

Matematyka Obliczeniowa

(kurs podstawowy dla studentów UW)

Leszek Plaskota
Instytut Matematyki Stosowanej i Mechaniki
Uniwersytet Warszawski

14 czerwca 2020

Spis treści

1	Specyfika obliczeń numerycznych	1
1.1	Zadania numeryczne, przykłady	1
1.2	Model obliczeniowy	2
1.3	Arytmetyka zmiennoprzecinkowa	4
2	Algorytmy i ich własności	9
2.1	Numeryczna poprawność algorytmu	10
2.2	Rola uwarunkowania zadania	11
2.3	Przykłady	13
3	Układy równań liniowych	19
3.1	Układy z macierzą trójkątną	19
3.2	Eliminacja Gaussa	20
3.3	Rozkład trójkątno-trójkątny macierzy	21
4	Analiza błędów w eliminacji Gaussa	28
4.1	Układy z macierzą trójkątną	28
4.2	Poprawność rozkładu trójkątno-trójkątnego	29
4.3	Poprawność rozwiązywania układu	31
4.4	Uwarunkowanie macierzy, a błąd w fl_ν	32
5	Zadanie wygładzania liniowego	36
5.1	Układ normalny	36
5.2	Odbicia Householdera	38
5.3	Algorytm dla zadania wygładzania	39
6	Interpolacja wielomianowa	45
6.1	Sformułowanie zadania interpolacji	45
6.2	Wybór bazy wielomianowej	46
6.3	Algorytm różnic dzielonych	48
6.4	Przypadek węzłów wielokrotnych	49
7	Interpolacja a aproksymacja	54
7.1	Błąd interpolacji	54
7.2	Wielomiany Czebyszewa	56
7.3	Optymalna aproksymacja wielomianowa	58
7.4	Interpolacja kawałkami wielomianowa	61

8	Interpolacja funkcjami sklejanymi	66
8.1	Co to są funkcje sklepane?	66
8.2	Interpolacja i gładkość	67
8.3	Kubiczne funkcje sklepane	69
9	Całkowanie numeryczne	75
9.1	Co to są kwadratury?	75
9.2	Błąd kwadratur interpolacyjnych	76
9.3	Kwadratury złożone	80
9.4	Przyspieszanie zbieżności kwadratur	81
10	Całkowanie a aproksymacja	88
10.1	Rząd kwadratury	88
10.2	Ciągi wielomianów ortogonalnych	90
10.3	Kwadratury Gaussa	93
11	Iteracje dla równań liniowych	99
11.1	Kiedy stosujemy iteracje?	99
11.2	Metoda Czebyszewa	101
11.3	Metoda najszybszego spadku	105
12	Iteracje dla równań nieliniowych	109
12.1	Bisekcja	109
12.2	Iteracje proste	110
12.3	Metody interpolacyjne	112
12.4	Obliczanie zer wielomianów	114

Rozdział 1

Specyfika obliczeń numerycznych

Matematyka obliczeniowa, nazywana również w trochę węższym sensie *analizą numeryczną*, jest dziedziną wiedzy zajmującą się problemami obliczeniowymi i konstrukcją algorytmów rozwiązywania zadań matematycznych, w których interesuje nas wartość numeryczna rozwiązania.

Aby w ogóle mówić w problemie obliczeniowym, musimy najpierw określić:

- dane początkowe i cel obliczeń, czyli sformułować *zadanie numeryczne*,
- dostępne środki obliczeniowe, czyli zdefiniować *model obliczeniowy*.

1.1 Zadania numeryczne, przykłady

Sformułowanie zadania polega na sprecyzowaniu tego, co mamy i co chcemy uzyskać. Formalnie polega to na zdefiniowaniu przestrzeni danych F , przestrzeni wyników G i wskazaniu zależności $f \mapsto g$, gdzie $f \in F$ i $g \in G$, między danymi a wynikami. Zależność tą reprezentuje *operator rozwiązania*

$$S : F \rightarrow G.$$

Będziemy zainteresowani zadaniami *numerycznymi*. Są to zadania, dla których wynikiem g jest skończony ciąg liczb rzeczywistych, czyli

$$G \subset \bigcup_{k=0}^{\infty} \mathbb{R}^k$$

(przyjmujemy przy tym, że $\mathbb{R}^0 = \emptyset$). Zbiór danych może być w zasadzie dowolny, ale my będziemy dla uproszczenia rozpatrywać tylko zadania dla których $F \subset \mathbb{R}^m$, albo ogólniej, dla których F jest pewną klasą funkcji zdefiniowanych na ustalonym zbiorze $D \subset \mathbb{R}^d$.

Przykład 1.1 (*Równanie kwadratowe.*) Załóżmy, że chcemy obliczyć wszystkie rzeczywiste pierwiastki równania

$$x^2 - 2p + q = 0,$$

dla danych liczb rzeczywistych p i q . Wtedy $F = \mathbb{R}^2$, $G = \mathbb{R}^0 \cup \mathbb{R}^1 \cup \mathbb{R}^2$, oraz

$$S(p, q) = \begin{cases} \emptyset & \Delta < 0, \\ p & \Delta = 0, \\ (p + \sqrt{\Delta}, p - \sqrt{\Delta}) & \Delta > 0, \end{cases} \quad (1.1)$$

gdzie $\Delta = p^2 - q$.

Przykład 1.2 (*Układ równań liniowych.*) Rozpatrzmy zadanie rozwiązywania układu równań liniowych

$$A\vec{x} = \vec{b},$$

dla nieosobliwej macierzy $A = (a_{i,j})_{i,j=1}^n$ i wektora $\vec{b} = (b_j)_{j=1}^n$. Wtedy

$$F = \{ (A, \vec{b}) \in \mathbb{R}^{n^2+n} : \det A \neq 0 \},$$

$G = \mathbb{R}^n$, oraz

$$S(A, \vec{b}) = A^{-1}\vec{b}.$$

Przykład 1.3 (*Całkowanie.*) Dla danej funkcji $f : [a, b] \rightarrow \mathbb{R}$ chcemy obliczyć jej całkę oznaczoną

$$\text{Int}(f) = \int_a^b f(x) dx.$$

W tym przypadku $G = \mathbb{R}$, a dane F to pewna klasa funkcji całkownych określonych na odcinku $[a, b]$. Oczywiście, S jest operatorem całkowania, $S = \text{Int}$.

Przykład 1.4 (*Aproksymacja funkcji.*) Załóżmy, że chcemy aproksymować ‘skomplikowaną’ funkcję $f : [a, b] \rightarrow \mathbb{R}$ z pewnej klasy F funkcją ‘prostszą’, np. wielomianem algebraicznym w stopnia co najwyżej n . Oczywiście, wielomian jako taki nie może być wynikiem obliczeń, ale możemy spróbować obliczyć jego współczynniki w wybranej bazie, np. w bazie potęgowej,

$$w(x) = \sum_{i=0}^n a_i x^i.$$

Wtedy $G = \mathbb{R}^{n+1}$, a operator rozwiązania S definiuje jednoznacznie zależność pomiędzy f i w poprzez współczynniki a_i , tzn.

$$S(f) = (a_0, a_1, \dots, a_n).$$

1.2 Model obliczeniowy

Aby zdefiniować nasz model obliczeniowy, posłużymy się pojęciem *programu*. Zastosujemy przy tym notację podobną do tej z języka programowania *Pascal*.

Program składa się z *deklaracji*, czyli opisu obiektów, których będziemy używać, oraz z *poleceń* (*instrukcji*), czyli opisu akcji, które będziemy wykonywać.

Dostępными obiektami są *stałe* i *zmienne* typu całkowitego (*integer*), rzeczywistego (*real*), albo logicznego (*Boolean*). Zmienne mogą być grupowane w wektory albo tablice.

Polecenia dzielimy na proste i złożone. Poleceniem prostym jest

- PODSTAWIENIE

$$z := \mathcal{W};$$

gdzie z jest zmienną, a \mathcal{W} jest *wyrażeniem* o wartościach tego samego typu co z .

Wyrażeniem jest pojedyncza stała lub zmienna, albo złożenie skończonej liczby *operacji elementarnych* na wyrażeniach. Operacje elementarne to:

arytmetyczno–arytmetyczne: $x \mapsto -x$, $(x, y) \mapsto x + y$, $(x, y) \mapsto x - y$, $(x, y) \mapsto x * y$, $(x, y) \mapsto x/y$, $y \neq 0$, gdzie x, y są stałymi lub zmiennymi liczbowymi,

arytmetyczno–logiczne: $(x, y) \mapsto x < y$, $(x, y) \mapsto x \leq y$, $(x, y) \mapsto x = y$, $(x, y) \mapsto x \neq y$, gdzie x, y są stałymi lub zmiennymi liczbowymi,

logiczno–logiczne: $p \mapsto \text{not } p$, $(p, q) \mapsto p \text{ and } q$, $(p, q) \mapsto p \text{ or } q$, gdzie p, q są stałymi lub zmiennymi logicznymi.

Dla niektórych zadań wygodnie jest (a czasem konieczne) uzupełnić ten zbiór o dodatkowe operacje, takie jak obliczanie wartości niektórych funkcji standardowych ($\sqrt{\quad}$, $\cos()$, $\sin()$, $\exp()$, $\log()$, itp.), czy nawet funkcji bardziej skomplikowanych. Na przykład, zastosowanie “szkolnych” wzorów na obliczanie pierwiatków równania kwadratowego z Przykładu 1.1 byłoby niemożliwe, gdyby pierwiastkowanie było niemożliwe. Należy jednak przy tym pamiętać, że w praktyce funkcje standardowe (o ile są dopuszczalne) są obliczane używając czterech podstawowych operacji arytmetycznych.

Mamy trzy podstawowe polecenia złożone.

- WARUNKOWE

if \mathcal{W} **then** \mathcal{A}_1 **else** \mathcal{A}_2 ;

gdzie \mathcal{W} jest wyrażeniem o wartościach logicznych, a \mathcal{A}_1 i \mathcal{A}_2 są poleceniami, przy czym dopuszczamy polecenia puste.

- POWTARZANE

while \mathcal{W} **do** \mathcal{A} ;

gdzie \mathcal{W} jest wyrażeniem o wartościach logicznych, a \mathcal{A} jest poleceniem.

- KOMBINOWANE

begin \mathcal{A}_1 ; \mathcal{A}_2 ; \dots ; \mathcal{A}_n **end**;

gdzie \mathcal{A}_j są poleceniami.

Na podstawie tych trzech poleceń można tworzyć inne, takie jak pętle **for** i **repeat**, czy **case**, itd.

Mamy też dwa szczególne polecenia, które odpowiadają za “wejście” i “wyjście”.

- WPROWADZANIE DANYCH

$\mathcal{IN}(x, t)$;

gdzie x jest zmienną rzeczywistą, a t “adresem” pewnego funkcjonału $L_t : F \rightarrow \mathbb{R}$ należącym do pewnego zbioru T . W wyniku wykonania tego polecenia w x zostaje umieszczona wartość $L_t(f)$.

Polecenie to pozwala zdobyć *informację* o danej f . Jeśli $F = \mathbb{R}^n$ to zwykle mamy $T = \{1, 2, \dots, n\}$ i $L_i(f) = f_i$, co w praktyce odpowiada wczytaniu i -tej współrzędnej wektora danych. W szczególności, ciąg poleceń $\mathcal{IN}(x[i], i)$, $i = 1, 2, \dots, n$, pozwala uzyskać pełną informację o f . Jeśli zaś F jest pewną klasą funkcji $f : [a, b] \rightarrow \mathbb{R}$, to możemy mieć np. $T = [a, b]$ i $L_t(f) = f(t)$. W tym przypadku, wykonanie polecenia $\mathcal{IN}(x, t)$ odpowiada w praktyce skorzystaniu ze specjalnej procedury (albo urządzenia zewnętrznego) obliczającej (mierzącego) wartość funkcji f w punkcie t .

- WYPROWADZANIE WYNIKÓW

$\mathcal{OUT}(\mathcal{W})$;

gdzie \mathcal{W} jest wyrażeniem o wartościach rzeczywistych. Polecenie to pozwala “wskazać” kolejną współrzędną wyniku.

Zakładamy, że na początku procesu obliczeniowego wartości wszystkich zadeklarowanych zmiennych są nieokreślone, oraz że dla dowolnych danych wykonanie programu wymaga wykonania skończonej liczby operacji elementarnych. Wynikiem obliczeń jest \emptyset albo skończony ciąg liczb rzeczywistych, którego kolejne współrzędne pokazywane są poleceniem *OUT*.

1.3 Arytmetyka zmiennoprzecinkowa

Przedstawiony model obliczeniowy jest modelem idealistycznym, tzn. zakłada on, że wszystkie operacje są wykonywane bezbłędnie. Dlatego w tym przypadku będziemy mówić o *arytmetyce idealnej*. W praktyce jednak, np. wykonując obliczenia na maszynie cyfrowej, operacje arytmetyczne na liczbach rzeczywistych wykonywane są z pewnymi błędami. Matematycznym modelem arytmetyki maszyny cyfrowej jest *arytmetyka fl_ν* , czyli arytmetyka *zmiennoprzecinkowa* (ang. *floating point arithmetic*), którą teraz przedstawimy. Posłużymy się przy tym dla uproszczenia pozycyjnym systemem dziesiętnym, chociaż trzeba pamiętać o tym, że we współczesnych systemach obliczeniowych dominuje system dwójkowy.

Dowolną liczbę rzeczywistą $x \neq 0$ można zapisać w postaci

$$x = s \cdot 10^c \cdot m,$$

gdzie $s \in \{-1, 1\}$ jest znakiem, $c \in \mathbb{Z}$ *cechą*, a $m \in [0.1, 1.0)$ *mantysą* liczby x ,

$$m = \sum_{j=1}^{\infty} c_j 10^{-j},$$

gdzie $c_j \in \{0, 1, 2, \dots, 9\}$ i $c_1 \neq 0$. Zauważmy, że taki rozkład jest jednoznaczny (jeśli nie dopuszczamy rozwinięć w których $c_j = 9$ dla wszystkich dostatecznie dużych indeksów j) i odpowiada przesuwaniu przecinka w rozwinięciu dziesiętnym liczby do pierwszej cyfry znaczącej, tj. różnej od zera. Mantysa ma w ogólności nieskończenie wiele cyfr c_j w swoim rozwinięciu dziesiętnym i jako taka nie może być zapamiętana dokładnie w pamięci maszyny cyfrowej. Zakładając, że pamiętamy tylko t cyfr znaczących i ostatnią cyfrę zaokrąglamy, zamiast m pamiętamy

$$m_t = \sum_{j=1}^t c_j 10^{-j} + d_t 10^{-j},$$

gdzie

$$d_t = \begin{cases} 0 & \text{gdy } 0 \leq c_{t+1} \leq 4, \\ 1 & \text{gdy } 5 \leq c_{t+1} \leq 9. \end{cases}$$

Stąd x będzie reprezentowana przez liczbę

$$rd(x) = s \cdot 10^c \cdot m_t.$$

Między liczbą x ($x \neq 0$) a jej reprezentacją $rd(x)$ zachodzi więc nierówność

$$\frac{|x - rd(x)|}{|x|} = \frac{|m - m_t|}{|m|} \leq \frac{5 \cdot 10^{-(t+1)}}{10^{-1}} = 5 \cdot 10^{-t},$$

charakteryzująca dokładność arytmetyki. Liczby rzeczywiste są więc w arytmetyce fl_ν reprezentowane z dokładnością *względną* nie większą niż

$$\nu = 5 \cdot 10^{-t}.$$

(We współczesnych systemach obliczeniowych ν jest zwykle rzędu 10^{-16} .) Przyjmując dodatkowo, że zero reprezentowane jest przez zero, zależność pomiędzy dowolną liczbą rzeczywistą x a jej reprezentacją $rd(x)$ możemy zapisać w równoważny sposób jako

$$rd(x) = x(1 + \varepsilon), \quad \text{gdzie } |\varepsilon| \leq \nu.$$

W arytmetyce fl_ν postulujemy ponadto, że działania arytmetyczne na reprezentacjach liczb rzeczywistych są wykonywane dokładnie i tylko wynik jest zaokrąglany. Mamy więc

$$fl_\nu(x \square y) = rd(rd(x) \square rd(y)),$$

gdzie $\square \in \{+, -, *, /\}$. Ogólniej, jeśli \mathcal{W}_1 i \mathcal{W}_2 są wyrażeniami o wartościach rzeczywistych, to dla dowolnych wartości zmiennych wynik w fl_ν działania $\mathcal{W}_1 \square \mathcal{W}_2$ wyraża się wzorem

$$fl_\nu(\mathcal{W}_1 \square \mathcal{W}_2) = rd(fl_\nu(\mathcal{W}_1) \square fl_\nu(\mathcal{W}_2)).$$

(Oczywiście, dla $x \in \mathbb{R}$ mamy $fl_\nu(x) = rd(x)$.)

Zwykle dla prostoty będziemy również zakładać podobną zależność dla niektórych funkcji standardowych, o ile należą one do zbioru operacji dopuszczalnych (w rzeczywistości są one obliczane przez procedury używające czterech podstawowych operacji arytmetycznych). I tak będziemy mieć, np.

$$\begin{aligned} fl_\nu(\sqrt{\mathcal{W}}) &= \left(\sqrt{fl_\nu(\mathcal{W})}\right)(1 + \beta_1), \\ fl_\nu(\cos(\mathcal{W})) &= (\cos(fl_\nu(\mathcal{W}))) (1 + \beta_2), \end{aligned}$$

gdzie $|\varepsilon_j| \leq \nu$, oraz $\beta_j \leq K_j \nu$ i K_j są “niewielkimi” stałymi.

Podobnie, jeśli Δ jest operatorem porównania, $\Delta \in \{<, \leq, =, \neq\}$, to wartością wyrażenia logicznego $\mathcal{W}_1 \Delta \mathcal{W}_2$ w fl_ν jest dokładna wartość wyrażenia $fl_\nu(\mathcal{W}_1) \Delta fl_\nu(\mathcal{W}_2)$.

Uwagi i uzupełnienia

U. 1.1 W maszynie cyfrowej cecha c liczby rzeczywistej nie może oczywiście mieć dowolnie dużej wartości bezwzględnej, $|c| \leq c_{\max}$. Powoduje to powstanie zjawiska *nadmiaru* gdy $c > c_{\max}$, oraz zjawiska *niedomiaru* gdy $c < -c_{\max}$. W pierwszym przypadku liczba jest tak duża (co do modułu), że nie zawiera się w przedziale liczb reprezentowalnych, a w drugim jest tak mała, że tracimy względną dokładność ν jej reprezentacji. (Np. jeśli taka liczba jest reprezentowana przez zero, to błąd względny reprezentacji wynosi 1, a nie ν .) W przedstawionym modelu i w dalszych rozważaniach zjawiska nadmiaru i niedomiaru będziemy dla uproszczenia zaniedbywać.

U. 1.2 Ważną konsekwencją stosowania arytmetyki zmiennopozycyjnej jest możliwość wystąpienia niebezpiecznego zjawiska *redukcji cyfr znaczących*, które powoduje nagłą utratę dokładności względnej. Rozpatrzmy dla przykładu arytmetykę fl_ν z czterema cyframi znaczącymi, tzn. z $\nu = 5 \cdot 10^{-4}$. Wtedy dla liczb $x = 1.2345$, $y = 1.234499$ mamy $rd(x) = 1.235$, $rd(y) = 1.234$, a stąd błąd względny odejmowania tych liczb wynosi

$$\frac{|fl_\nu(x - y) - (x - y)|}{|x - y|} = \frac{10^{-3} - 10^{-6}}{10^{-6}} = 999.$$

Widzimy, że nastąpiła istotna utrata dokładności względnej wyniku w porównaniu z dokładnością reprezentacji. Dzieje się tak dlatego, że w wyniku odejmowania liczb ‘podobnych’, takich jak x i y , początkowe cyfry znaczące tych liczb ulegają redukcji. Dla podniesienia dokładności potrzebowalibyśmy dalszych cyfr rozwinięć

x i y , które straciliśmy w wyniku reprezentacji w fl_ν . Dodajmy, że zjawisko to nie wystąpi przy dodawania tych samych liczb. Rzeczywiście,

$$\frac{|fl_\nu(x+y) - (x+y)|}{|x+y|} = \frac{2.469 - 2.468999}{2.468999} \cong 4 \cdot 10^{-7}.$$

Podobnie, nie pojawi się ono nigdy przy operacjach mnożenia i dzielenia.

Redukcji cyfr znaczących można często uniknąć, np. poprzez modyfikację sposobu obliczenia danego wyrażenia, np. zob. Ćw. 1.4.

