c_i v_i^T\right) \left(\sum_{i=1}^n c_i \lambda_i v_i\right) = \sum_{i,k} c_i c_k \lambda_k v_i^T v_k =$$
$$= \sum_i c_i^2 \lambda_i \leqslant \lambda_{\max}\left(A^T A\right) \sum_i c_i^2 = \lambda_{\max}\left(A^T A\right) \|x\|_2^2$$

skąd wynika, że

$$\|A\|_2 \leqslant \sqrt{\lambda_{\max}\left(A^T A\right)}.$$

Z kolei, dla unormowanego wektora własnego $v_{\max}$ macierzy $A^T A$ odpowiadającego jej maksymalnej wartości własnej otrzymujemy nierówność w drugą stronę:

$$\|A\|_2 = \max\left\{\|Ax\|_2 : \|x\|_2 = 1\right\} \geqslant \|Av_{\max}\|_2 = \sqrt{v_{\max}^T A^T A v_{\max}} =$$
$$= \sqrt{v_{\max}^T \lambda_{\max}\left(A^T A\right) v_{\max}} = \sqrt{\lambda_{\max}\left(A^T A\right)} \sqrt{v_{\max}^T v_{\max}} = \sqrt{\lambda_{\max}\left(A^T A\right)}.$$

To kończy dowód. ∎

Innym, bardzo ważnym przykładem normy macierzowej jest tzw. norma Frobeniusa

$$\|A\|_F = \sqrt{\sum_{i,j} |a_{ij}|^2}.$$

Jak łatwo się przekonać, norma ta nie jest indukowana przez żadną normę wektorową (aby to wykazać wystarczy obliczyć normę macierzy jednostkowej), ale jest zgodna z normą $\|\cdot\|_2$. Istotnie, z zależności

$$\|A\|_2^2 = \lambda_{\max}\left(A^T A\right) \leqslant \operatorname{tr}\left(A^T A\right) = \sum_{i,j} |a_{ij}|^2 = \|A\|_F^2$$

wynika warunek zgodności

$$\|Ax\|_2 \leqslant \|A\|_2 \|x\|_2 \leqslant \|A\|_F \|x\|_2 .$$

## Dodatek B

# Twierdzenie Banacha o punkcie stałym

Niech $(X, d)$ będzie przestrzenią metryczną zupełną.

**Definicja B.1.** *Odwzorowanie $T : X \to X$ jest odwzorowaniem zwężającym (kontrakcją), jeżeli istnieje stała $\lambda \in (0, 1)$, taka że*

$$d\left(T\left(x\right), T\left(y\right)\right) \leqslant \lambda d\left(x, y\right)$$

*dla wszystkich $x, y \in X$.*

Każda kontrakcja spełnia warunek Lipschitza, jest zatem funkcją ciągłą. Przypomnijmy, że element $x^* \in X$, dla którego $T\left(x^*\right) = x^*$, nazywamy punktem stałym odwzorowania $T$.

**Twierdzenie B.1 (Banacha o punkcie stałym).** *Jeżeli $(X, d)$ jest przestrzenią zupełną, a odwzorowanie $T : X \to X$ jest odwzorowaniem zwężającym, to*

**(a)** *odwzorowanie $T$ ma dokładnie jeden punkt stały w $X$;*
**(b)** *ciąg $x_{k+1} = T\left(x_k\right)$ jest zbieżny do punktu stałego odwzorowania $T$ niezależnie od wyboru wartości początkowej $x_0 \in X$.*

**Dowód**: Niech $x_0 \in X$ będzie dowolnym punktem oraz rozważmy ciąg

$$x_{k+1} = T\left(x_k\right), \qquad k = 0, 1, 2, \ldots.$$

Odwzorowanie $T$ jest zwężające zatem dla dowolnego $k \in \mathbb{N}$

$$d\left(x_{k+1}, x_k\right) = d\left(T\left(x_k\right), T\left(x_{k-1}\right)\right) \leqslant \lambda d\left(x_k, x_{k-1}\right) \leqslant \lambda^2 d\left(x_{k-1}, x_{k-2}\right) \leqslant \ldots \leqslant \lambda^k d\left(x_1, x_0\right).$$