U. 1.3 W następnych rozdziałach często będziemy się posługiwać normami wektorów $\vec{x} = (x_j)_{j=1}^n \in \mathbb{R}^n$ i macierzy $A = (a_{i,j})_{i,j} \in \mathbb{R}^{m \times n}$, a w szczególności błędami reprezentacji wektorów i macierzy w normie. Najczęściej używanymi normami wektorowymi będą normy p -te,

$$\|\vec{x}\| = \|\vec{x}\|_p = \left(\sum_{j=1}^n |x_j|^p \right)^{1/p}, \quad 1 \leq p < +\infty,$$

oraz

$$\|\vec{x}\|_\infty = \lim_{p \rightarrow +\infty} \|\vec{x}\|_p = \max_{1 \leq j \leq n} |x_j|.$$

Normą macierzową jest norma euklidesowa (albo Frobeniusa)

$$\|A\|_E = \sqrt{\sum_{i,j=1}^n |a_{i,j}|^2},$$

a także normy *indukowane* przez normy wektorowe (np. przez normy p -te)

$$\|A\| = \sup_{\vec{x} \neq 0} \frac{\|A\vec{x}\|}{\|\vec{x}\|} = \sup_{\|\vec{x}\|=1} \|A\vec{x}\|.$$

Jeśli norma macierzowa jest indukowana przez normę wektorową, to dla dowolnego wektora mamy

$$\|A\vec{x}\| \leq \|A\| \|\vec{x}\|.$$

U. 1.4 Przypomnijmy, że w przestrzeniach liniowych skończenie wymiarowych (a więc także w \mathbb{R}^n i w przestrzeni macierzy formatu $m \times n$) każde dwie normy są równoważne. To znaczy, że jeśli mamy dwie normy $\|\cdot\|$ i $\|\cdot\|'$ w przestrzeni skończenie wymiarowej X , to istnieją stałe $0 < K_1 \leq K_2 < +\infty$ takie, że

$$K_1 \|x\| \leq \|x\|' \leq K_2 \|x\| \quad \text{dla wszystkich } x \in X.$$

W szczególności, dla $\vec{x} \in \mathbb{R}^n$ mamy

$$\|\vec{x}\|_\infty \leq \|\vec{x}\|_1 \leq n \|\vec{x}\|_\infty, \quad (1.2)$$

$$\|\vec{x}\|_\infty \leq \|\vec{x}\|_2 \leq \sqrt{n} \|\vec{x}\|_\infty, \quad (1.3)$$

$$\|\vec{x}\|_2 \leq \|\vec{x}\|_1 \leq \sqrt{n} \|\vec{x}\|_2, \quad (1.4)$$

a dla $A \in \mathbb{R}^{m \times n}$ mamy

$$\|A\|_2 \leq \| |A| \|_2 \leq \|A\|_E \leq \sqrt{\min(m,n)} \|A\|_2, \quad (1.5)$$

gdzie $|A| = (|a_{i,j}|)_{i,j}$.

Ćwiczenia

Ćw. 1.1 Podaj przykłady liczb x i y , które są dokładnie reprezentowane w fl_ν , ale $fl_\nu(x \square y) \neq x \square y$, gdzie $\square \in \{+, -, *, /\}$.

Ćw. 1.2 Uzasadnij, że jeśli $x \leq y$ to $rd(x) \leq rd(y)$ oraz $fl_\nu(x/y) \leq 1$. Podaj przykład wyrażenia \mathcal{W} takiego, że dla pewnych wartości zmiennych $\mathcal{W} < 0$, ale $fl_\nu(\mathcal{W}) > 0$.

Ćw. 1.3 Jak zabezpieczyć się przed powstaniem nadmiaru albo niedomiaru przy obliczaniu wyrażenia $\sqrt{x^2 + y^2}$, gdy x i y leżą w przedziale liczb reprezentowalnych w fl_ν , ale x^2 lub y^2 już nie?

Ćw. 1.4 Jak obliczyć w fl_ν wartość wyrażenia $\sqrt{x^2 + 1} - \sqrt{x^2 - 1}$ aby uniknąć redukcji cyfr znaczących dla ‘dużych’ wartości x ?

Ćw. 1.5 Jak policzyć w fl_ν wartość funkcji $1 - \cos(4x)$ dla x na poziomie dokładności arytmetyki, aby otrzymać mały błąd względny? (Zakładamy, że funkcje trygonometryczne potrafimy liczyć z błędem względnym na poziomie błędu reprezentacji.)

Ćw. 1.6 Uzasadnij nierówności (1.2)–(1.5).

Ćw. 1.7 Pokaż, że dla macierzy $A = (a_{i,j})_{i,j} \in \mathbb{R}^{m \times n}$ mamy

$$\|A\|_\infty = \max_{1 \leq i \leq m} \sum_{j=1}^n |a_{i,j}| \quad \text{oraz} \quad \|A\|_1 = \|A^T\|_\infty = \max_{1 \leq j \leq n} \sum_{i=1}^m |a_{i,j}|.$$

Ćw. 1.8 Pokaż, że normę drugą dowolnej macierzy można wyrazić jako

$$\|A\|_2 = \max_{\lambda \in \text{Sp}(A^T A)} \sqrt{\lambda},$$

gdzie $\text{Sp}(B)$ jest zbiorem wszystkich wartości własnych macierzy B .

Wskazówka. Wykorzystaj fakt z algebry liniowej, że jeśli macierz $B \in \mathbb{R}^{n \times n}$ spełnia $B = B^T$ oraz $\vec{x}^T B \vec{x} \geq 0$ dla wszystkich $\vec{x} \in \mathbb{R}^n$, to istnieje w \mathbb{R}^n baza ortonormalna wektorów własnych macierzy B , a odpowiadające im wartości własne są rzeczywiste i nieujemne.

Ćw. 1.9 Wykaż, że dla dowolnej macierzy $A \in \mathbb{R}^{m \times n}$,

$$\|A\|_2 = \sup_{\|\vec{x}\|_2=1} \sup_{\|\vec{y}\|_2=1} \vec{y}^T A \vec{x}.$$

Wywnioskuj stąd, że $\|A^T\|_2 = \|A\|_2$.

Ćw. 1.10 Pokaż, że macierze $A^T A$ i $A A^T$ mają takie same niezerowe wartości własne, a podprzestrzenie własne im odpowiadające mają ten sam wymiar. Wywnioskuj stąd, że $\|A^T\|_2 = \|A\|_2$.

Ćw. 1.11 Macierz A dana jest w postaci blokowej

$$A = \begin{pmatrix} A_{1,1} & A_{1,2} & \cdots & A_{1,n} \\ A_{2,1} & A_{2,2} & \cdots & A_{2,n} \\ \vdots & \vdots & & \vdots \\ A_{m,1} & A_{m,2} & \cdots & A_{m,n} \end{pmatrix}.$$

Wykaż, że dla dowolnych i, j mamy $\|A_{i,j}\|_p \leq \|A\|_p$ dla $1 \leq p \leq +\infty$.

Ćw. 1.12 Wykaż nierówność $\|A\|_2^2 \leq \|A\|_1 \|A\|_\infty$.

Ćw. 1.13 Dla wektora $\vec{x} = (x_j)_{j=1}^n \in \mathbb{R}^n$, niech $rd(\vec{x}) = (rd(x_j))_{j=1}^n$. Pokaż, że

$$\|\vec{x} - rd(\vec{x})\|_p \leq \nu \|\vec{x}\|_p$$

dla $1 \leq p \leq +\infty$.

Ćw. 1.14 Dla macierzy $A = (a_{i,j})_{i,j} \in \mathbb{R}^{m \times n}$, niech $rd(A) = (rd(a_{i,j}))_{i,j}$. Pokaż, że

$$\|A - rd(A)\|_p \leq \nu \|A\|_p,$$

dla $p = 1, \infty$, oraz

$$\|A - rd(A)\|_2 \leq \|A - rd(A)\|_E \leq \nu \|A\|_E \leq \sqrt{\min(m, n)} \nu \|A\|_2.$$

Rozdział 2

Algorytmy i ich własności

Algorytm to dokładnie określona i dozwolona w danym modelu obliczeniowym sekwencja akcji, pozwalająca na rozwiązanie danego zadania (w sposób dokładny lub przybliżony).

Z każdym algorytmem związany jest operator

$$\Phi : F \rightarrow G,$$

taki że $\Phi(f)$ jest wynikiem działania algorytmu w arytmetyce idealnej dla danej f .

Zauważmy, że wynik $\Phi(f)$ działania algorytmu nie zależy bezpośrednio od f , ale raczej od *informacji* o f (uzyskanej dzięki poleceniu \mathcal{IN}). Informacja ta może być *pełna* albo tylko *częściowa*. Informacja jest pełna gdy, np. $f = (f_1, \dots, f_n) \in \mathbb{R}^n$ i wczytamy wszystkie współrzędne f_i . Informacja może być częściowa, gdy f jest funkcją. Wtedy wiele danych może posiadać tę samą informację, co łatwo zaobserwować na przykładzie zadań całkowania czy aproksymacji funkcji.

Niech $N : F \rightarrow \cup_{n=0}^{\infty} \mathbb{R}^n$ będzie *operatorem informacji*, tzn.

$$N(f) = (y_1, y_2, \dots, y_n)$$

jest informacją o f zebraną przy idealnej realizacji algorytmu. Informacja jest pełna gdy N jest przekształceniem różnowartościowym, tzn. jeśli $f_1 \neq f_2$ implikuje $N(f_1) \neq N(f_2)$. W przeciwnym przypadku mamy do czynienia z informacją częściową.

Każdy algorytm Φ może być przedstawiony jako złożenie operatora informacji i pewnego operatora $\varphi : N(F) \rightarrow G$ zdefiniowanego równością

$$\Phi(f) = \varphi(N(f)).$$

Zauważmy, że w przypadku informacji częściowej zwykle nie istnieje algorytm dający dokładne rozwiązanie zadania dla każdej danej $f \in F$, ponieważ dla danych o tej samej informacji mogą istnieć różne rozwiązania.

Wybór algorytmu jest najistotniejszą częścią całego procesu numerycznego rozwiązywania zadania. Kierujemy się przy tym przede wszystkim następującymi kryteriami:

- dokładnością algorytmu,
- złożonością algorytmu,
- własnościami numerycznymi algorytmu.

Przez dokładność algorytmu rozumiemy różnicę między rozwiązaniem dokładnym $S(f)$, a rozwiązaniem $\Phi(f)$ dawanym przez algorytm w arytmetyce idealnej. Jeśli $\Phi(f) = S(f), \forall f \in F$, to algorytm nazywamy *dokładnym*.

Mówiąc o złożoności, mamy na myśli złożoność pamięciową (zwykle jest to liczba stałych i zmiennych używanych przez algorytm), jak również złożoność obliczeniową. Na złożoność obliczeniową algorytmu dla danej f składa się koszt uzyskania informacji $y = N(f)$ (zwykle jest on proporcjonalny do liczby wywołań polecenia \mathcal{IN}), oraz koszt *kombinatoryczny* przetworzenia tej informacji, aż do uzyskania wyniku $\varphi(y)$. Koszt kombinatoryczny zwykle mierzymy liczbą operacji arytmetycznych i/lub logicznych wykonywanych przez algorytm.

Przez własności numeryczne algorytmu rozumiemy jego własności przy realizacji w arytmetyce fl_ν . Temu ważnemu tematowi poświęcimy teraz osobną sekcję.

2.1 Numeryczna poprawność algorytmu

Pożądane jest, aby algorytm dawał ‘dobry’ wynik zarówno w arytmetyce idealnej, jak i w arytmetyce fl_ν . Niestety, jak zobaczymy, nie zawsze jest to możliwe. Nawet jeśli algorytm jest dokładny to w wyniku jego realizacji w fl_ν możemy otrzymać wynik daleko odbiegający od $S(f)$. Zauważmy również, że o ile z reguły znamy dokładne zachowanie się algorytmu w arytmetyce idealnej dla danej informacji, to nie można tego samego powiedzieć o jego zachowaniu w arytmetyce fl_ν . W związku z tym powstaje pytanie, jak kontrolować błąd algorytmów wynikający z błędów zaokrągleń i jakie algorytmy uznamy za te o najwyższej jakości numerycznej.

Istnienie błędów reprezentacji liczb rzeczywistych powoduje, że informacja $y = N(f)$ o danej f nie jest w ogólności reprezentowana dokładnie. Znaczy to, że zamiast na informacji dokładnej, dowolny algorytm będzie operować na informacji *nieco zaburzonej* y_ν , tzn. zaburzonej na poziomie błędu reprezentacji. Tak samo wynik dawany przez algorytm będzie w ogólności zaburzony na poziomie błędu reprezentacji. W najlepszym więc wypadku wynikiem działania algorytmu w fl_ν będzie $(\varphi(y_\nu))_\nu$ zamiast $\varphi(y)$. Algorytmy dające tego rodzaju wyniki uznamy za posiadające najlepsze własności numeryczne w arytmetyce fl_ν i nazwiemy *numerycznie poprawnymi*.

Dokładniej, powiemy, że ciąg rzeczywisty $a_\nu = (a_{\nu,1}, \dots, a_{\nu,n})$ (a właściwie rodzina ciągów $\{a_\nu\}_\nu$) jest *nieco zaburzonym* ciągiem $a = (a_1, \dots, a_n)$, jeśli istnieje stała K taka, że dla wszystkich dostatecznie małych ν zachodzi

$$|a_{\nu,j} - a_j| \leq K \nu |a_j|, \quad 1 \leq j \leq n, \quad (2.1)$$

albo ogólniej

$$\|a_\nu - a\| \leq K \nu \|a\|, \quad (2.2)$$

gdzie $\|\cdot\|$ jest pewną normą w \mathbb{R}^n . W pierwszym przypadku mówimy o zaburzeniu “po współrzędnych”, a w drugim o zaburzeniu w normie $\|\cdot\|$.

Zauważmy, że niewielkie zaburzenia po współrzędnych pociągają za sobą niewielkie zaburzenia w normie. Rzeczywiście, jeśli (2.1) to również

$$\|a_\nu - a\|_\infty = \max_{1 \leq j \leq n} |a_{\nu,j} - a_j| \leq K \nu \max_{1 \leq j \leq n} |a_j| = K \nu \|a\|_\infty,$$

i korzystając z faktu, że w przestrzeni skończonej wymiarowej wszystkie normy są równoważne otrzymujemy dla pewnych stałych K_1 i K_2

$$\|a_\nu - a\| \leq K_1 \|a_\nu - a\|_\infty \leq K_1 K \nu \|a\|_\infty \leq K_2 K_1 K \nu \|a\|,$$

czyli nierówność (2.2) ze stałą K_2K_1K zamiast K .

Dalej dla uproszczenia zapisu przyjmujemy, że Φ oznacza zarówno operator stowarzyszony z algorytmem, ale również sam algorytm, czyli sekwencję akcji. W szczególności $fl_\nu(\Phi(f))$ jest wynikiem otrzymanym w fl_ν dla danej f .

Definicja 2.1 Algorytm Φ nazywamy *numerycznie poprawnym* w zbiorze danych $F_0 \subset F$, jeśli dla każdej danej $f \in F_0$ wynik $fl_\nu(\Phi(f))$ działania algorytmu w arytmetyce fl_ν można zinterpretować jako nieco zaburzony wynik algorytmu w arytmetyce idealnej dla nieco zaburzonej informacji $y_\nu = (N(f))_\nu \in N(F)$, przy czym poziom zaburzeń nie zależy od f .

Formalnie znaczy to, że istnieją stałe K_1, K_2 , oraz $\nu_0 > 0$ takie, że spełniony jest następujący warunek. Dla dowolnej $\nu \leq \nu_0$ oraz informacji $y \in N(F_0)$ można dobrać $y_\nu \in N(F)$ oraz $(\varphi(y_\nu))_\nu \in G$ takie, że

$$\|y_\nu - y\| \leq K_1 \nu \|y\|,$$

$$\|(\varphi(y_\nu))_\nu - \varphi(y_\nu)\| \leq K_2 \nu \|\varphi(y_\nu)\|,$$

oraz

$$fl_\nu(\Phi(f)) = fl_\nu(\varphi(N(f))) = (\varphi(y_\nu))_\nu.$$

□

Zauważmy, że jeśli $f \in \mathbb{R}^n$, $N(f) = (f_1, \dots, f_n)$, oraz algorytm jest dokładny, tzn. $\varphi = \Phi = S$, to numeryczną poprawność algorytmu można równoważnie zapisać jako

$$fl_\nu(\Phi(f)) = (S(f_\nu))_\nu.$$

2.2 Rola uwarunkowania zadania

Niech $\Phi(\cdot) = \varphi(N(\cdot))$ będzie algorytmem numerycznie poprawnym dla danych $F_0 \subset F$. Wtedy jego błąd w fl_ν można oszacować następująco:

$$\begin{aligned} \|S(f) - fl_\nu(\Phi(f))\| &= \|S(f) - (\varphi(y_\nu))_\nu\| \\ &\leq \|S(f) - \varphi(y)\| + \|\varphi(y) - \varphi(y_\nu)\| + \|\varphi(y_\nu) - (\varphi(y_\nu))_\nu\| \\ &\leq \|S(f) - \Phi(f)\| + \|\varphi(y) - \varphi(y_\nu)\| + K_2 \nu \|\varphi(y_\nu)\| \\ &\leq \|S(f) - \Phi(f)\| + (1 + K_2 \nu) \|\varphi(y) - \varphi(y_\nu)\| + K_2 \nu \|\varphi(y)\|, \end{aligned}$$

przy czym $\|y_\nu - y\| \leq K_1 \nu \|y\|$. Stąd w szczególności wynika, że jeśli algorytm jest numerycznie poprawny i ciągły ze względu na informację y , to

$$\lim_{\nu \rightarrow 0} \|S(f) - fl_\nu(\Phi(f))\| = \|S(f) - \Phi(f)\|.$$

To zaś oznacza, że dla dostatecznie silnej arytmetyki algorytm będzie się zachowywał w fl_ν prawie tak jak w arytmetyce idealnej.

Z powyższych wzorów wynika, że błąd w fl_ν algorytmu numerycznie poprawnego zależy w dużym stopniu od:

- dokładności algorytmu w arytmetyce idealnej,

- dokładności ν arytmetyki f_{ν} ,
- wrażliwości algorytmu na małe względne zaburzenia informacji y .

Ponieważ dwa pierwsze punkty są raczej oczywiste, poświęcimy trochę więcej uwagi jedynie trzeciemu z nich.

Jeśli φ spełnia warunek Lipschitza ze stałą L , tzn. $\|\varphi(y_{\nu}) - \varphi(y)\| \leq L \|y_{\nu} - y\|$, to

$$\begin{aligned} \|S(f) - f_{\nu}(\Phi(f))\| &\leq \|S(f) - \Phi(f)\| + (1 + K_2\nu)L\|y_{\nu} - y\| + K_2\nu\|\varphi(y)\| \\ &\leq \|S(f) - \Phi(f)\| + (1 + K_2\nu)LK_1\nu\|y\| + K_2\nu\|\varphi(y)\|. \end{aligned}$$

W tym przypadku, błędy zaokrążeń zwiększają błąd bezwzględny algorytmu proporcjonalnie do ν . Bardziej jednak interesuje nas błąd *względny*. Przypuśćmy, że

$$\|\varphi(y_{\nu}) - \varphi(y)\| \leq M K_1 \nu \max(\eta, \|\varphi(y)\|),$$

dla pewnych M i "małego" η , przy czym η może zależeć od ν . To znaczy, że błąd względny informacji, $\|y_{\nu} - y\| \leq K_1\nu\|y\|$, przenosi się na błąd względny wyniku (w arytmetyce idealnej) ze "współczynnikiem wzmocnienia" M , albo na błąd bezwzględny ze współczynnikiem $M\eta$. (Najchętniej przyjęlibyśmy $\eta = 0$, co jak się przekonamy jest często możliwe dla zadań liniowych S . Jednak w ogólności oznaczałoby to, że dla y takiej, że $\varphi(y) = 0$ musiałyby być $\varphi(y_{\nu}) = 0$, co dla zadań nieliniowych zwykle nie jest prawdą.) Wtedy

$$\begin{aligned} \|S(f) - f_{\nu}(\Phi(f))\| &\leq \|S(f) - \Phi(f)\| + (1 + K_2\nu)MK_1\nu \max(\eta, \|\varphi(y)\|) + K_2\nu\|\varphi(y)\| \\ &= \|S(f) - \Phi(f)\| + \nu \left(MK_1(1 + K_2\nu) + K_2 \right) \max(\eta, \|\varphi(y)\|). \end{aligned}$$

Ta nierówność pokazuje jak M , czyli wrażliwość algorytmu numerycznie poprawnego na względne zaburzenia informacji, przenosi się na błąd algorytmu w f_{ν} .

Załóżmy teraz, że algorytm jest dokładny, tzn. $\Phi = S$, oraz korzysta z pełnej informacji o f , tzn. $\varphi = \Phi$. (To zwykle zachodzi np. w rozwiązywaniu skończonych układów liniowych). Wtedy zamiast o wrażliwości odwzorowania φ na zaburzenia informacji y możemy równoważnie mówić o wrażliwości samego zadania S na zaburzenia danych f . W tym przypadku mamy

$$\frac{\|S(f) - f_{\nu}(\Phi(f))\|}{\max(\eta, \|S(f)\|)} \leq \left(MK_1(1 + K_2\nu) + K_2 \right) \nu \approx (M K_1 + K_2) \nu.$$

Jeśli $(MK_1 + K_2)\nu \ll 1$ to błąd względny w f_{ν} algorytmu numerycznie poprawnego jest mały, o ile $\|S(f)\| \geq \eta$. Błąd względny jest proporcjonalny do dokładności ν , współczynników proporcjonalności K_i oraz do wrażliwości M zadania S na małe względne zaburzenia danych. Zadania, które nie są zbyt wrażliwe na "małe" względne zaburzenia danych, tzn. dla których M jest "niewielkie", nazywamy ogólnie zadaniami *dobrze uwarunkowanymi*.

Zwróćmy uwagę na istotny fakt, że aby zbadać jakość wyniku algorytmu numerycznie poprawnego wystarczy zbadać jego wrażliwość na zaburzenia tylko tych danych (informacji), które powstają przy analizie zachowania się algorytmu w f_{ν} , zob. Przykład 2.1 albo U. 2.2.

2.3 Przykłady

Podamy teraz dwa proste przykłady zadań, które mogą być rozwiązane algorytmami numerycznie poprawnymi. Oszacujemy też błędy tych algorytmów. Dla uproszczenia rachunków, będziemy przy tym używać przybliżonej nierówności ' \lesssim ', której formalna definicja znajduje się w U. 2.1.

Przykład 2.1 (Iloczyn skalarny.)

Załóżmy, że dla danych ciągów rzeczywistych $a = (a_1, a_2, \dots, a_n)$ i $b = (b_1, b_2, \dots, b_n)$ chcemy obliczyć

$$S(a, b) = \sum_{j=1}^n a_j b_j.$$

Rozpatrzmy algorytm dokładny zdefiniowany powyższym wzorem i korzystający z pełnej informacji o kolejnych współrzędnych.

Oznaczmy przez \tilde{a}_j i \tilde{b}_j reprezentacje liczb a_j i b_j w fl_ν , $\tilde{a}_j = a_j(1 + \alpha_j)$, $\tilde{b}_j = b_j(1 + \beta_j)$, oraz przez γ_j i δ_j błędy względne powstałe przy kolejnych mnożeniach i dodawaniach. Oczywiście $|\alpha_j|, |\beta_j|, |\gamma_j|, |\delta_j| \leq \nu$. Otrzymujemy

$$\begin{aligned} fl_\nu\left(\sum_{j=1}^n a_j b_j\right) &= \left(fl_\nu\left(\sum_{j=1}^{n-1} a_j b_j\right) + \tilde{a}_n \tilde{b}_n (1 + \gamma_n)\right) (1 + \delta_n) = \dots \\ &= \left(\dots \left(\tilde{a}_1 \tilde{b}_1 (1 + \gamma_1) + \tilde{a}_2 \tilde{b}_2 (1 + \gamma_2)\right) (1 + \delta_2) + \dots + \tilde{a}_n \tilde{b}_n (1 + \gamma_n)\right) (1 + \delta_n) \\ &= \tilde{a}_1 \tilde{b}_1 (1 + \gamma_1) (1 + \delta_2) \dots (1 + \delta_n) + \dots + \tilde{a}_j \tilde{b}_j (1 + \gamma_j) (1 + \delta_j) \dots (1 + \delta_n) \\ &= \sum_{j=1}^n a_j b_j (1 + e_j), \end{aligned}$$

gdzie w przybliżeniu (tzn. gdy $\nu \rightarrow 0$) mamy $|e_1| \lesssim (n+2)\nu$ i $|e_j| \lesssim (n-j+4)\nu$, $2 \leq j \leq n$. Algorytm naturalny jest więc numerycznie poprawny w całym zbiorze danych ciągów długości n , gdyż wynik otrzymany w fl_ν można zinterpretować jako dokładny wynik dla danych $a_{\nu,j} = a_j$ i $b_{\nu,j} = b_j(1 + e_j)$, przy czym $\|b_\nu - b\|_p \lesssim (n+2)\nu \|b\|_p$.

Zobaczmy teraz, jak błąd we współrzędnych b_j wpływa na błąd wyniku. Mamy

$$\begin{aligned} \left|\sum_{j=1}^n a_j b_j - fl_\nu\left(\sum_{j=1}^n a_j b_j\right)\right| &= \left|\sum_{j=1}^n a_j b_j - \sum_{j=1}^n a_j b_j (1 + e_j)\right| \\ &= \left|\sum_{j=1}^n e_j a_j b_j\right| \leq \sum_{j=1}^n |e_j| |a_j b_j| \lesssim (n+2)\nu \sum_{j=1}^n |a_j b_j|. \end{aligned}$$

Stąd (tu przyjmujemy $\eta = 0$)

$$\frac{|\sum_{j=1}^n a_j b_j - fl_\nu(\sum_{j=1}^n a_j b_j)|}{|\sum_{j=1}^n a_j b_j|} \lesssim M(n+2)\nu, \quad \text{gdzie} \quad M = \frac{\sum_{j=1}^n |a_j b_j|}{|\sum_{j=1}^n a_j b_j|}.$$

Zauważmy, że jeśli iloczyny $a_j b_j$ są wszystkie dodatnie albo wszystkie ujemne, to $M = 1$, tzn. zadanie jest dobrze uwarunkowane, a błąd względny jest zawsze na poziomie co najwyżej $n\nu$. W tym przypadku algorytm zachowuje się bardzo dobrze, o ile liczba n składników nie jest horrendalnie duża. W ogólności jednak M może być liczbą dowolnie dużą i wtedy nie możemy być pewni uzyskania dobrego wyniku w fl_ν .

Przykład 2.2 (Całkowanie.)

Zadanie całkowania z Przykładu 1.3 często rozwiązuje się (a raczej przybliża) stosując formułę

$$\Phi(f) = \varphi(y) = \sum_{j=1}^n c_j y_j,$$

gdzie informacja $y_j = f(t_j)$, t_j są tu ustalonymi punktami z $[a, b]$, a c_j ustalonymi współczynnikami rzeczywistymi, niezależnymi od f . Algorytmy korzystające z takich formuł nazywamy *kwadraturami*, a przykładem jest zwykła suma Riemanna,

$$\Phi_R(f) = \frac{1}{n} \sum_{j=1}^n f(t_j^*),$$

gdzie t_j^* należy do przedziału $[a + (j-1)h, a + jh]$, $1 \leq j \leq n$, gdzie $h = \frac{1}{n}(b-a)$.

Z analizy algorytmu obliczającego iloczyn skalarny wynika, że kwadratura jest algorytmem numerycznie poprawnym. Mamy bowiem

$$fl_\nu \left(\sum_{j=1}^n c_j f(t_j) \right) = \sum_{j=1}^n c_j y_j (1 + e_j),$$

gdzie $|e_j| \lesssim (n+1)\nu$. (W porównaniu z iloczynem skalarnym mamy tu $n+1$ zamiast $n+2$, bo c_j nie jest formalnie daną i nie podlega zaburzeniu.) Stąd błąd bezwzględny kwadratury w fl_ν można oszacować następująco:

$$\left| \int_a^b f(x) dx - fl_\nu \left(\sum_{j=1}^n c_j f(t_j) \right) \right| \leq \left| \int_a^b f(x) dx - \sum_{j=1}^n c_j f(t_j) \right| + (n+1)\nu \sum_{j=1}^n |c_j f(t_j)|.$$

Uwagi i uzupełnienia

U. 2.1 Wprowadzimy teraz formalną definicję zastosowanej już wcześniej przybliżonej nierówności ‘ \lesssim ’. Dla dwóch funkcji ϕ_1 i ϕ_2 zmiennej ν piszemy

$$|\phi_1(\nu)| \lesssim |\phi_2(\nu)| \quad \text{gdzie} \quad \limsup_{\nu \rightarrow 0^+} \frac{|\phi_1(\nu)|}{|\phi_2(\nu)|} \leq 1.$$

Na przykład, jeśli $|\phi(\nu)| \leq \sum_{j=1}^n K_j \nu^j$ i $K_1 > 0$, to $|\phi(\nu)| \lesssim K_1 \nu$, co już zauważyliśmy w przykładzie z iloczynem skalarnym. Zapis ten wyraża prosty fakt, że dla praktycznych wartości ν ($\nu \leq 10^{-16}$) wyrazy typu ν^2 , ν^3 , itd. są tak małe w porównaniu z ν , że można je zaniedbać.

Łatwo sprawdzić, że dla nierówności ‘ \lesssim ’ zachodzą następujące fakty. Jeśli $|\varepsilon| \lesssim K\nu$, $|\varepsilon_1| \lesssim K_1\nu$ i $|\varepsilon_2| \lesssim K_2\nu$ (gdzie ε , ε_1 , ε_2 są funkcjami ν), to

$$(1 + \varepsilon_1)(1 + \varepsilon_2) = (1 + \delta), \quad \text{gdzie} \quad |\delta| \lesssim (K_1 + K_2)\nu, \quad (2.3)$$

$$(1 + \varepsilon)^{-1} = (1 + \delta), \quad \text{gdzie} \quad |\delta| \lesssim K\nu, \quad (2.4)$$

$$(1 + \varepsilon)^{1/2} = (1 + \delta), \quad \text{gdzie} \quad |\delta| \lesssim \frac{1}{2}K\nu, \quad (2.5)$$

i ogólnie

$$(1 + \varepsilon)^p = (1 + \delta), \quad \text{gdzie} \quad |\delta| \lesssim |p|\nu. \quad (2.6)$$

Niekiedy wygodnie będzie również użyć asymptotycznej równości $|\phi_1(\nu)| \approx |\phi_2(\nu)|$, która oznacza, że

$$\lim_{\nu \rightarrow 0^+} \frac{|\phi_1(\nu)|}{|\phi_2(\nu)|} = 1.$$

U. 2.2 Rozpatrzmy teraz zadanie obliczenia pierwiastków rzeczywistych równania kwadratowego z Przykładu 1.1. Będziemy zakładać, że model obliczeniowy dopuszcza obliczanie pierwiastków kwadratowych z liczb nieujemnych oraz $f_{\nu}(\sqrt{x}) = rd(\sqrt{rd(x)})$.

Okazuje się, że nie umiemy pokazać numerycznej poprawności “szkolnego” algorytmu obliczającego pierwiastki równania bezpośrednio ze wzorów (1.1). Można jednak pokazać numeryczną poprawność drobnej jego modyfikacji wykorzystującej wzory Viete’a.

```

Δ := p * p - q;
if (Δ < 0) then OUT(∅) else
if (Δ = 0) then OUT(p) else
begin
  Δ1 := sqrt(d);
  if (p ≥ 0) then
    begin
      x1 := p + Δ1;
      x2 := q/x1;
    end else
      begin
        x2 := p - Δ1;
        x1 := q/x2;
      end;
  OUT(x1); OUT(x2)
end.

```

Mamy bowiem (zakładamy, że $\Delta > 0$, bo pozostałe przypadki są oczywiste)

$$\begin{aligned}
f_{\nu}(\Delta(p, q)) &= \left(p^2(1 + \alpha)^2(1 + \varepsilon_1) - q(1 + \beta) \right) (1 + \varepsilon_2) \\
&= \left(p^2 - q \frac{(1 + \beta)}{(1 + \alpha)^2(1 + \varepsilon_1)} \right) (1 + \varepsilon_2)(1 + \alpha)^2(1 + \varepsilon_1) \\
&= \left(p^2 - q(1 + \delta) \right) (1 + \gamma) = \Delta(p, q(1 + \delta))(1 + \gamma),
\end{aligned}$$

gdzie $|\delta|, |\gamma| \lesssim 4\nu$. Wyróżnik obliczony w f_{ν} jest więc nieco zaburzonym wyróżnikiem dokładnym dla danych p i $q_{\nu} = q(1 + \delta)$. W szczególności

$$\operatorname{sgn}(f_{\nu}(\Delta(p, q))) = \operatorname{sgn}(\Delta(p, q_{\nu})).$$

Jeśli $p \geq 0$ to

$$\begin{aligned}
f_{\nu}(x_1(p, q)) &= \left(p(1 + \alpha) + \sqrt{f_{\nu}(\Delta(p, q))(1 + \varepsilon_3)} \right) (1 + \varepsilon_4) \\
&= \left(p(1 + \alpha) + \sqrt{\Delta(p, q_{\nu})(1 + \gamma)(1 + \varepsilon_3)} \right) (1 + \varepsilon_4) \\
&= \left(p + \sqrt{\Delta(p, q_{\nu})} \frac{\sqrt{1 + \gamma(1 + \varepsilon_3)}}{1 + \alpha} \right) (1 + \varepsilon_4)(1 + \alpha) \\
&= \left(p + \sqrt{\Delta(p, q_{\nu})} \right) (1 + e_1),
\end{aligned}$$

gdzie $|e_1| \lesssim 6\nu$. Zauważmy, że ostatnia równość zachodzi dlatego, że dodajemy liczby tego samego znaku. (Inaczej $|e_1|$ mogłaby być dowolnie duża i tak byłoby w algorytmie szkolnym.) Dla drugiego pierwiastka mamy

$$f_{\nu}(x_2(p, q)) = \frac{q(1 + \beta)}{f_{\nu}(x_1(p, q))} (1 + \varepsilon_5) = \frac{q_{\nu}}{f_{\nu}(x_1(p, q))} (1 + e_2),$$

gdzie $|e_2| \lesssim 8\nu$.

Podobny wynik otrzymalibyśmy dla $p < 0$. Algorytm zmodyfikowany jest więc numerycznie poprawny, gdyż otrzymane w f_ν pierwiastki są nieco zaburzonymi dokładnymi pierwiatkami dla danych $p_\nu = p$ i $q_\nu = q(1 + \delta)$.

Aby oszacować błąd algorytmu, wystarczy zbadać uwarunkowanie zadania ze względu na zaburzenie danej q , ponieważ pokazaliśmy, że zaburzenia p można przenieść na zaburzenia q i wyniku. Niestety, chociaż algorytm jest numerycznie poprawny, zaburzenia q mogą sprawić, że nawet znak wyróżnika Δ może być obliczony nieprawidłowo. Na przykład dla $p = 1$ i $q = 1 \pm 10^{t+1}$ mamy $\Delta(p, q) = \mp 10^{t+1}$, ale $\Delta(\text{rd}(p), \text{rd}(q)) = \Delta(1, 1) = 0$. Ogólnie

$$|f_\nu(\Delta(p, q)) - \Delta(p, q)| \lesssim 4\nu(p^2 + 2|q|),$$

a więc tylko dla $|\Delta(p, q)| = |p^2 - q| > 4\nu(p^2 + 2|q|)$ możemy być pewni obliczenia właściwego znaku Δ . Aby zobaczyć jak zaburzenie danej q przenosi się na zaburzenie wyniku posłużymy się wynikiem z Ćw. 2.2. Mamy

$$\left| \frac{\partial x_1(p, q)}{\partial q} \right| = \left| \frac{\partial x_2(p, q)}{\partial q} \right| = \frac{1}{\sqrt{p^2 - q}},$$

a stąd dla $|q_\nu - q| \leq \varepsilon|q|$ błąd względny w normie drugiej wynosi

$$\frac{\left(|x_1(p, q_\nu) - x_1(p, q)|^2 + |x_2(p, q_\nu) - x_2(p, q)|^2 \right)^{1/2}}{\left(|x_1(p, q)|^2 + |x_2(p, q)|^2 \right)^{1/2}} \lesssim \frac{|q|/p^2}{\sqrt{1 - q/p^2} \sqrt{2 - q/p^2}} \varepsilon.$$

Zadanie jest więc dobrze uwarunkowane dla $q/p^2 \ll 1$ i może być źle uwarunkowane dla $q/p^2 \approx 1$. W ostatnim przypadku nie możemy być pewni otrzymania dobrego wyniku w f_ν .

U. 2.3 Najważniejszym wnioskiem tego rozdziału jest, że jeśli algorytm jest dokładny i numerycznie poprawny oraz zadanie jest dobrze uwarunkowane to możemy być pewni dobrej jakości wyniku w f_ν . Okazuje się, że dla tego wyniku wystarczy numeryczna poprawność algorytmu "po współrzędnych".

Rzeczywiście, rozpatrzmy zadanie $S = (S_1, S_2, \dots, S_n) : X \rightarrow \mathbb{R}^n$ i dokładny algorytm $\Phi = (\Phi_1, \Phi_2, \dots, \Phi_n)$ je rozwiązujący, który jest numerycznie poprawny "po współrzędnych" w klasie $X_0 \subseteq X$. To znaczy, że każde Φ_i spełnia warunek: dla każdego $x \in X_0$ istnieje $\tilde{x} \in X$ (przy czym \tilde{x} może być różne dla różnych i !) takie, że

$$\|\tilde{x} - x\| \leq K_i \nu \|x\| \quad \text{i} \quad |S_i(\tilde{x}) - f_\nu(\Phi_i(x))| \leq L_i \nu |S_i(\tilde{x})|.$$

Załóżmy dodatkowo, że uwarunkowanie zadania S wynosi M , tzn.

$$\text{jeśli } \|\tilde{x} - x\| \leq \varepsilon \|x\| \quad \text{to} \quad \|S(\tilde{x}) - S(x)\|_p \leq M\varepsilon \|S(x)\|_p.$$

Wtedy, wobec tego, że $|S_i(x)| \leq \|S(x)\|_p$ oraz $|S_i(\tilde{x}) - S_i(x)| \leq M\varepsilon \|S(x)\|_p$, dla każdej "współrzędnej" Φ_i mamy

$$\begin{aligned} |S_i(x) - f_\nu(\Phi_i(x))| &\leq |S_i(x) - S_i(\tilde{x})| + |S_i(\tilde{x}) - f_\nu(\Phi_i(x))| \\ &\leq MK_i \nu \|S(x)\|_p + L_i \nu (|S_i(\tilde{x}) - S_i(x)| + |S_i(x)|) \\ &\leq (MK_i + L_i + MK_i L_i \nu) \nu \|S(x)\|_p. \end{aligned}$$

To zaś oznacza, że

$$\|S(x) - f_\nu(\Phi(x))\|_p \lesssim K \nu \|S(x)\|_p \quad \text{gdzie} \quad K = \left(\sum_{i=1}^n (MK_i + L_i)^p \right)^{1/p}.$$

Ćwiczenia

Ćw. 2.1 Pokaż równości (2.3)–(2.6).

Ćw. 2.2 Niech $0 < a_1 < a_2 < \dots < a_n$. Czy z punktu widzenia błędów w fl_ν lepiej jest policzyć sumę tych liczb w kolejności od najmniejszej do największej czy odwrotnie?

Ćw. 2.3 Aby obliczyć $S(a, b) = a^2 - b^2$ można zastosować dwa algorytmy: $\Phi_1(a, b) = a * a - b * b$ oraz $\Phi_2(a, b) = (a + b) * (a - b)$. Uzasadnij, że oba algorytmy są numerycznie poprawne, ale drugi z nich wywołuje mniejszy błąd względny wyniku w przypadku, gdy $rd(a) = a$ i $rd(b) = b$.

Ćw. 2.4 Czy zjawisko redukcji cyfr znaczących opisane w U. 1.2 oznacza, że zadanie odejmowania liczb ‘podobnych’ nie jest dobrze uwarunkowane?

Ćw. 2.5 Rozpatrzmy zadanie $S : [a, b] \rightarrow \mathbb{R}$, gdzie S jest funkcją różniczkowalną na $[a, b]$. Pokaż, że wtedy za uwarunkowanie zadania dla danej x można w przybliżeniu przyjąć

$$\frac{|xS'(x)|}{\max(\eta, |S(x)|)}$$

Ćw. 2.6 Wykaż, że algorytmy obliczania wartości każdej z niewiadomych x i y w układzie dwóch równań liniowych z dwiema niewiadomymi za pomocą wzorów Cramera są numerycznie poprawne.

Ćw. 2.7 Wykaż, że naturalny algorytm obliczania cosinusa kąta między dwoma wektorami $\vec{a}, \vec{b} \in \mathbb{R}^n$,

$$\cos(\vec{a}, \vec{b}) = \frac{\sum_{j=1}^n a_j b_j}{\sqrt{\left(\sum_{j=1}^n a_j^2\right) \left(\sum_{j=1}^n b_j^2\right)}},$$

jest numerycznie poprawny. Oszacować błąd względny wyniku w fl_ν .

Ćw. 2.8 Pokaż, że naturalny algorytm obliczania $\|A\vec{x}\|_2$ dla danej macierzy $A \in \mathbb{R}^{n \times n}$ i wektora $\vec{x} \in \mathbb{R}^n$ jest numerycznie poprawny. Dokładniej,

$$fl_\nu(\|A\vec{x}\|_2) = (A + E)\vec{x},$$

gdzie $\|E\|_2 \lesssim k(n)\nu\|A\|_2$ i $k(n) = 3\sqrt{n}(\frac{n}{2} + 1)$. Ponadto, jeśli A jest nieosobliwa to

$$|fl_\nu(\|A\vec{x}\|_2) - \|A\vec{x}\|_2| \lesssim k(n)\nu \left(\|A\|_2\|A^{-1}\|_2\right) \|A\vec{x}\|_2.$$

Ćw. 2.9 Niech Φ będzie algorytmem numerycznie poprawnym w zbiorze danych $f \in F_0$, przy czym dla małych ν , $fl_\nu(\Phi(f)) = \varphi(y_\nu)$, gdzie $\|y_\nu - y\| \leq K\nu\|y\|$ i K nie zależy od ν i f ($y = N(f)$). Pokaż, że w ogólności Φ nie musi być “numerycznie poprawny po współrzędnych”, tzn. w ogólności nie istnieje bezwzględna stała K_1 taka, że dla małych ν i dla dowolnej $f \in F_0$

$$|y_{\nu,j} - y_j| \leq K_1 \nu |y_j|, \quad 1 \leq j \leq n,$$

gdzie $y = (y_1, \dots, y_n)$.