Stąd, dla dowolnych $m > n$

$$\begin{aligned} d\left(x_m, x_n\right) &\leqslant d\left(x_m, x_{m-1}\right) + d\left(x_{m-1}, x_{m-2}\right) + \ldots + d\left(x_{n+1}, x_n\right) \leqslant \\ &\leqslant \lambda^{m-1} d\left(x_1, x_0\right) + \lambda^{m-2} d\left(x_1, x_0\right) + \ldots + \lambda^n d\left(x_1, x_0\right) = \\ &= d\left(x_1, x_0\right) \sum_{j=n}^{m-1} \lambda^j < d\left(x_1, x_0\right) \sum_{j=n}^{+\infty} \lambda^j = d\left(x_1, x_0\right) \frac{\lambda^n}{1-\lambda}. \end{aligned}$$

Otrzymana nierówność pozwala wnioskować, że dla dowolnego $\varepsilon > 0$ oraz liczby naturalnej $N$ spełniającej warunek

$$d\left(x_1, x_0\right) \frac{\lambda^N}{1-\lambda} < \varepsilon$$

zachodzi

$$d\left(x_m, x_n\right) < \varepsilon$$

dla dowolnych $m, n > N$. To oznacza, że ciąg $\{x_k\}$ jest ciągiem Cauchy'ego. Zupełność przestrzeni $X$ gwarantuje, że jest on zbieżny, a jego granica $x^*$ leży w $X$. Stąd, na podstawie ciągłości $T$,

$$T\left(x^*\right) = T\left(\lim_{k\to\infty} x_k\right) = \lim_{k\to\infty} T\left(x_k\right) = \lim_{k\to\infty} x_{k+1} = x^*.$$

Odwzorowanie $T$ posiada więc punkt stały. Zauważmy na koniec, że odwzorowanie zwężające nie może mieć dwóch różnych punktów stałych. Istotnie, dla $x = T\left(x\right)$ oraz $y = T\left(y\right)$ mielibyśmy

$$d\left(x, y\right) = d\left(T\left(x\right), T\left(y\right)\right) \leqslant \lambda d\left(x, y\right),$$

skąd wynika, że $d(x, y) = 0$, czyli $x = y$. ■

# Dodatek C

# Poetycki wielomian Wilkinsona

**Poetycki wielomian Wilkinsona**
Autor: Marysia Pawlak[1]

Oto nowa ma wytyczna:
Analiza numeryczna!
I choć inne przy niej bledną
Ta się przyda mi na pewno!

Kto by wiedział, kto podejrzał,
Kto tak mocno w siebie wejrzał,
Że matematyka wyższa,
Ta trudniejsza, mniej nam bliższa,
Życia sensem moim będzie,
Serce moje też zdobędzie,
Rozum ułagodzi srogi,
I pokaże lepsze drogi?

Bo tak oto w ławce siedzą,
Wciąż niewiele więcej wiedząc,
Usłyszałam zdanie ważne,
Wręcz doniosłe i poważne.

Że na kartce swej Wilkinson
Powziął nową myśl i inszą
Przedstawiając wielomiany
W rys każdemu bardziej znany.

I jak wszyscyśmy odkryli,
Wplątał w to stado motyli,
Z pogodą złączył Lorenza,
I teorie porozkręcał.

Dumnie spojrzał w lustro,
Zmiennych użył mnóstwo,
Jeszcze więcej włożył potęg,
Robiąc z nas przy tym miernotę.

Bo młodzieniec byłby dziwny,
Znając wielomian perfidny.
Gdzie zaburzenie małe środka,
Z efektu zmianą wielką się spotka.

Tak oto bogata w wiedzę,
Przez uczelni przeszłam miedzę,
Mając nową swą wytyczną,
Analizę numeryczną.

[1] http://www.emultipoetry.eu/de/poem/8273,poetycki-wielomian-wilkinsona