Ćw. 2.10 Rozpatrzmy dwa zadania, $S : F \rightarrow G$ i $T : G \rightarrow H$. Niech Φ i Ψ będą algorytmami dokładnymi rozwiązującymi, odpowiednio, zadania S i T . Oba algorytmy są numerycznie poprawne w zbiorach danych $F_0 \subset F$ i $G_0 \subset G$, przy czym $S(F_0) \subset G_0$, tzn. istnieją $\nu_0 > 0$, $K_1, K_2, K_3, K_4 > 0$, takie że dla $\nu \leq \nu_0$ i dowolnych $f \in F_0$ i $g \in G_0$ można wskazać $\tilde{f} \in F$, $\tilde{g} \in G$, spełniające:

$$\begin{aligned} \|\tilde{f} - f\| &\leq K_1\nu\|f\|, & \|\mathcal{I}_\nu(\Phi(f)) - S(\tilde{f})\| &\leq K_2\nu\|S(\tilde{f})\|, \\ \|\tilde{g} - g\| &\leq K_3\nu\|g\|, & \|\mathcal{I}_\nu(\Psi(g)) - T(\tilde{g})\| &\leq K_4\nu\|T(\tilde{g})\|. \end{aligned}$$

W danej przestrzeni unormowanej X , niech $\mathcal{B}(x, r) = \{x_1 \in X : \|x_1 - x\| \leq r\|x\|\}$. Załóżmy, że istnieją $\delta_0 > 0$ i $M > 0$ takie, że

- (a) $\forall \delta \leq \delta_0 \quad \forall f \in F \quad \mathcal{B}(S(f), \delta) \subseteq S(\mathcal{B}(f, \delta M))$, albo
 (b) $\forall \delta \leq \delta_0 \quad \forall g \in G \quad T(\mathcal{B}(g, \delta)) \subseteq \mathcal{B}(T(g), \delta M)$.

Wykaż, że wtedy złożenie $\Psi \circ \Phi$ jest algorytmem dokładnym dla zadania złożenia $T \circ S : F \rightarrow H$, który jest też numerycznie poprawny w F_0 . Dokładniej dla $f \in F_0$ istnieje $f^* \in F$ takie że: w przypadku (a)

$$\|f^* - f\| \leq K\nu\|f\|, \quad \|\mathcal{I}_\nu((\Psi \circ \Phi)(f)) - (T \circ S)(f^*)\| \leq K_4\nu\|(T \circ S)(f^*)\|,$$

gdzie $K \lesssim (K_2 + K_3)M + K_1$, a w przypadku (b)

$$\|f^* - f\| \leq K_1\nu\|f\|, \quad \|\mathcal{I}_\nu((\Psi \circ \Phi)(f)) - (T \circ S)(f^*)\| \leq K\nu\|(T \circ S)(f^*)\|,$$

gdzie $K \lesssim (MK_5 + K_4)$.

Rozdział 3

Układy równań liniowych

Rozpoczynamy analizę metod dokładnych rozwiązywania układów równań liniowych postaci

$$A\vec{x} = \vec{b}, \quad (3.1)$$

gdzie $A = (a_{i,j})_{i,j=1}^n$ jest macierzą nieosobliwą ($\det A \neq 0$) formatu $n \times n$, a $\vec{b} = (b_i)_{i=1}^n$ jest wektorem z \mathbb{R}^n . Zakładamy, że informacja o zadaniu dana jest przez współczynniki $a_{i,j}$ macierzy i współrzędne b_i wektora. W wyniku powinniśmy uzyskać współrzędne x_j^* wektora rozwiązania $\vec{x}^* = (x_j^*)_{j=1}^n$,

$$\vec{x}^* = A^{-1}\vec{b}.$$

Przypomnijmy, że nieosobliwość macierzy A zapewnia, że rozwiązanie \vec{x}^* istnieje i jest wyznaczone jednoznacznie.

3.1 Układy z macierzą trójkątną

Szczególnym przypadkiem (3.1) jest układ z macierzą trójkątną A . Będą nas interesować macierze *trójkątne górne*, dla których $a_{i,j} = 0$ gdy $i > j$, oraz macierze *trójkątne dolne* z jedynekami na przekątnej, tzn. $a_{i,j} = 0$, $i < j$, oraz $a_{i,i} = 1$. Macierze pierwszego rodzaju będziemy oznaczać przez U , a drugiego rodzaju przez L .

Układ z macierzą trójkątną górną

$$U\vec{x} = \vec{c}, \quad (3.2)$$

gdzie $U = (r_{i,j})$, $\vec{c} = (c_j)$, można rozwiązać stosując algorytm:

```
x_n^* := c_n / u_{n,n};  
for i := n - 1 downto 1 do  
  x_i^* := (c_i - \sum_{j=i+1}^n u_{i,j} x_j^*) / u_{i,i}.
```

(Algorytm ten jest wykonalny, ponieważ nieosobliwość macierzy implikuje, że $r_{i,i} \neq 0$ dla wszystkich i .) Podobnie, układ $L\vec{x} = \vec{c}$ rozwiązujemy algorytmem:

```
x_1^* := c_1;  
for i := 2 to n do  
  x_i^* := c_i - \sum_{j=1}^{i-1} l_{i,j} x_j^*.
```

Oba algorytmy wymagają $\frac{1}{2}n^2$ mnożeń i dzieleni oraz $\frac{1}{2}n^2$ dodawań i odejmowań, a więc n^2 wszystkich działań arytmetycznych. (Tu i dalej uwzględniamy jedynie składniki kosztu o najwyższym rzędzie.)

3.2 Eliminacja Gaussa

Eliminacja Gaussa jest algorytmem dokładnym rozwiązywania układów równań z macierzą pełną A , polegającym na sprowadzeniu układu wyjściowego (3.1) do układu trójkątnego (3.2).

W postaci rozwiniętej układ (3.1) ma postać

$$\begin{aligned} a_{1,1}x_1 + a_{1,2}x_2 + \cdots + a_{1,n}x_n &= b_1 \\ a_{2,1}x_1 + a_{2,2}x_2 + \cdots + a_{2,n}x_n &= b_2 \\ \cdots & \\ a_{n,1}x_1 + a_{n,2}x_2 + \cdots + a_{n,n}x_n &= b_n. \end{aligned}$$

Zauważmy, że jeśli od i -tego wiersza tego układu odejmiemy wiersz pierwszy pomnożony przez $l_{i,1} = a_{i,1}/a_{1,1}$, dla $i = 2, 3, \dots, n$ (zakładamy, że $a_{1,1} \neq 0$) to otrzymamy układ równoważny $A^{(1)}\vec{x} = \vec{b}^{(1)}$,

$$\begin{aligned} a_{1,1}x_1 + a_{1,2}x_2 + \cdots + a_{1,n}x_n &= b_1 \\ a_{2,2}^{(1)}x_2 + \cdots + a_{2,n}^{(1)}x_n &= b_2^{(1)} \\ \cdots & \\ a_{n,2}^{(1)}x_2 + \cdots + a_{n,n}^{(1)}x_n &= b_n^{(1)}, \end{aligned}$$

z wyrugowaną zmienną x_1 z równań od drugiego do n -tego. Teraz z kolei możemy od i -tego wiersza odjąć wiersz drugi pomnożony przez $l_{i,2} = a_{i,2}^{(1)}/a_{2,2}^{(1)}$, dla $i = 3, 4, \dots, n$ (o ile $a_{2,2}^{(1)} \neq 0$), aby otrzymać układ $A^{(2)}\vec{x} = \vec{b}^{(2)}$ z wyrugowanymi zmiennymi x_1 i x_2 z równań od trzeciego do n -tego. Postępując tak dalej otrzymujemy po $n - 1$ krokach układ $A^{(n-1)}\vec{x} = \vec{b}^{(n-1)}$ postaci

$$\begin{aligned} a_{1,1}^{(0)}x_1 + a_{1,2}^{(0)}x_2 + \cdots + a_{1,n}^{(0)}x_n &= b_1 \\ a_{2,2}^{(1)}x_2 + \cdots + a_{2,n}^{(1)}x_n &= b_2^{(1)} \\ \cdots & \\ a_{n,n}^{(n-1)}x_n &= b_n^{(n-1)}. \end{aligned}$$

Układ ten jest trójkątny górny, $U\vec{x} = \vec{c}$, z macierzą $U = A^{(n-1)}$ i wektorem $\vec{c} = \vec{b}^{(n-1)}$, można go więc rozwiązać znaną już metodą.

Kolejne operacje w eliminacji Gaussa można zapisać następująco:

```

for  $k := 1$  to  $n - 1$  do
  for  $i := k + 1$  to  $n$  do
    begin
       $l_{i,k} := a_{i,k}^{(k-1)} / a_{k,k}^{(k-1)}$ ;
       $b_i^{(k)} := b_i^{(k-1)} - l_{i,k} * b_k^{(k-1)}$ ;
      for  $j := k + 1$  to  $n$  do
         $a_{i,j}^{(k)} := a_{i,j}^{(k-1)} - l_{i,k} * a_{k,j}^{(k-1)}$ 
    end.

```

Zauważmy, że k -ty krok wymaga $(n - k)^2$ dodawań i odejmowań oraz tyle samo mnożeń i dzielení. Koszt całej eliminacji wynosi więc $2 \sum_{k=1}^{n-1} (n - k)^2 \approx \frac{2}{3}n^3$ operacji arytmetycznych. Ponieważ układ trójkątny rozwiązujemy kosztem n^2 , całkowity koszt rozwiązania układu $A\vec{x} = \vec{b}$ jest poporcjonalny do $\frac{2}{3}n^3$.

Opisany powyżej proces eliminacyjny załamie się gdy dla pewnego k element $a_{k,k}^{(k-1)}$ będzie zerem. W takim przypadku prosta modyfikacja polegająca na przestawieniu wiersza k -tego układu z wierszem s_k -tym ($k+1 \leq s_k \leq n$) dla którego $a_{s_k,k}^{(k-1)} \neq 0$, pozwala kontynuować eliminację. Taki niezerowy element oczywiście istnieje, bo inaczej macierz $A^{(k-1)}$, a tym samym i macierz A , byłyby osobliwe.

Najczęściej wybieramy s_k tak, aby

$$|a_{s_k,k}^{(k-1)}| = \max_{k+1 \leq i \leq n} |a_{i,k}^{(k-1)}|,$$

co wymaga dodatkowo rzędu $n^2/2$ porównań liczb rzeczywistych i żadnych operacji arytmetycznych. Taki wybór elementu niezerowego prowadzi do *eliminacji Gaussa z wyborem elementu głównego w kolumnie*. Wtedy współczynniki $l_{i,k} = a_{i,k}^{(k-1)} / a_{k,k}^{(k-1)}$ są wszystkie co do modułu nie większe od jedności co, jak się przekonamy później, ma niebagatelne znaczenie dla własności numerycznych całego algorytmu.

3.3 Rozkład trójkątno-trójkątny macierzy

Podamy teraz ważne twierdzenia wiążące eliminację Gaussa z rozkładem trójkątno-trójkątnym macierzy A .

Twierdzenie 3.1 *Eliminacja Gaussa bez przestawień wierszy (o ile jest wykonalna) jest równoważna rozkładowi macierzy A na iloczyn macierzy trójkątnej dolnej $L = (l_{i,j})$ (z jedynkami na przekątnej) i trójkątnej górnej $U = (r_{i,j})$. Dokładniej,*

$$A = LU,$$

gdzie

$$l_{i,j} = \begin{cases} a_{i,j}^{(i-1)} / a_{i,i}^{(i-1)} & i > j, \\ 1 & i = j, \\ 0 & i < j, \end{cases}$$

oraz

$$u_{i,j} = \begin{cases} a_{i,j}^{(i-1)} & i < j, \\ 0 & i \geq j. \end{cases}$$

Dowód. Zauważmy, że rugowanie elementów pod główną przekątną w k -tej kolumnie odpowiada mnożeniu układu $A^{(k-1)}\vec{x} = \vec{b}^{(k-1)}$ z lewej strony przez macierz

$$L_k = \begin{pmatrix} 1 & & & & & \\ & 1 & & & & \\ & & \ddots & & & \\ & & & 1 & & \\ & & & -l_{k+1,k} & 1 & \\ & & & \vdots & & \ddots \\ & & & -l_{n,k} & & & 1 \end{pmatrix}.$$

Stąd

$$L_{n-1}L_{n-2} \cdots L_2L_1A = U$$

i

$$A = (L_1^{-1}L_2^{-1} \cdots L_{n-2}^{-1}L_{n-1}^{-1})U.$$

Łatwo sprawdzić, że macierz L_k^{-1} różni się od L_k tylko tym, że elementy $-l_{i,k}$ dla $k+1 \leq i \leq n$ zmieniają się na $l_{i,k}$. Ponadto, mnożenie macierzy L_k^{-1} przez siebie w podanej kolejności odpowiada "przepisywaniu" elementów $l_{i,k}$ do macierzy wynikowej tak, że ostatecznie mamy $L_1^{-1} \cdots L_{n-1}^{-1} = L$ i $A = LU$. \square

Zobaczmy teraz jak wybór elementu głównego w kolumnie i przestawianie odpowiednich wierszy macierzy układu wpływa na rozkład macierzy A z Twierdzenia 3.1.

Przestawienie wiersza k -tego z s -tym w danej macierzy jest formalnie równoważne pomnożeniu tej macierzy z lewej strony przez macierz *permutacji elementarnej* $P_{k,s}$. Jest to macierz, która powstaje z macierzy jednostkowej przez przestawienie jej k -tego wiersza z s -tym (albo k -tej kolumny z s -tą). Eliminacja Gaussa z wyborem elementu głównego w kolumnie odpowiada więc mnożeniu macierzy A z lewej strony kolejno przez macierze permutacji elementarnej P_{k,s_k} i macierze "eliminacji" L_k . Mamy więc

$$L_{n-1}P_{n-1,s_{n-1}}L_{n-2}P_{n-2,s_{n-2}} \cdots L_2P_{2,s_2}L_1P_{1,s_1}A = U.$$

Ponieważ $P_{k,l}^2 = I$ (identyczność), mamy

$$P_{2,s_2}L_1P_{1,s_1}A = (P_{2,s_2}L_1P_{2,s_2})(P_{2,s_2}P_{1,s_1}A) = L_1^{(1)}(P_{2,s_2}P_{1,s_1}A),$$

gdzie $L_1^{(1)}$ różni się od L_1 tym, że zostały przestawione elementy $(2, 1)$ i $(s_2, 1)$. Następnie,

$$\begin{aligned} P_{3,s_3}L_2L_1^{(1)}(P_{2,s_2}P_{1,s_1}A) &= (P_{3,s_3}L_2P_{3,s_3})(P_{3,s_3}L_1^{(1)}P_{3,s_3})(P_{3,s_3}P_{2,s_2}P_{1,s_1}A) \\ &= L_2^{(2)}L_1^{(2)}(P_{3,s_3}P_{2,s_2}P_{1,s_1}A), \end{aligned}$$

gdzie $L_1^{(2)}$ różni się od $L_1^{(1)}$ tym, że zostały przestawione elementy $(3, 1)$ i $(s_3, 1)$, a $L_2^{(2)}$ różni się od L_2 przestawieniem elementów $(3, 2)$ i $(s_3, 2)$. Postępując tak dalej otrzymujemy

$$\begin{aligned} L_{n-1}P_{n-1,s_{n-1}} \cdots L_2P_{2,s_2}L_1P_{1,s_1}A \\ = L_{n-1}^{(n-1)}L_{n-1}^{(n-1)} \cdots L_2^{(n-1)}L_1^{(n-1)}(P_{n-1,s_{n-1}} \cdots P_{2,s_2}P_{1,s_1})A = U, \end{aligned}$$

gdzie $L_k^{(n-1)}$ różni się od L_k jedynie pewną permutacją elementów w k -tej kolumnie pod główną przekątną,

$$L_k^{(n-1)} = P_{n-1,s_{n-1}} \cdots P_{k+1,s_{k+1}}L_kP_{k+1,s_{k+1}} \cdots P_{n-1,s_{n-1}}.$$

Postępując dalej tak, jak w przypadku eliminacji bez przestawień wierszy, otrzymujemy następujący wniosek.

Wniosek 3.1 *Eliminacja Gaussa z wyborem elementu głównego w kolumnie jest wykonalna i jest równoważna rozkładowi*

$$PA = LU,$$

gdzie $P = P_{n-1,s_{n-1}} \cdots P_{2,s_2}P_{1,s_1}$ jest macierzą permutacji, a L i U są macierzami konstruowanymi tak jak w Twierdzeniu 3.1, ale dla macierzy PA zamiast A . Ponadto wszystkie elementy macierzy L są co do modułu nie większe od jedności. \square

Dodajmy, że znalezienie czynników rozkładu $PA = LU$ dla danej macierzy A kosztuje tyle samo co eliminacja Gaussa z wyborem elementu głównego, czyli $\frac{2}{3}n^3$ operacji arytmetycznych i $\frac{1}{2}n^2$ porównań. (Tu uwzględniamy jedynie składniki najwyższego rzędu.)

W praktycznej realizacji rozkładu $PA = LU$ oczywiście nie musimy przy przestawieniu wierszy dokonywać fizycznie przepisać elementów danego wiersza do innego. Wystarczy mieć dodatkowy n -wymiarowy wektor permutacji $p[\cdot]$ interpretowany w ten sposób, że $p[i]$ jest wskaźnikiem do i -tego wiersza macierzy A . Przestawienie wierszy możemy wtedy realizować po prostu przez zamianę wskaźników, czyli elementów wektora p . (Zob. U. 3.5.)

Twierdzenie 3.1 i Wniosek 3.1 mają nie tylko teoretyczne znaczenie. Rozkładu macierzy na czynniki trójkątne (co w praktyce sprowadza się do pamiętania kolejnych permutacji i mnożników $l_{i,j}$) opłaca się dokonywać w przypadku, gdy zadanie polega na rozwiązaniu nie jednego, ale wielu układów równań z tą samą macierzą A i ze zmieniającym się wektorem prawej strony,

$$A\vec{x} = \vec{b}_s \quad \text{dla } 1 \leq s \leq k.$$

Jeśli każdy z tych układów rozwiązujemy “od początku” to musimy wykonać $\frac{2}{3}n^3k$ operacji arytmetycznych. Koszt ten możemy znacznie zredukować, jeśli dokonamy wstępnego rozkładu $PA = LU$. Wtedy układ $A\vec{x} = \vec{b}_s$ jest równoważny układowi $LU\vec{x} = P\vec{b}_s$, a więc jego rozwiązanie sprowadza się do rozwiązania dwóch znanych już układów trójkątnych:

$$L\vec{y} = P\vec{b}_s \quad \text{i} \quad U\vec{x} = \vec{y}.$$

Taki sposób postępowania wymaga tylko $\frac{2}{3}n^3 + 2n^2k = 2n^2\left(\frac{1}{3}n + k\right)$ operacji arytmetycznych, co np. przy $k = n$ powoduje obniżenie rzędu kosztu z n^4 do n^3 .

Uwagi i uzupełnienia

U. 3.1 Zaproponowany algorytm dla układów $U\vec{x} = \vec{c}$ z macierzą trójkątną U ma prawie optymalną złożoność kombinatoryczną n^2 wśród algorytmów dokładnych. Zauważmy bowiem, że jeśli algorytm jest dokładny to wykorzystuje wszystkie dane $r_{i,j}$ i c_j , $1 \leq i, j \leq n$ (zob. Ćw.3.3). Ponieważ każda kolejna operacja arytmetyczna (+, −, *, /) może wykorzystywać co najwyżej dwie nowe dane, a wszystkich danych jest $\frac{1}{2}n(n+1) + n \approx \frac{1}{2}n^2$, algorytm dokładny musi wykonywać co najmniej $\frac{1}{4}n^2$ działań.

U. 3.2 Znane są algorytmy dokładne, które rozwiązują układ równań z macierzą pełną kosztem proporcjonalnym do n^p , gdzie $p < 3$, a więc mniejszym niż w eliminacji Gaussa. Algorytmy te są jednak nieużyteczne ze względu na skomplikowaną konstrukcję, zwykle dużą stałą przy n^p , a także ze względu na złe własności numeryczne. Oczywiście, ograniczeniem dolnym na wykładnik p w koszcie rozwiązania układu z macierzą pełną jest 2 (bo mamy rzędu n^2 danych), ale nie wiadomo, czy może on być osiągnięty przez jakiś algorytm.

U. 3.3 Układy równań z tą samą macierzą, ale ze zmieniającą się prawą stroną równania powstają często przy rozwiązywaniu, np. równań różniczkowych cząstkowych, gdzie prawa strona układu odpowiada zmieniającym się warunkom brzegowym. Z wieloma układami tego typu mamy również do czynienia gdy chcemy znaleźć macierz odwrotną do danej macierzy $A \in \mathbb{R}^{n \times n}$. Rzeczywiście, kolejne kolumny macierzy A^{-1} są rozwiązaniami układów

$$A\vec{x} = \vec{e}_j, \quad 1 \leq j \leq n,$$

gdzie \vec{e}_j oznacza j -ty element bazy kanonicznej.

U. 3.4 Obok wyboru elementu głównego w kolumnie dokonuje się również wyboru elementu głównego w całej macierzy. Dokładniej, w k -tym kroku eliminacyjnym wybiera się element $a_{s_k, t_k}^{(k-1)}$, $k \leq s_k, t_k \leq n$, taki, że

$$|a_{s_k, t_k}^{(k-1)}| = \max_{k \leq i, j \leq n} |a_{i, j}^{(k-1)}|,$$

a następnie przestawia się wiersze k -ty z s_k -tym i kolumny k -tą z t_k -tą w macierzy $A^{(k-1)}$. Zauważmy, że przestawienie kolumn macierzy odpowiada mnożeniu jej z prawej strony przez macierz permutacji P_{k,t_k} . Przyjmując $R = P_{1,t_1} P_{2,t_2} \cdots P_{n-1,t_{n-1}}$ i korzystając z Twierdzenia 3.1 i Wniosku 3.1 mamy, że eliminacja Gaussa z wyborem elementu głównego w całej macierzy jest równoważna rozkładowi

$$PAR = LU.$$

U. 3.5 Podamy teraz jedną z możliwych implementacji algorytmu rozkładu macierzy na iloczyn macierzy trójkątnych, $PA = LR$. W poniższym programie współczynniki macierzy wyjściowej są pamiętane w tablicy $a[\cdot, \cdot]$. Po wykonaniu programu, informacja o permutacji wierszy zapamiętana będzie w wektorze $p[\cdot]$. W $a[p[i], j]$ będą elementy (i, j) macierzy L dla $i > j$, oraz macierzy U dla $i \leq j$.

```

{ inicjacja wskaźników }
for  $i := 1$  to  $n$  do  $p[i] := i$ ;
for  $j := 1$  to  $n - 1$  do
begin
  { wybór elementu głównego }
   $im := j$ ;  $val := \text{abs}(a[p[j], j])$ ;
  for  $i := j + 1$  to  $n$  do
  begin
     $v := \text{abs}(a[p[i], j])$ ;
    if  $v > val$  then
    begin
       $im := i$ ;  $val := v$ 
    end;
  { zamiana wskaźników }
   $s := p[im]$ ;  $p[im] := p[j]$ ;  $p[j] := s$ ;
  { eliminacja }
   $v := a[s, j]$ ;
  for  $i := j + 1$  to  $n$  do
  begin
     $im := p[i]$ ;
     $l := a[im, j] / v$ ;
     $a[im, j] := l$ ;
    for  $k := j + 1$  to  $n$  do
     $a[im, k] := a[im, k] - l * a[s, k]$ 
  end;
  end;
end;
end.

```

U. 3.6 Przedstawiony algorytm eliminacji Gaussa rozwiązuje układy z macierzami dowolnej postaci. Jeśli macierz A jest szczególnej postaci to czasem proste jego modyfikacje pozwalają znacznie zmniejszyć koszt uzyskania rozwiązania. Na przykład, niech A będzie macierzą trójdziagonalną postaci

$$A = \begin{pmatrix} a_1 & c_1 & & & \\ b_2 & a_2 & c_2 & & \\ & b_3 & a_3 & c_3 & \\ & & \ddots & \ddots & \ddots \\ & & & b_n & a_n \end{pmatrix}.$$

Wtedy układ $A\vec{x} = \vec{e}$ można rozwiązać stosując tzw. algorytm “przegania”:

```

d1 := a1;  f1 := e1;
for i := 2 to n do
begin
  l := bi/ai-1;
  di := ai - l * ci-1;
  fi := ei - l * fi-1
end;
x1* := fn;
for i := n - 1 downto 1 do
xi* := (fi - ci * xi+1*)/di.

```

Okazuje się, że jeśli macierz A ma dominującą przekątną, tzn.

$$|a_i| > |b_i| + |c_i|, \quad 1 \leq i \leq n, \quad (3.3)$$

($b_1 = 0 = c_n$) to algorytm przeganiań jest wykonalny bez przestawień wierszy. Ponadto wymaga on $7n$ operacji arytmetycznych, a więc jest prawie optymalny.

U. 3.7 Innym ważnym przykładem macierzy szczególnej postaci są macierze symetryczne i dodatnio określone. Są to macierze spełniające $A = A^T$ oraz

$$\vec{x}^T A \vec{x} > 0 \quad \text{dla} \quad \vec{x} \neq 0.$$

Dla takich macierzy można nieco zmniejszyć koszt kombinatoryczny i zużycie pamięci przeprowadzając eliminację tak, aby otrzymać rozkład

$$A = L D L^T$$

zamiast $PA = LU$, przy czym L jest tu jak zwykle macierzą trójkątną dolną z jedynkami na przekątnej, a D jest macierzą diagonalną z dodatnimi elementami na diagonalu.

Rozkład taki przeprowadzamy mnożąc macierz A przez znane już macierze eliminacji nie tylko z lewej, ale też i z prawej strony, bez przestawień wierszy. Bez zmniejszenia ogólności rozpatrzmy tylko pierwszy krok. W tym celu, zauważmy najpierw, że $a_{1,1} = \vec{e}_1^T A \vec{e}_1 > 0$ (gdzie \vec{e}_1 jest pierwszym wektorem bazy kanonicznej), a więc nie musimy przestawiać wierszy, bo element na diagonalu jest niezerowy. W pierwszym kroku mnożymy macierz A z lewej strony przez odpowiednią macierz L_1 , a potem z prawej przez L_1^T . Kluczem do zrozumienia algorytmu jest uwaga, że efektem mnożenia macierzy $L_1 A$ z prawej strony przez L_1^T jest wyzerowanie elementów pierwszego wiersza poza $a_{1,1}$ i pozostawienie niezmiennych pozostałych elementów. Ponadto macierz $A^{(1)} = L_1 A L_1^T$ jest symetryczna i dodatnio określona. Rzeczywiście,

$$\left(A^{(1)}\right)^T = \left(L_1 A L_1^T\right)^T = \left(L_1^T\right)^T A^T L_1^T = L_1 A L_1^T,$$

oraz dla $\vec{x} \neq 0$

$$\vec{x}^T A^{(1)} \vec{x} = \vec{x}^T L_1 A L_1^T \vec{x} = \left(L_1^T \vec{x}\right)^T A \left(L_1^T \vec{x}\right) > 0,$$

bo $\vec{x} \neq 0$ implikuje $L_1^T \vec{x} \neq 0$. Stąd $a_{2,2}^{(1)} = \vec{e}_2^T A^{(1)} \vec{e}_2 > 0$. Postępując tak dalej otrzymujemy

$$L_{n-1} L_{n-2} \cdots L_2 L_1 A L_1^T L_2^T \cdots L_{n-2}^T L_{n-1}^T = D,$$

przy czym macierz D jest diagonalna i dodatnio określona, a więc wyrazy na diagonalu są dodatnie. Oznaczając

$$L = L_1^{-1} L_2^{-1} \cdots L_{n-1}^{-1}$$

dostajemy żądany rozkład.

Zauważmy, że przy praktycznej realizacji rozkładu $A = LDL^T$ wystarczy modyfikować jedynie wyrazy pod i na głównej przekątnej macierzy wyjściowej, ponieważ, jak zauważyliśmy, wszystkie kolejne macierze $A^{(k)}$ są symetryczne. Pozwala to zmniejszyć koszt kombinatoryczny o połowę do $\frac{1}{3}n^3$ operacji arytmetycznych. Opisany sposób rozkładu macierzy $A = A^T > 0$ nosi nazwę metody Banachiewicza-Choleskiego. (Zob. Ćw. 3.11.)

U. 3.8 Rozkład trójkątno-trójkątny macierzy może być zastosowany również do policzenia wyznacznika macierzy A . Jeśli bowiem $PA = LU$ to

$$\det A = (-1)^s u_{1,1} u_{2,2} \cdots u_{n,n},$$

gdzie s jest liczbą przestawień wierszy w eliminacji.

Ćwiczenia

Ćw. 3.1 Stosując (dosłownie) algorytm eliminacji Gaussa z wyborem elementu głównego w kolumnie dokonaj rozkładu macierzy

$$A = \begin{pmatrix} -1 & 1 & 0 & -3 \\ 1 & 0 & 3 & 1 \\ 0 & 1 & -1 & -1 \\ 3 & 0 & 1 & 2 \end{pmatrix}$$

na iloczyn $PA = LR$, gdzie P jest macierzą permutacji, L macierzą trójkątną dolną z jedynekami na głównej przekątnej, a R macierzą trójkątną górną.

Ćw. 3.2 Niech

$$A = \begin{pmatrix} 4 & -2 & 0 & -2 \\ -2 & 2 & 1 & 1 \\ 0 & 1 & 5 & 2 \\ -2 & 1 & 2 & 3 \end{pmatrix}, \quad \vec{b} = \begin{pmatrix} 8 \\ -3 \\ 13 \\ 2 \end{pmatrix}.$$

Znajdź macierz trójkątną dolną L i diagonalną D , takie że $A = LDL^T$. Korzystając z tego rozkładu rozwiąż układ równań $A\vec{x} = \vec{b}$.

Ćw. 3.3 Pokaż, że jeśli algorytm rozwiązywania układu równań z macierzą pełną (lub trójkątną) nie wykorzystuje wszystkich współczynników macierzy i/lub wektora, to istnieją macierz A i wektor \vec{b} takie, że algorytm dla tych danych nie daje dokładnego rozwiązania w arytmetyce idealnej.

Ćw. 3.4 Znajdź macierz odwrotną do macierzy trójkątnej.

Ćw. 3.5 Pokaż, że iloczyn elementarnych macierzy permutacji jest macierzą, która w każdym wierszu i w każdej kolumnie ma dokładnie jedną jedynekę. I odwrotnie, jeśli pewna macierz ma w każdym wierszu i kolumnie dokładnie jedną jedynekę to jest ona iloczynem pewnej liczby elementarnych macierzy permutacji.

Ćw. 3.6 Pokaż, że dla danej macierzy A i permutacji P istnieje co najwyżej jeden rozkład $PA = LU$ na macierz trójkątną dolną L z jedynekami na przekątnej i na macierz trójkątną górną U .

Ćw. 3.7 Pokaż, że w algorytmie eliminacji Gaussa (bez przestawień) elementy $l_{i,j}$ i $u_{i,j}$ obliczane są według wzorów

$$l_{i,j} = \left(a_{i,j} - \sum_{k=1}^{j-1} l_{i,k} u_{k,j} \right) / u_{j,j}, \quad j = 1, 2, \dots, i-1,$$

$$u_{i,j} = a_{i,j} - \sum_{k=1}^{i-1} l_{i,k} u_{k,j}, \quad j = i, i+1, \dots, n,$$

dla $i = 1, 2, \dots, n$. Wywnioskuj stąd rozkład $A = LU$ z Twierdzenia 3.1.

Ćw. 3.8 Pokaż, że przy spełnieniu warunku (3.3) algorytm “przeganiania” z U. 3.6 jest wykonalny bez przestawień wierszy. Opracuj algorytm rozwiązywania układu z macierzą trójdziagonalną nie spełniającą warunku (3.3).

Ćw. 3.9 Pokaż, że jeśli macierz pełna A ma dominującą przekątną, tzn.

$$2|a_{i,i}| > \sum_{j=1}^n |a_{i,j}|, \quad 1 \leq i \leq n,$$

to jest to macierz nieosobliwa. Ponadto, eliminacja Gaussa jest dla takiej macierzy wykonalna bez przestawień wierszy.

Ćw. 3.10 Opracuj algorytm eliminacyjny dla rozwiązywania układów liniowych z macierzą *Hessenberga*, czyli macierzą dla której $a_{i,j} = 0$, o ile $i \geq j + 2$.

Ćw. 3.11 Opracuj program rozkładający macierz symetryczną i dodatnio określoną A na iloczyn $A = LDL^T$ według metody Banachiewicza-Choleskiego z U. 3.7. Obliczenia przeprowadź w tej samej macierzy A pozostawiając elementy na i nad diagonalą nie zmienione i otrzymując w wyniku macierz L pod główną diagonalą macierzy A , a macierz D jako dodatkowy wektor.

Rozdział 4

Analiza błędów w eliminacji Gaussa

Zastanowimy się teraz, jak przebiega w arytmetyce fl_ν realizacja algorytmu eliminacji Gaussa z wyborem elementu głównego w kolumnie.

4.1 Układy z macierzą trójkątną

Najpierw pokażemy, że algorytmy rozwiązywania układów trójkątnych zaprezentowane w Rozdziale 3.1 są numerycznie poprawne. Rozpatrzmy najpierw układ trójkątny górny, $U\vec{x} = \vec{c}$. Wykorzystując fakt, że liczenie iloczynu skalarnego jest numerycznie poprawne, mamy

$$fl_\nu(x_n^*) = \frac{c_n(1 + \beta_n)}{u_{n,n}(1 + \alpha_{n,n})}(1 + \delta_n) = \frac{c_n^\nu}{u_{n,n}^\nu}$$

i dla $i = n - 1, n - 2, \dots, 1$,

$$\begin{aligned} fl_\nu(x_i^*) &= \frac{\left(c_i(1 + \beta_i) - \sum_{j=i+1}^n u_{i,j}(1 + \alpha_{i,j})fl_\nu(x_j^*)(1 + \gamma_{i,j})\right)(1 + \omega_i)}{u_{i,i}(1 + \alpha_{i,i})(1 + \delta_i)^{-1}} \\ &= \frac{c_i^\nu - \sum_{j=i+1}^n u_{i,j}^\nu fl_\nu(x_j^*)}{u_{i,i}^\nu}, \end{aligned}$$

gdzie $c_i^\nu = c_i(1 + \beta_i)$, $u_{i,i}^\nu = u_{i,i}(1 + \alpha_{i,i})(1 + \delta_i)^{-1}(1 + \omega_i)^{-1}$ ($\omega_n = 0$), $u_{i,j}^\nu = u_{i,j}(1 + \alpha_{i,j})(1 + \gamma_{i,j})$, przy czym β_i i $\alpha_{i,j}$ pochodzą z błędów reprezentacji, δ_i z dzielenia, ω_i z odejmowania, a $\gamma_{i,j}$ z liczenia sum. Stąd $|c_i^\nu - c_i| \leq \nu|c_i|$ i $|u_{i,j}^\nu - u_{i,j}| \lesssim (n+1)\nu|u_{i,j}|$. Stąd otrzymane w fl_ν rozwiązanie jest dokładnym rozwiązaniem dla danych $U^\nu = (u_{i,j}^\nu)_{i,j}$ i $\vec{c}^\nu = (c_j^\nu)_j$, przy czym

$$\|U^\nu - U\|_\infty \lesssim (n+1)\nu\|U\|_\infty \quad \text{i} \quad \|\vec{c}^\nu - \vec{c}\|_\infty \lesssim \nu\|\vec{c}\|_\infty.$$

Zauważmy, że macierz zaburzona U^ν jest nieosobliwa. Powyższy algorytm jest więc numerycznie poprawny w klasie nieosobliwych macierzy trójkątnych górnych. Zauważmy również, że jeśli wektor prawej strony \vec{c} jest reprezentowany dokładnie to $\vec{c}^\nu = \vec{c}$. Fakt ten wykorzystamy później.

Przeprowadzając podobną analizę dla układu $L\vec{x} = \vec{c}$ z macierzą trójkątną dolną z jedynekami na przekątnej dostajemy, że otrzymane w fl_ν rozwiązanie jest dokładnym rozwiązaniem układu $L^\nu\vec{x} = \vec{c}^\nu$, gdzie

$$\|L^\nu - L\|_\infty \lesssim (n+1)\nu\|L\|_\infty \quad \text{i} \quad \|\vec{c}^\nu - \vec{c}\|_\infty \lesssim \nu\|\vec{c}\|_\infty.$$

4.2 Poprawność rozkładu trójkątno-trójkątnego

Zajmiemy się teraz jakością numeryczną rozkładu macierzy na iloczyn trójkątno-trójkątny $PA = LU$. W poniższym twierdzeniu norma macierzy jest dowolna, ale ustalona.

Twierdzenie 4.1 *Niech $A \in \mathbb{R}^{n \times n}$ będzie macierzą nieosobliwą. Dla dostatecznie silnej arytmetyki (tzn. dla dostatecznie małego ν) rozkład trójkątno-trójkątny macierzy za pomocą eliminacji Gaussa z wyborem elementu głównego w kolumnie jest wykonalny w arytmetyce fl_ν . Otrzymane w wyniku macierze permutacji P^ν , trójkątna górna U^ν i trójkątna dolna L^ν (z jedynkami na przekątnej i elementami co do modułu nie większymi od jednośc) spełniają równość*

$$P^\nu (A + E) = L^\nu U^\nu,$$

gdzie

$$\|E\| \leq K(n) \nu \|A\|,$$

a $K(n)$ jest pewnym czynnikiem niezależnym od ν i A .

Dowód. Załóżmy najpierw dla uproszczenia, że eliminacja jest w fl_ν wykonalna i że nie musimy dokonywać przestawień wierszy, tzn. w każdym kroku eliminacji element główny znajduje się na głównej diagonalu. Dla uniknięcia nadmiaru indeksów będziemy oznaczać przez \tilde{w} wartość w obliczoną w fl_ν . W szczególności, $\tilde{A}^{(0)}$ jest reprezentacją macierzy $A^{(0)} = A$, a $\tilde{A}^{(k)}$ jest obliczoną macierzą $A^{(k)}$. Oczywiście, $\tilde{a}_{i,j}^{(0)} = a_{i,j} + \varepsilon_{i,j}^{(0)}$, gdzie $|\varepsilon_{i,j}| \leq \nu |a_{i,j}|$. W pierwszym kroku eliminacji dostajemy

$$\tilde{l}_{i,1} = \frac{\tilde{a}_{i,1}^{(0)}}{\tilde{a}_{1,1}^{(0)}} (1 + \delta_i), \quad |\delta_i| \leq \nu,$$

(zauważmy, że $|\tilde{l}_{i,1}| \leq 1$), a następnie dla $i, j \geq 2$

$$\tilde{a}_{i,j}^{(1)} = \tilde{a}_{i,j}^{(0)} (1 + \alpha_{i,j}^{(1)}) - \tilde{l}_{i,1} \tilde{a}_{1,j}^{(0)} (1 + \beta_{i,j}^{(1)}), \quad |\alpha_{i,j}^{(1)}| \leq \nu, \quad |\beta_{i,j}^{(1)}| \lesssim 2\nu.$$

Stąd

$$\tilde{a}_{i,j}^{(1)} = \tilde{a}_{i,j}^{(0)} - \tilde{l}_{i,1} \tilde{a}_{1,j}^{(0)} + \varepsilon_{i,j}^{(1)}, \quad |\varepsilon_{i,j}^{(1)}| \lesssim 2\nu (|\tilde{a}_{i,j}^{(0)}| + |\tilde{a}_{1,j}^{(0)}|).$$

Postępując indukcyjnie dla $k = 2, 3, \dots, n-1$ otrzymujemy w ogólności dla $i, j \geq k+1$, że

$$\tilde{a}_{i,j}^{(k)} = \tilde{a}_{i,j}^{(k-1)} - \tilde{l}_{i,k} \tilde{a}_{k,j}^{(k-1)} + \varepsilon_{i,j}^{(k)}, \quad |\varepsilon_{i,j}^{(k)}| \lesssim 2\nu (|\tilde{a}_{i,j}^{(k-1)}| + |\tilde{a}_{k,j}^{(k-1)}|),$$

przy czym $|\tilde{l}_{i,k}| \leq 1$. Wprowadźmy macierze zaburzeń $E^{(k)} = (\varepsilon_{i,j}^{(k)})_{i,j=1}^n$, gdzie $\varepsilon_{i,j}^{(k)} = 0$ gdy $i \leq k$ lub $j \leq k$. Kolejne kroki eliminacji Gaussa w fl_ν przedstawiają się w postaci macierzowej następująco:

$$\begin{aligned} \tilde{A}^{(0)} &= A^{(0)} + E^{(0)}, \\ \tilde{A}^{(1)} &= \tilde{L}_1 \tilde{A}^{(0)} + E^{(1)} = \tilde{L}_1 A^{(0)} + \tilde{L}_1 E^{(0)} + E^{(1)}, \\ \tilde{A}^{(2)} &= \tilde{L}_2 \tilde{A}^{(1)} + E^{(2)} = \tilde{L}_2 \tilde{L}_1 A^{(0)} + \tilde{L}_2 \tilde{L}_1 E^{(0)} + \tilde{L}_2 E^{(1)} + E^{(2)}, \\ &\dots \\ \tilde{A}^{(n-1)} &= \tilde{L}_{n-1} \tilde{A}^{(n-2)} + E^{(n-1)} \\ &= \tilde{L}_{n-1} \tilde{L}_{n-2} \dots \tilde{L}_1 A^{(0)} + \tilde{L}_{n-1} \tilde{L}_{n-2} \dots \tilde{L}_1 E^{(0)} + \dots \\ &\quad + \tilde{L}_{n-1} \tilde{L}_{n-2} E^{(n-3)} + \tilde{L}_{n-1} E^{(n-2)} + E^{(n-1)}, \end{aligned}$$

przy czym $\tilde{A}^{(n-1)} = \tilde{U}$. Mnożąc obie strony ostatniej równości przez $\tilde{L} = \tilde{L}_1^{-1}\tilde{L}_2^{-1}\cdots\tilde{L}_{n-1}^{-1}$ dostajemy rozkład trójkątno-trójkątny

$$\tilde{L}\tilde{U} = A + E,$$

gdzie

$$E = E^{(0)} + \tilde{L}_1^{-1}E^{(1)} + \tilde{L}_1^{-1}\tilde{L}_2^{-1}E^{(2)} + \cdots + \tilde{L}_1^{-1}\tilde{L}_2^{-1}\cdots\tilde{L}_{n-1}^{-1}E^{(n-1)}.$$

Zauważmy teraz, że dla każdego składnika powyższej sumy mamy $\tilde{L}_1^{-1}\tilde{L}_2^{-1}\cdots\tilde{L}_k^{-1}E^{(k)} = E^{(k)}$, ponieważ wyrazy $e_{i,j}^{(k)}$ macierzy $E^{(k)}$ zerują się jeśli $i \leq k$ lub $j \leq k$, a stąd

$$E = E^{(0)} + E^{(1)} + \cdots + E^{(n-1)}.$$

Pozostaje oszacować normę macierzy E . W tym celu, oszacujemy najpierw maksymalny element kolejnych macierzy $\tilde{A}^{(k)}$. Niech

$$N = \max_{1 \leq i,j \leq n} |a_{i,j}| \leq \|A\|_\infty.$$

Wtedy $|\tilde{a}_{i,j}^{(0)}| \lesssim N$,

$$|\tilde{a}_{i,j}^{(1)}| \lesssim |\tilde{a}_{i,j}^{(0)}| + |\tilde{a}_{1,j}^{(0)}| \approx |a_{i,j}^{(0)}| + |a_{1,j}^{(0)}| \leq 2N,$$

i postępując tak indukcyjnie dla $k = 2, 3, \dots, n-1$ mamy w ogólności, że

$$|\tilde{a}_{i,j}^{(k)}| \lesssim |\tilde{a}_{i,j}^{(k-1)}| + |\tilde{a}_{k,j}^{(k-1)}| \approx |a_{i,j}^{(k-1)}| + |a_{k,j}^{(k-1)}| \lesssim 2^k N.$$

Stąd $|\varepsilon_{i,j}^{(k)}| \lesssim 2^{k+1}\nu N$, co implikuje $\|E^{(k)}\|_\infty \leq 2^{k+1}n\nu N$, i ostatecznie

$$\|E\|_\infty \leq \sum_{k=0}^{n-1} \|E^{(k)}\|_\infty \lesssim n\nu N \sum_{k=0}^{n-1} 2^{k+1} \leq 2^{n+1}n\nu \|A\|_\infty.$$

Czynnik $K(n)$ w treści twierdzenia jest więc na poziomie $2^{n+1}n$. Z równoważności norm w $\mathbb{R}^{n \times n}$ wynika, że podobna nierówność zachodzi dla dowolnej, innej niż $\|\cdot\|_\infty$ normy, ale być może z inną wartością $K(n)$.

Oczywiście, ewentualne permutacje wierszy macierzy nie zmieniają tych oszacowań, bo nie wywołują błędów w fl_ν .

Aby zakończyć dowód, trzeba jeszcze pokazać, że dla dostatecznie małych ν rozkład jest wykonalny, co jest równoważne temu, że macierz $(A + E)$ jest nieosobliwa. Rzeczywiście, weźmy $\nu > 0$ spełniające

$$\nu K(n) \|A\|_\infty \|A^{-1}\|_\infty < 1. \quad (4.1)$$

Jeśliby macierz $(A + E)$ była osobliwa to dla pewnego niezerowego wektora \vec{x} mielibyśmy $(A + E)\vec{x} = A(I + A^{-1}E)\vec{x} = 0$, a w konsekwencji $\|A^{-1}E\vec{x}\|_\infty = \|\vec{x}\|_\infty$ i

$$1 \leq \|A^{-1}E\|_\infty \leq \|A^{-1}\|_\infty \|E\|_\infty \leq K(n)\nu \|A\|_\infty \|A^{-1}\|_\infty,$$

co przeczy (4.1). \square

Zauważmy, że warunek (4.1) na wykonalność rozkładu w fl_ν zależy od macierzy A poprzez

$$\text{cond}(A) = \|A\| \|A^{-1}\|.$$

Wielkość tę nazywa się *uwarunkowaniem macierzy*. Z Twierdzenia 4.1 oraz warunku (4.1) wynika ważny wniosek o numerycznej poprawności rozkładu w klasie macierzy o uwarunkowaniu wspólnie ograniczonym przez pewną stałą.

Wniosek 4.1 Niech $M > 0$. W klasie macierzy nieosobliwych A formatu $n \times n$ takich, że

$$\text{cond}(A) = \|A\| \|A^{-1}\| \leq M$$

rozkład $PA = LR$ jest numerycznie poprawny. Dokładniej, dla $\nu < 1/(MK(n))$ rozkład w \mathcal{fl}_ν jest wykonalny, a otrzymane macierze P^ν , L^ν i U^ν pochodzą z dokładnego rozkładu macierzy $A + E$, gdzie $\|E\| \leq K(n)\nu\|A\|$. \square

4.3 Poprawność rozwiązywania układu

Przypomnijmy, że algorytm eliminacji Gaussa rozwiązywania układów równań $A\vec{x} = \vec{b}$ polega na rozkładzie macierzy na iloczyn $PA = LU$, a następnie na rozwiązaniu dwóch układów trójkątnych (albo, równoważnie, na sprowadzeniu układu do postaci trójkątnej i rozwiązaniu go). Okazuje się, że numeryczna poprawność każdego z tych etapów algorytmu daje nam również numeryczną poprawność całego algorytmu.

Twierdzenie 4.2 Niech $M > 0$. Algorytm eliminacji Gaussa z wyborem elementu głównego w kolumnie zastosowany do rozwiązywania układu równań

$$A\vec{x} = \vec{b}$$

jest numerycznie poprawny w klasie danych wektorów $\vec{b} \in \mathbb{R}^n$ oraz macierzy nieosobliwych $A \in \mathbb{R}^{n \times n}$ spełniających

$$\text{cond}(A) \leq M.$$

Dowód. Dla dostatecznie silnej arytmetyki, rozkład macierzy A daje w \mathcal{fl}_ν rozkład macierzy zaburzonej $P_1(A + E_1) = L_1 U_1$, gdzie $\|E_1\| \leq K_1\nu\|A\|$. W drugim kroku, otrzymane w \mathcal{fl}_ν rozwiązanie \vec{y}_1 układu $L_1\vec{y} = P_1\vec{b}$ jest dokładnym rozwiązaniem układu $(L_1 + E_2)\vec{y} = P_1(\vec{b} + \vec{e})$, gdzie $\|E_2\| \leq K_2\nu\|L_1\|$ oraz $\|\vec{e}\| \leq K_3\nu\|\vec{b}\|$. Z kolei w ostatnim kroku algorytmu, rozwiązując układ $U_1\vec{x} = \vec{y}_1$ dostajemy w \mathcal{fl}_ν rozwiązanie \vec{x}^ν , które jest dokładnym rozwiązaniem układu $(U_1 + E_3)\vec{x} = \vec{y}_1$, gdzie $\|E_3\| \leq K_3\nu\|U_1\|$. (Zauważmy, że wektor \vec{y}_1 nie jest zaburzony, bo jest on reprezentowany dokładnie, zob. Rozdział 4.1). Stąd otrzymane w \mathcal{fl}_ν rozwiązanie \vec{x}^ν jest dokładnym rozwiązaniem układu

$$A^\nu\vec{x} = \vec{b}^\nu,$$

gdzie $\vec{b}^\nu = \vec{b} + \vec{e}$, a $A^\nu = A + E$ spełnia równanie

$$P_1(A + E) = (L_1 + E_2)(U_1 + E_3).$$

Aby zakończyć dowód, należy teraz oszacować $\|E\|$. Ponieważ $P_1(A + E_1) = L_1 U_1$, mamy

$$P_1 E = P_1 E_1 + L_1 E_3 + E_2 U_1 + E_2 E_3.$$

Stąd

$$\begin{aligned} \|E\|_\infty &\leq \|E_1\|_\infty + \|L_1\|_\infty \|E_3\|_\infty + \|E_2\|_\infty \|U_1\|_\infty + \|E_2\|_\infty \|E_3\|_\infty \\ &\lesssim \nu K_1 \|A\|_\infty + \|L_1\|_\infty \nu K_3 \|U_1\|_\infty + \|U_1\|_\infty \nu K_2 \|L_1\|_\infty \\ &= \nu \left(K_1 \|A\|_\infty + (K_2 + K_3) \|L_1\|_\infty \|U_1\|_\infty \right). \end{aligned}$$

Ponieważ elementy macierzy L_1 są co do modułu nie większe od jedności, mamy $\|L_1\|_\infty \leq n$. Jak zauważyliśmy wcześniej, mamy też $\|U_1\|_\infty \leq n2^{n-1}\|A\|_\infty$. Przyjmując $K = K_1 + (K_2 + K_3)n^22^{n-1}$ ostatecznie otrzymujemy

$$\|E\|_\infty \leq K\nu\|A\|_\infty,$$

czyli numeryczną poprawność, ponieważ K nie zależy od ν i A . \square

4.4 Uwarunkowanie macierzy, a błąd w fl_ν

Pokazaliśmy, że eliminacja Gaussa jest numerycznie poprawna w klasie macierzy $A \in \mathbb{R}^{n \times n}$ takich, że

$$\text{cond}(A) \leq M,$$

gdzie $M < \infty$ jest dowolna. Okazuje się, że wielkość uwarunkowania macierzy, $\text{cond}(A)$, ma też zasadniczy wpływ na uwarunkowanie zadania rozwiązywania układu równań, a tym samym także na błąd wytworzony w fl_ν przy realizacji algorytmu eliminacji Gaussa. Rzeczywiście, mamy bowiem następujące twierdzenie. (Poniżej norma wektorowa jest dowolna, ale ustalona, a norma macierzowa jest przez nią indukowana, zob. U. 1.3.)

Twierdzenie 4.3 Niech E i \vec{e} będą zaburzeniami odpowiednio macierzy A i wektora \vec{b} takimi, że

$$\|E\| \leq K_1\nu\|A\| \quad i \quad \|\vec{e}\| \leq K_2\nu\|\vec{b}\|,$$

Jeśli

$$K_1\nu\text{cond}(A) < 1$$

to układ zaburzony $(A + E)\vec{x} = (\vec{b} + \vec{e})$ ma jednoznaczne rozwiązanie \vec{z}^* spełniające

$$\|\vec{z}^* - \vec{x}^*\| \leq (K_1 + K_2)\nu \frac{\text{cond}(A)}{1 - K_1\nu\text{cond}(A)} \|\vec{x}^*\|.$$

Dowód. Zauważmy najpierw, że jeśli F jest macierzą taką, że $\|F\| < 1$ to macierz $(I - F)$ (gdzie I jest macierzą identycznościową) jest nieosobliwa oraz

$$\|(I - F)^{-1}\| \leq \frac{1}{1 - \|F\|}. \quad (4.2)$$

Rzeczywiście, gdyby $(I - F)$ była osobliwa to istniałby niezerowy wektor \vec{x} taki, że $(I - F)\vec{x} = 0$, co implikuje $\|F\vec{x}\|/\|\vec{x}\| = 1$ i w konsekwencji $\|F\| \geq 1$. Aby pokazać (4.2) zauważmy, że

$$1 = \|I\| = \|(I - F)(I - F)^{-1}\| \geq \|(I - F)^{-1}\| - \|F\| \|(I - F)^{-1}\| = (1 - \|F\|) \|(I - F)^{-1}\|,$$

skąd bezpośrednio wynika (4.2).

Po podstawieniu $F = -A^{-1}E$ mamy teraz

$$\|F\| \leq \|A^{-1}\| \|E\| \leq K_1\nu\|A\| \|A^{-1}\| < 1,$$

co wobec równości $A + E = A(I + A^{-1}E)$ daje, że macierz $(A + E)$ jest nieosobliwa i układ zaburzony ma jednoznaczne rozwiązanie \vec{z}^* . Przedstawmy to rozwiązanie w postaci $\vec{z}^* = \vec{x}^* + (\vec{z}^* - \vec{x}^*)$. Rozpisując

układ zaburzony i wykorzystując równość $A\vec{x}^* = \vec{b}$ otrzymujemy, że $(A + E)(\vec{z}^* - \vec{x}^*) = \vec{e} - E\vec{x}^*$, czyli

$$\vec{z}^* - \vec{x}^* = (I + A^{-1}E)^{-1}A^{-1}(\vec{e} - E\vec{x}^*),$$

a stąd

$$\begin{aligned} \|\vec{z}^* - \vec{x}^*\| &\leq \|(I + A^{-1}E)^{-1}\| \|A^{-1}\| (\|\vec{e}\| + \|E\| \|\vec{x}^*\|) \\ &\leq \frac{\|A^{-1}\|}{1 - K_1\nu\|A\| \|A^{-1}\|} (K_2\nu\|\vec{b}\| + K_1\nu\|A\| \|\vec{x}^*\|) \\ &\leq \frac{\|A\| \|A^{-1}\|}{1 - K_1\nu\|A\| \|A^{-1}\|} (K_1 + K_2)\nu \|\vec{x}^*\|, \end{aligned}$$

co kończy dowód. \square

Podsumowując Twierdzenia 4.2 i 4.3 otrzymujemy wniosek, który jest końcowym wynikiem tego rozdziału.

Wniosek 4.2 *Niech A będzie macierzą nieosobliwą. Dla dostatecznie silnej arytmetyki,*

$$\nu \overline{K}(n) \text{cond}(A) \ll 1$$

(gdzie $\overline{K}(n)$ jest pewnym czynnikiem niezależnym od A i ν), eliminacja Gaussa z wyborem elementu głównego w kolumnie zastosowana do rozwiązania układu $A\vec{x} = \vec{b}$ jest w fl_ν wykonalna i daje wynik $fl_\nu(\vec{x}^*)$ spełniający nierówność

$$\|fl_\nu(\vec{x}^*) - \vec{x}^*\| \lesssim \overline{K}(n)\nu \text{cond}(A) \|\vec{x}^*\|. \quad \square$$

Uwagi i uzupełnienia

U. 4.1 Uwarunkowanie macierzy może być bardzo duże. Na przykład, dla macierzy Hilberta $H = (h_{i,j})_{i,j=1}^n$,

$$h_{i,j} = \frac{1}{i+j-1},$$

już dla $n = 16$ uwarunkowanie w normie drugiej jest rzędu 10^{17} .

U. 4.2 Jak zauważyliśmy w dowodzie Twierdzenia 4.1 stała kumulacji $K(n)$ w eliminacji Gaussa z wyborem elementu głównego w kolumnie zależy zasadniczo od wzrostu maksymalnego elementu w macierzach $A^{(k)}$ powstających w kolejnych krokach eliminacji. Okazuje się, że uzyskane pesymistyczne oszacowanie jest praktycznie niespotykane, chociaż teoretycznie może być osiągnięte. Przykładem jest macierz

$$A = \begin{pmatrix} 1 & 0 & 0 & \cdots & 0 & 1 \\ -1 & 1 & 0 & & \vdots & 1 \\ -1 & -1 & 1 & & \vdots & 1 \\ & & & \ddots & 0 & \vdots \\ & & & & 1 & 1 \\ -1 & -1 & -1 & \cdots & -1 & 1 \end{pmatrix},$$

dla której $K(n) = 2^{n-1}$. Dodajmy, że uwarunkowanie tej macierzy jest dobre, np. nawet dla $n = 64$ uwarunkowanie w normie drugiej wynosi około 28.6.

U. 4.3 Uważna analiza dowodu Twierdzenia 4.1 pokazuje, że rozkład trójkątno-trójkątny jest numerycznie poprawny bez przestawień wierszy/kolumn gdy element maksymalny w kolejnych macierzach $A^{(k)}$ wzrasta w sposób niezależny od A . Ma to miejsce np. wtedy gdy A jest symetryczna i dodatnio określona, albo ma dominującą główną przekątną, zob. U. 4.4 oraz Ćw. 4.6 i 4.7.

U. 4.4 Dla macierzy symetrycznych i dodatnio określonych, $A = A^T > 0$, eliminacja Gaussa jest wykonalna bez przestawień wierszy, zob. U. 3.7. W klasie tych macierzy eliminacja bez przestawień wierszy jest też numerycznie poprawna, a więc w szczególności algorytm Banachiewicza-Choleskiego jest numerycznie poprawny. Wykażemy to pokazując, że maksymalny element w kolejnych macierzach $A^{(k)}$ wzrasta co najwyżej dwukrotnie. W tym celu wykorzystamy fakt, że dla dowolnej symetrycznej i dodatnio określonej macierzy $B = (b_{i,j})_{i,j=1}^n$ mamy $b_{i,i} > 0$ oraz spełniona jest nierówność

$$b_{i,j}^2 < b_{i,i}b_{j,j}, \quad 1 \leq i, j \leq n. \quad (4.3)$$

(Zob. Ćw. 4.1). Jak wykazaliśmy w U. 3.7, każda z macierzy $A^{(k)}$ jest symetryczna i dodatnio określona, Stąd

$$\begin{aligned} |a_{i,j}^{(k)}| &= |a_{i,j}^{(k-1)} - l_{i,k}a_{k,j}^{(k-1)}| \leq |a_{i,j}^{(k-1)}| + \frac{|a_{i,k}^{(k-1)}|}{|a_{k,k}^{(k-1)}|} |a_{k,j}^{(k-1)}| \\ &\leq \sqrt{a_{i,i}^{(k-1)}a_{j,j}^{(k-1)}} + \frac{\sqrt{a_{i,i}^{(k-1)}a_{k,k}^{(k-1)}}\sqrt{a_{k,k}^{(k-1)}a_{j,j}^{(k-1)}}}{a_{k,k}^{(k-1)}} \\ &= 2\sqrt{a_{i,i}^{(k-1)}a_{j,j}^{(k-1)}} \leq 2\max(a_{i,i}^{(k-1)}, a_{j,j}^{(k-1)}). \end{aligned}$$

Ćwiczenia

Ćw. 4.1 Wykaż, że macierz symetryczna 2×2

$$\begin{pmatrix} a & c \\ c & b \end{pmatrix}$$

jest dodatnio określona wtedy i tylko wtedy, gdy $a > 0$ i $ab > c^2$. Wywnioskuj stąd nierówność (4.3) dla dowolnych macierzy symetrycznych i dodatnio określonych. Ponadto, największy co do modułu element takiej macierzy leży na głównej diagonalnej.

Ćw. 4.2 Wykaż, że macierz A jest dodatnio określona wtedy i tylko wtedy gdy dla każdego $\vec{x} \in \mathbb{R}^n$ wektory $A\vec{x}$ i \vec{x} tworzą w \mathbb{R}^n kąt ostry.

Ćw. 4.3 Zaproponuj algorytm rozwiązywania układu równań liniowych $A\vec{x} = \vec{b}$ z macierzą nieosobliwą $A = (a_{i,j}) \in \mathbb{R}^{n \times n}$ taką, że $a_{i,j} = 0$ dla $|i - j| \geq 2$, $1 \leq i \leq n - 1$, $1 \leq j \leq n$. (Zauważ, że ostatni wiersz jest w ogólności pełny.) Algorytm ma działać w czasie liniowym w n i być numerycznie poprawny.

Ćw. 4.4 Pokaż, że jeśli eliminację Gaussa z wyborem elementu głównego w kolumnie zastosujemy do macierzy trójdzielnej, to wzrost elementu maksymalnego macierzy nie będzie zależał od n . Dokładniej,

$$\max_{i,j,k} |a_{i,j}^{(k)}| \leq 2 \max_{i,j} |a_{i,j}|.$$

Ćw. 4.5 Pokaż, że dla macierzy Hessenberga ($a_{i,j} = 0$ dla $i \geq j + 2$) eliminacja Gaussa z wyborem elementu głównego w kolumnie daje

$$\max_{i,j,k} |a_{i,j}^{(k)}| \leq (k + 1) \max_{i,j} |a_{i,j}|.$$

Ćw. 4.6 Pokaż numeryczną poprawność algorytmu przegania z U. 3.6 przy założeniu, że macierz trójkątna ma dominującą przekątną.

Ćw. 4.7 Wykaż numeryczną poprawność eliminacji Gaussa bez przestawień wierszy dla macierzy z dominującą wierszowo przekątną, tzn. gdy

$$2|a_{i,i}| > \sum_{j=1}^n |a_{i,j}|, \quad 1 \leq i \leq n.$$

Wskazówka. Zauważ, że $\max_{i,j} |a_{i,j}^{(k)}| \leq 2 \max_{i,j} |a_{i,j}^{(k-1)}|$.

Ćw. 4.8 Jeśli

$$(A + E)\vec{z} = \vec{b}, \quad (4.4)$$

gdzie $\|E\|_p \leq K\nu\|A\|_p$, to oczywiście dla residuum $\vec{r} = \vec{b} - A\vec{z}$ mamy

$$\|\vec{r}\|_p \leq K\nu\|A\|_p\|\vec{z}\|_p. \quad (4.5)$$

Pokaż, że dla $p = 1, 2, \infty$ zachodzi też twierdzenie odwrotne, tzn. jeśli spełniony jest warunek (4.5) to istnieje macierz pozornych zaburzeń E taka, że $\|E\|_p \leq K\nu\|A\|_p$ oraz spełniona jest równość (4.4).

Wskazówka. Rozpatrz $E = \vec{r}(\text{sgn } z_i)_{1 \leq i \leq n}^T / \|\vec{z}\|_1$ dla $p = 1$, $E = \vec{r}\vec{z}^T / \|\vec{z}\|_2^2$ dla $p = 2$, oraz $E = \vec{r}(\text{sgn } z_k)\vec{e}_k^T / \|\vec{z}\|_\infty$ dla $p = \infty$, gdzie k jest indeksem dla którego $|z_k| = \|\vec{z}\|_\infty$.

Ćw. 4.9 Przetestuj rozwiązywanie układu równań liniowych $H\vec{x} = \vec{b}$ z macierzą Hilberta H z U. 4.1 i wektorem $\vec{b} = H\vec{1}$, gdzie $\vec{1} = (1, 1, \dots, 1)^T$. Zaobserwuj gwałtowny wzrost błędu względnego wyniku przy wzrastającym wymiarze n . Dlaczego tak się dzieje?

Ćw. 4.10 Przetestuj rozwiązywanie układu równań liniowych $A\vec{x} = \vec{b}$ stosując eliminację Gaussa z wyborem elementu głównego w kolumnie, dla macierzy A z U. 4.2 i wektorem $\vec{b} = A\vec{1}$. Zaobserwuj i skomentuj różnicę w błędzie względnym wyniku dla $n = 36$ i $n = 72$. Czy podobne wyniki otrzymasz gdy wybór elementu głównego wykonasz w całej macierzy?

Rozdział 5

Zadanie wygładzania liniowego

W tym rozdziale zajmiemy się zadaniem wygładzania liniowego, nazywanym też często *liniowym zadaniem najmniejszych kwadratów*. Jest ono uogólnieniem zadania rozwiązywania kwadratowych układów równań liniowych do przypadku, gdy układ jest nadokreślony.

5.1 Układ normalny

Niech A będzie daną macierzą o m wierszach i n kolumnach, $A \in \mathbb{R}^{m \times n}$, taką że

$$m \geq n = \text{rank}(A),$$

albo równoważnie, taką że jej wektory kolumnowe są liniowo niezależne. Niech także dany będzie wektor $\vec{b} \in \mathbb{R}^m$. Jasne jest, że wtedy układ równań $A\vec{x} = \vec{b}$ nie zawsze ma rozwiązanie - mówimy, że układ jest *nadokreślony*.

Zadanie wygładzania liniowego polega na znalezieniu wektora $\vec{x}^* \in \mathbb{R}^n$, który najbardziej "pasuje" do równania w tym sensie, że minimalizuje wektor *residualny* $\vec{r} = \vec{b} - A\vec{x}$ w normie drugiej, tzn.

$$\|\vec{b} - A\vec{x}^*\|_2 = \min_{\vec{x} \in \mathbb{R}^n} \|\vec{b} - A\vec{x}\|_2.$$

Przykład 5.1 Przypuśćmy, że dla pewnej funkcji $f : [a, b] \rightarrow \mathbb{R}$ obserwujemy jej wartości f_i (dokładne lub zaburzone) w punktach t_i , $1 \leq i \leq m$. Funkcję tą chcielibyśmy przybliżyć inną funkcją w należącą do pewnej n wymiarowej przestrzeni liniowej W , np. przestrzeni wielomianów stopnia mniejszego niż n . Jakość przybliżenia mierzymy wielkością

$$\sum_{i=1}^m (f_i - w(t_i))^2. \tag{5.1}$$

Wybierając pewną bazę $(w_j)_{j=1}^n$ w W i rozwijając w w tej bazie, $w = \sum_{j=1}^n c_j w_j$, sprowadzamy problem do minimalizacji wyrażenia

$$\sum_{i=1}^m \left(f_i - \sum_{j=1}^n c_j w_j(t_i) \right)^2$$

względem c_j , a więc do zadania wygładzania liniowego. Rzeczywiście, kładąc $A = (a_{i,j}) \in \mathbb{R}^{m \times n}$ z $a_{i,j} = w_j(t_i)$, $\vec{b} = (f_i)_{i=1}^m$ i $\vec{x} = (c_j)_{j=1}^n$, wielkość (5.1) jest równa $\|\vec{b} - A\vec{x}\|_2^2$.

Lemat 5.1 Zadanie wygładzania liniowego ma jednoznaczne rozwiązanie \vec{x}^* , które spełnia układ równań

$$A^T A \vec{x} = A^T \vec{b}. \quad (5.2)$$

Dowód. Niech $P \subset \mathbb{R}^m$ będzie obrazem A jako odwzorowania liniowego z \mathbb{R}^n w \mathbb{R}^m ,

$$P = \{ A\vec{x} : \vec{x} \in \mathbb{R}^n \}.$$

Ponieważ kolumny macierzy A są liniowo niezależne, tworzą one bazę w P . Stąd $\dim(P) = n$ i odwzorowanie $A : \mathbb{R}^n \rightarrow P$ jest różnowartościowe. Ponadto przestrzeń \mathbb{R}^m z normą drugą jest przestrzenią unitarną. Residuum jest więc minimalizowane dla wektora $\vec{x}^* \in \mathbb{R}^n$ takiego, że $A\vec{x}^*$ jest rzutem prostopadłym wektora \vec{b} na podprzestrzeń P . (Przypomnijmy, że skończony wymiar P zapewnia, że taki rzut istnieje i jest wyznaczony jednoznacznie.) Równoważnie można powiedzieć, że residuum $\vec{b} - A\vec{x}^*$ jest prostopadłe do P ,

$$\forall \vec{x} \in \mathbb{R}^n \quad (A\vec{x})^T (\vec{b} - A\vec{x}^*) = 0,$$

albo

$$\forall \vec{x} \in \mathbb{R}^n \quad \vec{x}^T (A^T \vec{b} - A^T A \vec{x}^*) = 0.$$

Otrzymaliśmy więc, że wektor $A^T \vec{b} - A^T A \vec{x}^*$ jest prostopadły w \mathbb{R}^n do każdego innego wektora. Ponieważ jedynym wektorem o tej własności jest wektor zerowy, to $A^T A \vec{x}^* = A^T \vec{b}$. \square

Zauważmy, że jeśli macierz A jest kwadratowa, $m = n$, to rozwiązaniem zadania jest $\vec{x}^* = A^{-1} \vec{b}$ i residuum jest zerowe. Zadanie wygładzania liniowego jest więc uogólnieniem rozwiązywania kwadratowych układów równań liniowych.

Równanie (5.2) nazywa się układem *normalnym*. Może ono nam sugerować sposób rozwiązania zadania wygładzania liniowego. Wystarczy bowiem pomnożyć macierz A^T przez A i rozwiązać układ normalny. Zauważmy ponadto, że macierz $A^T A$ jest symetryczna i dodatnio określona, bo $(A^T A)^T = A^T A$ i dla $\vec{x} \neq 0$ mamy $\vec{x}^T (A^T A) \vec{x} = (A\vec{x})^T (A\vec{x}) = \|A\vec{x}\|_2^2 > 0$, przy czym ostatnia nierówność wynika z faktu, że kolumny macierzy A są liniowo niezależne i dlatego $A\vec{x} \neq \vec{0}$. Przy mnożeniu A^T przez A wystarczy więc obliczyć tylko elementy na głównej przekątnej i pod nią, a do rozwiązania równania z macierzą $A^T A$ można zastosować algorytm Banachiewicza-Choleskiego opisany w U. 3.7. Jak łatwo się przekonać, koszt takiego algorytmu wynosi $n^2(k + \frac{n}{3})$, przy czym dominuje koszt obliczenia macierzy $A^T A$.

Ma on jednak pewne wady. Mnożenie macierzy powoduje w fl_ν powstanie “po drodze” dodatkowych błędów, które mogą nawet zmienić rząd macierzy. Na przykład, dla macierzy

$$A = \begin{pmatrix} 1 & 1 & 1 & 1 \\ \varepsilon & & & \\ & \varepsilon & & \\ & & \varepsilon & \\ & & & \varepsilon \end{pmatrix}$$

mamy

$$A^T A = \begin{pmatrix} 1 + \varepsilon^2 & 1 & 1 & 1 \\ 1 & 1 + \varepsilon^2 & 1 & 1 \\ 1 & 1 & 1 + \varepsilon^2 & 1 \\ 1 & 1 & 1 & 1 + \varepsilon^2 \end{pmatrix}.$$

Jeśli $\varepsilon^2 < \nu$ to $fl_\nu(1 + \varepsilon^2) = 1$, co implikuje $\text{rank}(fl_\nu(A^T A)) = 1$, podczas gdy $\text{rank}(fl_\nu(A)) = 4$.

Poniżej przedstawimy inną metodę rozwiązywania zadania wygładzania liniowego, która oparta jest na lustrzanych odbiciach zwanych odbiciami Householdera (który pierwszy zauważył ich istotną przydatność w numerycznej algebrze liniowej).

5.2 Odbicia Householdera

Dla danego wektora $\vec{w} \in \mathbb{R}^m$ o normie $\|\vec{w}\|_2 = \sqrt{\vec{w}^T \vec{w}} = 1$, odbicie (macierz) Householdera zdefiniowane jest jako

$$H = I - 2\vec{w}\vec{w}^T.$$

Zauważmy, że

$$H\vec{x} = \vec{x} - 2(\vec{w}^T \vec{x})\vec{w},$$

a ponieważ $(\vec{w}^T \vec{x})\vec{w} = \langle \vec{x}, \vec{w} \rangle_2 \vec{w}$ jest rzutem prostopadłym \vec{x} na kierunek wektora \vec{w} (tutaj $\langle \cdot, \cdot \rangle_2$ oznacza zwykły iloczyn skalarny w \mathbb{R}^m), to $H\vec{x}$ jest odbiciem lustrzanym wektora \vec{x} względem hiperpłaszczyzny (wymiaru $m - 1$) prostopadłej do \vec{w} .

Odbicia Householdera są przekształceniami nieosobliwymi spełniającymi

$$H^{-1} = H = H^T.$$

Rzeczywiście, ponieważ \vec{w} ma normę jednostkową, mamy

$$H^2 = (I - 2\vec{w}\vec{w}^T)^2 = I - 4\vec{w}\vec{w}^T + 4\vec{w}(\vec{w}^T \vec{w})\vec{w}^T = I,$$

oraz

$$H^T = (I - 2\vec{w}\vec{w}^T)^T = I - 2(\vec{w}^T)^T \vec{w} = I.$$

W szczególności H jest więc przekształceniem *ortogonalnym*, $H^{-1} = H^T$, czyli nie zmienia długości wektora,

$$\|H\vec{x}\|_2 = \sqrt{(H\vec{x})^T (H\vec{x})} = \sqrt{\vec{x}^T (H^T H)\vec{x}} = \sqrt{\vec{x}^T \vec{x}} = \|\vec{x}\|_2.$$

Odbicia Householdera zastosujemy do przeprowadzenia danego wektora $\vec{a} \neq \vec{0}$ na kierunek innego niezerowego wektora, powiedzmy \vec{e} , tzn.

$$H\vec{a} = (I - 2\vec{w}\vec{w}^T)\vec{a} = \alpha \vec{e}.$$

Założmy dla uproszczenia, że $\|\vec{e}\|_2 = 1$. Aby wyznaczyć H zauważmy, że

$$\vec{w} = \frac{\vec{a} - \alpha \vec{e}}{2(\vec{w}^T \vec{a})},$$

a ponieważ $\alpha = \pm \|\vec{a}\|_2$ i $\|\vec{w}\|_2 = 1$ to

$$\vec{w} = \frac{\vec{a} \mp \|\vec{a}\|_2 \vec{e}}{\|\vec{a} \mp \|\vec{a}\|_2 \vec{e}\|_2}.$$

W szczególności, jeśli $\vec{e} = \vec{e}_1$ jest pierwszym wektorem bazy kanonicznej to powyższe wzory dają

$$H = I - \frac{\vec{u} \vec{u}^T}{\gamma},$$

gdzie

$$u_i = \begin{cases} a_1 \mp \|\vec{a}\|_2, & i = 1, \\ a_i, & 2 \leq i \leq m, \end{cases}$$

oraz

$$\begin{aligned} \gamma &= \frac{1}{2} \|\vec{a}\|_2^2 = \frac{1}{2} \left((a_1 \mp \|\vec{a}\|_2)^2 + \sum_{i=2}^m a_i^2 \right) \\ &= \frac{1}{2} \left(\sum_{i=1}^m a_i^2 + \|\vec{a}\|_2^2 \mp 2a_1 \|\vec{a}\|_2 \right) = \|\vec{a}\|_2^2 \mp a_1 \|\vec{a}\|_2. \end{aligned}$$

Otrzymaliśmy dwa odbicia Householdera przekształcające dany wektor \vec{a} na kierunek wektora \vec{e}_1 , w zależności od wybranego znaku przy $\|\vec{a}\|_2$. Ustalimy ten znak na plus gdy $a_1 \geq 0$, oraz na minus gdy $a_1 < 0$, co pozwoli na obliczenie u_1 i γ z małym błędem w fl_ν . Wtedy bowiem mamy

$$u_1 = \begin{cases} a_1 + \|\vec{a}\|_2 & a_1 \geq 0, \\ a_1 - \|\vec{a}\|_2 & a_1 < 0, \end{cases}$$

oraz $\gamma = \|\vec{a}\|_2^2 + |a_1| \|\vec{a}\|_2$, czyli zawsze dodajemy liczby tych samych znaków. Ponadto pierwsza współrzędna wektora $H\vec{a}$ jest równa $-\|\vec{a}\|_2$ dla $a_1 \geq 0$, oraz $+\|\vec{a}\|_2$ dla $a_1 < 0$.

5.3 Algorytm dla zadania wygładzania

Odbić Householdera można użyć do rozkładu macierzy $A \in \mathbb{R}^{m \times n}$ na iloczyn ortogonalno-trójkątny.

Niech $A = (\vec{a}_1, \vec{a}_2, \dots, \vec{a}_n)$, gdzie \vec{a}_j są wektorami-kolumnami macierzy A . Wybierzmy pierwsze odbicie Householdera $H_1 = I_m - \vec{u}_1 \vec{u}_1^T / \gamma_1$ tak, aby przekształcało pierwszy wektor-kolumnę macierzy A na kierunek \vec{e}_1 . Efektem pomnożenia macierzy A z lewej strony przez H_1 będzie wtedy macierz

$$A^{(1)} = (\vec{a}_1^{(1)}, \dots, \vec{a}_n^{(1)}) = (H_1 \vec{a}_1, \dots, H_1 \vec{a}_n),$$

w której pierwsza kolumna, czyli $\vec{a}_1^{(1)}$, ma niezerową tylko pierwszą współrzędną. W następnym kroku wybieramy drugie przekształcenie Householdera $\bar{H}_2 = I_{m-1} - \vec{v}_2 \vec{v}_2^T / \gamma_2$ wymiaru $m-1$ tak, aby przeprowadzało wektor $(a_{i,2}^{(1)})_{i=2}^m$ na kierunek pierwszego wektora bazy kanonicznej w \mathbb{R}^{m-1} . Rozszerzając $\vec{v}_2 \in \mathbb{R}^{m-1}$ do wektora $\vec{u}_2 \in \mathbb{R}^m$ przez dodanie zera jako jego pierwszej współrzędnej, $\vec{u}_2 = (0, \vec{v}_2)^T$, otrzymujemy przekształcenie (macierz) Householdera $H_2 = I_m - \vec{u}_2 \vec{u}_2^T / \gamma_2$ w \mathbb{R}^m postaci

$$H_2 = \begin{pmatrix} 1 & \vec{0}^T \\ \vec{0} & \bar{H}_2 \end{pmatrix}.$$

Pomnożenie macierzy $A^{(1)}$ z lewej strony przez H_2 spowoduje teraz wyzerowanie drugiej kolumny macierzy pod elementem $a_{2,2}^{(1)}$, przy czym pierwszy wiersz i pierwsza kolumna pozostaną niezmienione. Postępując tak dalej n razy (albo $n-1$ razy gdy $m=n$) otrzymujemy

$$H_n H_{n-1} \cdots H_2 H_1 A = R,$$

gdzie $R \in \mathbb{R}^{m \times n}$ jest uogólnioną macierzą trójkątną górną, tzn. $r_{i,j} = 0$ dla $i > j$. Stąd, podstawiając $Q = H_1 H_2 \cdots H_n$, dostajemy rozkład macierzy na iloczyn ortogonalno-trójkątny

$$A = QR. \tag{5.3}$$

Rzeczywiście, macierz $Q \in \mathbb{R}^{m \times m}$ jest ortogonalna, bo

$$\begin{aligned} Q^{-1} &= (H_1 H_2 \cdots H_n)^{-1} = H_n^{-1} \cdots H_2^{-1} H_1^{-1} \\ &= H_n^T \cdots H_2^T H_1^T = (H_1 H_2 \cdots H_n)^T = Q^T. \end{aligned}$$

Dysponując rozkładem (5.3) zadanie wygładzania liniowego można rozwiązać następująco. Ponieważ mnożenie przez macierz ortogonalną nie zmienia normy drugiej wektora, mamy

$$\|\vec{r}\|_2 = \|\vec{b} - A\vec{x}\|_2 = \|\vec{b} - QR\vec{x}\|_2 = \|Q(Q^T\vec{b} - R\vec{x})\|_2 = \|\vec{c} - R\vec{x}\|_2,$$

gdzie $\vec{c} = Q^T\vec{b} = H_n \cdots H_2 H_1 \vec{b}$. Rozbijając wektor \vec{c} na $\vec{c} = (\vec{c}_I, \vec{c}_{II})^T$, gdzie $\vec{c}_I \in \mathbb{R}^n$ i $\vec{c}_{II} \in \mathbb{R}^{m-n}$, oraz macierz R na

$$R = \begin{pmatrix} R_I \\ 0 \end{pmatrix},$$

gdzie $R_I \in \mathbb{R}^{n \times n}$ jest macierzą trójkątną górną, a 0 jest macierzą zerową formatu $(m-n) \times n$, otrzymujemy

$$\|\vec{r}\|_2^2 = \|\vec{c}_I - R_I\vec{x}\|_2^2 + \|\vec{c}_{II}\|_2^2.$$

Rozwiązanie \vec{x}^* zadania wygładzania jest więc rozwiązaniem układu liniowego trójkątnego,

$$R_I\vec{x} = \vec{c}_I$$

oraz $\|\vec{r}^*\|_2 = \|\vec{b} - A\vec{x}^*\|_2 = \|\vec{c}_{II}\|_2$.

Zastanówmy się nad praktyczną realizacją tego algorytmu. Każde z kolejnych przekształceń Householdera H_k wyznaczamy przez obliczenie γ_k oraz współrzędnych wektora \vec{u}_k . Wektor ten ma tylko $m - k + 1$ współrzędnych niezerowych, a ponadto $u_{k,i} = a_{i,k}^{(k-1)}$ dla $k + 1 \leq i \leq m$. Dzięki takiej reprezentacji H_k , mnożenia $H_k\vec{x}$ możemy dla dowolnego \vec{x} realizować według wzoru

$$(H_k\vec{x})_i = x_i - s u_{k,i}, \quad \text{gdzie } s = \vec{u}_k^T \vec{x} / \gamma_k.$$

Uwzględnijając obecność zerowych elementów w \vec{u}_k , przejście od macierzy $A^{(k-1)}$ do $A^{(k)}$ kosztuje $4(m - k + 1)(n - k)$ operacji arytmetycznych i obliczenie jednego pierwiastka kwadratowego. Cały rozkład $A = QR$ kosztuje więc

$$\sum_{k=1}^n 4(m - k + 1)(n - k) \approx \frac{4}{3}n^3 + 2n^2(m - n) = 2n^2\left(m - \frac{1}{3}n\right)$$

operacji arytmetycznych i n pierwiastków kwadratowych. Zauważmy, że w przypadku $m = n$, a więc dla kwadratowego układu równań, koszt ten wynosi $\frac{4}{3}n^3$ i jest dwa razy większy od kosztu eliminacji Gaussa.

Uwagi i uzupełnienia

U. 5.1 Pokazaliśmy, że rozwiązaniem zadania wygładzania liniowego jest wektor $\vec{x}^* = A^+\vec{b}$, gdzie

$$A^+ = (A^T A)^{-1} A^T.$$

Macierz $A^+ \in \mathbb{R}^{n \times m}$ nazywa się macierzą *pseudoodwrotną* do $A \in \mathbb{R}^{m \times n}$, o ile $\text{rank}(A) = n$. Dla nieosobliwych macierzy kwadratowych mamy oczywiście $A^+ = A^{-1}$, ponieważ wtedy $\vec{x}^* = A^{-1}\vec{b}$.

U. 5.2 Pokażemy, że każdą macierz $A^{m \times n}$, $\text{rank}(A) = n$, można rozłożyć na iloczyn

$$A = U \Sigma V^T, \quad (5.4)$$

gdzie $U \in \mathbb{R}^{m \times m}$ i $V \in \mathbb{R}^{n \times n}$ są macierzami ortogonalnymi, a $\Sigma \in \mathbb{R}^{m \times n}$ jest macierzą diagonalną (tzn. $\sigma_{i,j} = 0$ dla $i \neq j$) o dodatnich wyrazach $\sigma_{i,i}$.

Ponieważ macierz $A^T A$ jest symetryczna i dodatnio określona, znane twierdzenie z algebry liniowej mówi, że istnieje w \mathbb{R}^n baza ortonormalna $(\vec{\xi}_j)_{j=1}^n$ wektorów własnych tej macierzy, a odpowiadające im wartości własne są rzeczywiste i dodatnie, tzn.

$$\langle \vec{\xi}_i, \vec{\xi}_j \rangle_2 = \begin{cases} 0 & i \neq j, \\ 1 & i = j, \end{cases}$$

oraz $(A^T A)\vec{\xi}_j = \lambda_j \vec{\xi}_j$, gdzie

$$\lambda_1 \geq \lambda_2 \geq \dots \geq \lambda_n > 0.$$

Zauważmy, że wektory $\vec{\eta}_i = \lambda_i^{-1/2}(A\vec{\xi}_i)$ są ortonormalne w \mathbb{R}^m , bowiem

$$\begin{aligned} \langle \vec{\eta}_i, \vec{\eta}_j \rangle_2 &= (\lambda_i \lambda_j)^{-1/2} \langle A\vec{\xi}_i, A\vec{\xi}_j \rangle_2 = (\lambda_i \lambda_j)^{-1/2} \langle (A^T A)\vec{\xi}_i, \vec{\xi}_j \rangle_2 \\ &= (\lambda_i \lambda_j)^{-1/2} \lambda_i \langle \vec{\xi}_i, \vec{\xi}_j \rangle_2 = \begin{cases} 0, & i \neq j, \\ 1, & i = j. \end{cases} \end{aligned}$$

Wektory $\vec{\eta}_i$, $1 \leq i \leq n$, można uzupełnić $m - n$ dodatkowymi wektorami tak, aby cały układ $(\vec{\eta}_i)_{i=1}^m$ był bazą ortonormalną w \mathbb{R}^m . Zdefiniujmy teraz macierze ortogonalne U o kolumnach $\vec{\eta}_i$, $1 \leq i \leq m$, oraz V o kolumnach $\vec{\xi}_j$, $1 \leq j \leq n$. Bezpośredni rachunek pokazuje, że macierz $U^T A V$ jest diagonalna z wyrazami na przekątnej $\sqrt{\lambda_j}$. Stąd $A = U \Sigma V^T$ z $\sigma_{j,j} = \sqrt{\lambda_j}$, $1 \leq j \leq n$.

Liczby $\sqrt{\lambda_j}$ nazywa się *wartościami szczególnymi* macierzy A , a rozkład (5.4) rozkładem macierzy według wartości szczególnych.

Zauważmy jeszcze, że jeśli $A = U \Sigma V^T$ to $A^+ = V \Sigma^+ U^T$ oraz $\Sigma^+ \in \mathbb{R}^{n \times m}$ jest macierzą diagonalną z wyrazami na diagonalu równymi $1/\sqrt{\lambda_j}$. Rozwiązanie zadania wygładzania liniowego można więc zapisać jako

$$\vec{x}^* = V \Sigma^+ U^T \vec{b}.$$

U. 5.3 Przedstawimy teraz jedną z możliwych implementacji algorytmu Householdera rozkładu macierzy na iloczyn $A = QR$. Po wykonaniu poniższego programu wyrazy $r_{i,j}$ macierzy R dla $1 \leq i \leq j$ zostaną zapamiętane na miejscach $a[i, j]$, współrzędne $u_{j,i}$ wektora \vec{u}_j dla $j + 1 \leq i \leq m$ na miejscach $a[i, k]$, a współrzędna $u_{j,j}$ i liczba γ_j odpowiednio na $u[j]$ i $gam[j]$, $1 \leq j \leq n$.

```

for k := 1 to n do
  begin
    { obliczanie kolejnego odbicia Householdera }
    norm2 := 0.0;
    for l := k to m do
      begin
        aa := a[l, k];
        norm2 := norm2 + aa * aa
      end;
    norm := sqrt(norm2);
    aa := a[k, k];
    if (aa >= 0.0) then
      begin

```

```

    uu := aa + norm;
    akk := -norm
end   else
begin
    uu := aa - norm;
    akk := norm
end;
gamma := norm2 + abs(aa) * norm;
u[k] := uu;
gam[k] := gamma;
{ modyfikacja kolumn macierzy }
a[k, k] := akk;
for j := k + 1 to n do
begin
    s := uu * a[k, j];
    for l := k + 1 to m do
        s := s + a[l, k] * a[l, j];
    s := s/gamma;
    a[k, j] := a[k, j] - s * uu;
    for l := k + 1 to m do
        a[l, j] := a[l, j] - s * a[l, k]
    end;
end.

```

U. 5.4 Można pokazać, że przedstawiony algorytm Householdera rozkładu macierzy na iloczyn ortogonalno-trójkątny jest numerycznie poprawny. To znaczy, otrzymane w fl_ν macierz trójkątna górna R^ν i ortogonalna Q^ν (ale reprezentowana przez n wektorów \vec{u}_k i liczb γ_k) spełniają $(A+E) = Q^\nu R^\nu$, gdzie $\|E\| \leq K(m, n)\nu\|A\|$.

U. 5.5 Zadanie wygładzania liniowego można również rozwiązać stosując innego rodzaju rozkład ortogonalno-trójkątny macierzy $A \in \mathbb{R}^{m \times n}$; mianowicie przez zastosowanie *ortogonalizacji Grama-Schmidta* do wektorów kolumn \vec{a}_j macierzy A . W wyniku otrzymujemy ciąg wektorów \vec{q}_j , $1 \leq j \leq n$, tworzący układ ortonormalny w \mathbb{R}^m , oraz spełniający

$$\text{span}(\vec{q}_1, \vec{q}_2, \dots, \vec{q}_j) = \text{span}(\vec{a}_1, \vec{a}_2, \dots, \vec{a}_j), \quad 1 \leq j \leq n. \quad (5.5)$$

Zauważmy, że jeśli z wektorów \vec{q}_j stworzymy macierz $Q = (\vec{q}_1, \vec{q}_2, \dots, \vec{q}_n) \in \mathbb{R}^{m \times n}$, to (5.5) implikują istnienie kwadratowej macierzy trójkątnej górnej $R \in \mathbb{R}^{n \times n}$ takiej, że

$$A = QR.$$

(W odróżnieniu od efektu działania algorytmu Householdera, macierz Q nie jest tu kwadratowa, ale za to kwadratowa jest macierz R .)

Wektory ortonormalne \vec{q}_j oraz współczynniki $r_{i,j}$ macierzy R można wyznaczyć z układu równań

$$\vec{a}_j = \sum_{s=1}^j r_{s,j} \vec{q}_s, \quad 1 \leq j \leq n.$$

Mnożąc j -te równanie skalarnie przez \vec{q}_i , $1 \leq i \leq j$, otrzymujemy $r_{i,j} = \langle \vec{a}_j, \vec{q}_i \rangle_2 = \vec{q}_i^T \vec{a}_j$, a stąd wzory rekurencyjne

```

for  $j := 1$  to  $n$  do
  begin
    for  $i := 1$  to  $j - 1$  do  $r_{i,j} := \langle \vec{q}_i, \vec{a}_j \rangle_2$ ;
     $\vec{p}_j := \vec{a}_j - \sum_{s=1}^{j-1} r_{s,j} \vec{q}_s$ ;
     $r_{j,j} := \|\vec{p}_j\|_2$ ;
     $\vec{q}_j := \vec{p}_j / r_{j,j}$ 
  end.

```

Dysponując rozkładem $A = QR$, rozwiązanie \vec{x}^* zadania wygładzania wyznaczamy z równania $R\vec{x} = Q^T\vec{b}$. Rzeczywiście, układ $(\vec{q}_j)_{j=1}^n$ można formalnie uzupełnić do układu ortonormalnego w \mathbb{R}^m dodając do niego pewne wektory \vec{q}_j dla $n+1 \leq j \leq m$. Tworząc macierz ortogonalną $\bar{Q} = (\vec{q}_1, \dots, \vec{q}_m)$ i rozszerzając macierz $R \in \mathbb{R}^{n \times n}$ do macierzy $\bar{R} \in \mathbb{R}^{m \times n}$ przez dodanie do niej $m-n$ zerowych wierszy, otrzymujemy $A = \bar{Q}\bar{R}$, a więc rozkład taki jak w algorytmie Householdera. Postępując dalej tak, jak w analizie algorytmu Householdera, otrzymujemy żądany wynik.

U. 5.6 Okazuje się, że algorytm ortogonalizacyjny Grama-Schmidta z U. 5.5 ma niedobre własności numeryczne, gdy wektory-kolumny macierzy wyjściowej A są “słabo liniowo niezależne”; tzn. otrzymane w $f_{l,\nu}$ wektory $\vec{q}_j^{(1)}$ mogą być wtedy “słabo” ortogonalne. W takim wypadku należy stosować podwójną ortogonalizację, najpierw do wektorów \vec{a}_j , a potem do $\vec{q}_j^{(1)}$, $1 \leq j \leq n$. (Zob. Ćw. 5.9.)

Ćwiczenia

Ćw. 5.1 Dla danych

$$(x_1, y_1) = (-1, 0), \quad (x_2, y_2) = (0, -21), \quad (x_3, y_3) = (1, 26), \quad (x_4, y_4) = (2, -8)$$

wyznacz współczynniki $a, b, c \in \mathbb{R}$ takie, że funkcja $f(t) = a + bt + ct(t-1)(16t-56)$ minimalizuje sumę $\sum_{k=1}^4 |f(x_k) - y_k|^2$. W tym celu sformułuj odpowiednie zadanie wygładzania liniowego i rozwiąż je metodą równań normalnych.

Ćw. 5.2 Niech

$$A = \begin{pmatrix} 1 & -1 \\ 1 & -1 \\ 1 & -2 \\ 1 & 0 \end{pmatrix} \in \mathbb{R}^{4 \times 2}.$$

Stosując odbicia Householdera $H_i = I - \vec{u}_i \vec{u}_i^T / \gamma_i$ sprowadź macierz A do postaci trójkątnej górnej $U = H_2 H_1 A$. Wskaż współczynniki macierzy R oraz odpowiednie wektory \vec{u}_i i liczby γ_i , $i = 1, 2$.

Ćw. 5.3 Niech

$$A = \begin{pmatrix} -1 & 3 & 162 & 21 \\ -1 & -8 & -261 & -188 \\ 1 & 5 & -81 & 77 \\ -1 & -8 & 18 & 244 \end{pmatrix}, \quad \vec{b} = \begin{pmatrix} 185 \\ -458 \\ 2 \\ 253 \end{pmatrix}.$$

Znajdź wektory $\vec{v}_1, \vec{v}_2, \vec{v}_3 \in \mathbb{R}^4$ wyznaczające odbicia symetryczne, reprezentowane przez macierze ortogonalne $H_i = I - \vec{v}_i \vec{v}_i^T / \gamma_i$, $\gamma_i = \|\vec{v}_i\|_2 / 2$, takie że macierz $R = H_3 H_2 H_1 A$ jest trójkątna górna. Korzystając z tego rozkładu, rozwiąż układ równań liniowych $A\vec{x} = \vec{b}$.

Ćw. 5.4 Pokaż, że dla macierzy pseudoodwrotnej A^+ do danej macierzy $A \in \mathbb{R}^{m \times n}$, $\text{rank}(A) = n$, macierz (albo operator) $A^+A = I_n$ jest identycznością w $\mathbb{R}^{n \times n}$, natomiast AA^+ jest rzutem prostopadłym w $\mathbb{R}^{m \times m}$ na podprzestrzeń rozpiętą na wektorach-kolumnach macierzy A .

Ćw. 5.5 Niech $A \in \mathbb{R}^{m \times n}$. Uzasadnij, że jądro macierzy A^T ,

$$\ker(A^T) = \{ \vec{y} \in \mathbb{R}^m : A^T \vec{y} = \vec{0} \},$$

jest podprzestrzenią prostopadłą do obrazu A ,

$$\text{im}(A) = \{ A\vec{x} \in \mathbb{R}^m : \vec{x} \in \mathbb{R}^n \}.$$

Ćw. 5.6 Niech \vec{u} będzie niezerowym wektorem w \mathbb{R}^m oraz $\gamma = \|\vec{u}\|_2^2/2$. Uzasadnij, że algorytm obliczania

$$H\vec{x} = \vec{x} - s\vec{u}, \quad s = (\vec{u}^T \vec{x})/\gamma,$$

według powyższego wzoru, jest numerycznie poprawny ze względu na dany wektor $\vec{x} \in \mathbb{R}^m$.

Ćw. 5.7 Policz złożoność algorytmu ortogonalizacyjnego opisanego w U. 5.5 rozwiązania zadania wygładzania liniowego.

Ćw. 5.8 Rozpatrzmy dwa rozkłady ortogonalno-trójkątne macierzy $A \in \mathbb{R}^{m \times n}$, $\text{rank}(A) = n$. Pierwszy to $A = QR$, gdzie $Q \in \mathbb{R}^{m \times m}$ i $R \in \mathbb{R}^{m \times n}$ (np. pochodzący z algorytmu Householdera), a drugi to $A = Q_1 R_1$, gdzie $Q_1 \in \mathbb{R}^{m \times n}$ i $R_1 \in \mathbb{R}^{n \times n}$ (np. pochodzący z algorytmu Grama-Schmidta z U. 5.5). Pokaż, że n pierwszych kolumn macierzy Q różnią się od n pierwszych kolumn macierzy Q_1 co najwyżej znakami; tzn. istnieje macierz diagonalna $D = \text{diag}(d_{1,1}, \dots, d_{n,n}) \in \mathbb{R}^{m \times n}$ taka, że $d_{i,i} = \pm 1$ dla $1 \leq i \leq n$ oraz $Q_1 = QD$.

Ćw. 5.9 Zastosujmy ortogonalizację Grama-Schmidta do dwóch wektorów liniowo niezależnych \vec{a} i \vec{b} o normach $\|\vec{a}\|_2 = 1 = \|\vec{b}\|_2$. Załóżmy dla uproszczenia, że w obliczeniach *jedynie* iloczyn skalarny tych wektorów $s = \langle \vec{a}, \vec{b} \rangle_2$ liczy się z błędem, $f_\nu(s) = s(1 + \varepsilon)$, gdzie $|\varepsilon|$ jest dodatnie i na poziomie ν . Pokaż, że wtedy dla otrzymanej w wyniku ortogonalizacji unormowanej pary wektorów \vec{a} i \vec{c} mamy

$$\langle \vec{a}, \vec{c} \rangle_2 = \frac{-\varepsilon s}{\sqrt{1 - s^2(1 - \varepsilon^2)}}.$$

Wynioskuj stąd, że gdy \vec{a} i \vec{b} są "prawie liniowo zależne", to \vec{a} i \vec{c} są dalekie od ortogonalnych.

Ćw. 5.10 Zaprogramuj algorytm Grama-Schmidta z U. 5.5 (z podwójną ortogonalizacją) rozwiązujący zadanie wygładzania. Przetestuj zachowanie się wyniku dla wektów wyjściowych "prawie liniowo zależnych".

Ćw. 5.11 Zaprogramuj algorytm obliczający macierz odwrotną do danej nieosobliwej macierzy $A \in \mathbb{R}^{n \times n}$ wykorzystujący rozkład Householdera $A = QR$.

Ćw. 5.12 Przetestuj rozwiązywanie układów równań liniowych z Ćw. 4.10 stosując algorytm Householdera. Porównaj wyniki z tymi z Ćw. 4.10.

Rozdział 6

Interpolacja wielomianowa

Dotychczas rozpatrywaliśmy zadania, w których danymi są skończone ciągi liczb rzeczywistych. Teraz zajmiemy się zadaniami, w których danymi są *funkcje* o wartościach rzeczywistych. Pierwszym z nich jest interpolacja wielomianowa. To jeden z popularnych i naturalnych sposobów aproksymacji (albo przybliżania) funkcji, zdefiniowanej w ogólności w Przykładzie 1.4. Dodajmy, że aproksymację stosujemy wtedy gdy chcemy skomplikowane funkcje zastąpić czymś prostszym, albo dokładne odtworzenie funkcji nie jest możliwe.

6.1 Sformułowanie zadania interpolacji

Niech $D \subset \mathbb{R}$ i niech F będzie pewnym zbiorem funkcji $f : D \rightarrow \mathbb{R}$. Niech x_0, x_1, \dots, x_n będzie ustalonym zbiorem parami różnych punktów z D , zwanych później *węzłami*.

Powiemy, że wielomian w *interpoluje* funkcję $f \in F$ w węzłach x_j , gdy

$$w(x_j) = f(x_j), \quad 0 \leq j \leq n.$$

Oznaczmy przez Π_n przestrzeń liniową wielomianów stopnia co najwyżej n o współczynnikach rzeczywistych,

$$\Pi_n = \{ w(x) = a_n x^n + a_{n-1} x^{n-1} + \dots + a_1 x + a_0 : a_j \in \mathbb{R}, 0 \leq j \leq n \}.$$

Lemat 6.1 *Dla dowolnej funkcji $f : D \rightarrow \mathbb{R}$ istnieje dokładnie jeden wielomian $w_{f,n} \in \Pi_n$ interpolujący f w węzłach x_j , $0 \leq j \leq n$.*

Dowód. Wybierzmy w Π_n dowolną bazę wielomianów ϕ_j , $0 \leq j \leq n$,

$$\Pi_n = \text{span}\{ \phi_0, \phi_1, \dots, \phi_n \}.$$

Wtedy każdy wielomian z Π_n można jednoznacznie przedstawić w postaci rozwinięcia względem wybranej bazy. Warunkiem koniecznym i dostatecznym na to, aby wielomian $w_{f,n} = \sum_{j=0}^n c_j \phi_j$ interpolował f jest spełnienie układu $n + 1$ równań liniowych

$$\sum_{j=0}^n c_j \phi_j(x_i) = f(x_i), \quad 0 \leq i \leq n,$$

z $n + 1$ niewiadomymi c_j , który w postaci macierzowej wygląda następująco:

$$\begin{pmatrix} \phi_0(x_0) & \phi_1(x_0) & \cdots & \phi_n(x_0) \\ \phi_0(x_1) & \phi_1(x_1) & \cdots & \phi_n(x_1) \\ \vdots & \vdots & & \vdots \\ \phi_0(x_n) & \phi_1(x_n) & \cdots & \phi_n(x_n) \end{pmatrix} \begin{pmatrix} c_0 \\ c_1 \\ \vdots \\ c_n \end{pmatrix} = \begin{pmatrix} f(x_0) \\ f(x_1) \\ \vdots \\ f(x_n) \end{pmatrix}. \quad (6.1)$$

Aby wykazać, że układ ten ma jednoznaczne rozwiązanie wystarczy, aby wektor zerowy był jedynym rozwiązaniem układu jednorodnego. Rzeczywiście, układ jednorodny odpowiada interpolacji danych zerowych, $f(x_i) = 0$ dla $0 \leq i \leq n$. Istnienie niezerowego rozwiązania byłoby więc równoważne istnieniu niezerowego wielomianu stopnia nie większego od n , który miałby $n + 1$ różnych zer x_i , co jest niemożliwe. \square

Zauważmy, że wielomian interpolacyjny $w_{f,n}$ jako taki nie może być wynikiem obliczeń w naszym modelu obliczeniowym (bo co to znaczy “obliczyć wielomian?”), możemy natomiast spróbować wyznaczyć jego współczynniki c_j w wybranej bazie dysponując wartościami $f(x_j)$.

Definicja 6.1 Niech $(\phi_j)_{j=0}^n$ będzie ustaloną bazą w przestrzeni Π_n wielomianów stopnia co najwyżej n . Zadanie (obliczeniowe) interpolacji wielomianowej polega na obliczeniu dla danej funkcji f współczynników c_j takich, że wielomian

$$w_{f,n} = \sum_{j=0}^n c_j \phi_j \quad (6.2)$$

interpoluje f w punktach x_j , $0 \leq j \leq n$.

Zauważmy, że interpolacja wielomianowa jest zadaniem liniowym, tzn. współczynniki $\{c_j\}$ wielomianu interpolacyjnego zależą liniowo od interpolowanej funkcji f .

6.2 Wybór bazy wielomianowej

Można powiedzieć, że trudność zadania interpolacji jest zdeterminowana przez trudność rozwiązania układu (6.1), ta zaś zależy w istotny sposób od wybranej bazy $(\phi_j)_{j=0}^n$. W naturalny sposób powstaje więc problem wyboru “wygodnej” bazy w Π_n . Rozpatrzmy trzy bazy: Lagrange’a, potęgową i Newtona.

BAZA LAGRANGE’A (KANONICZNA)

Zdefiniujmy dla $0 \leq j \leq n$ wielomiany

$$l_j(x) = \frac{(x - x_0)(x - x_1) \cdots (x - x_{j-1})(x - x_{j+1}) \cdots (x - x_n)}{(x_j - x_0)(x_j - x_1) \cdots (x_j - x_{j-1})(x_j - x_{j+1}) \cdots (x_j - x_n)}. \quad (6.3)$$

Zauważmy, że każdy z l_j jest stopnia dokładnie n oraz

$$l_j(x_i) = \begin{cases} 0 & i \neq j, \\ 1 & i = j. \end{cases}$$

Stąd łatwo widać, że wielomiany te stanowią bazę w Π_n , którą nazywamy bazą Lagrange’a. Macierz układu (6.1) jest identycznością i w konsekwencji dla wszystkich j mamy $c_j = f(x_j)$. Wielomian interpolacyjny dla funkcji f można więc w tym przypadku zapisać jako

$$w_{f,n} = \sum_{j=0}^n f(x_j) l_j.$$

Koszt kombinatoryczny rozwiązania zadania interpolacji jest przy tym zerowy.

Przypuśćmy, że chcielibyśmy obliczyć wartość wielomianu interpolacyjnego $w_{f,n}$ w punkcie x różnym od x_j , $0 \leq j \leq n$. Podstawiając

$$a_j = \frac{1}{(x_j - x_0)(x_j - x_1) \cdots (x_j - x_{j-1})(x_j - x_{j+1}) \cdots (x_j - x_n)}$$

oraz $a(x) = (x - x_0) \cdots (x - x_n)$ mamy

$$w_{f,n}(x) = a(x) \sum_{j=0}^n \frac{a_j f(x_j)}{x - x_j}, \quad (6.4)$$

albo

$$w_{f,n}(x) = \frac{\sum_{j=0}^n q_j(x) f(x_j)}{\sum_{j=0}^n q_j(x)},$$

gdzie $q_j(x) = a_j/(x - x_j)$. W ostatniej równości wykorzystaliśmy fakt, że $a(x) = (\sum_{j=0}^n q_j(x))^{-1}$, co wynika z (6.4) po podstawieniu $f = 1$.

BAZA POTĘGOWA (NATURALNA)

Znacznie prościej można obliczyć wartość wielomianu interpolacyjnego, (a także jego pochodnych), gdy jest on dany w najczęściej używanej bazie potęgowej, $\phi_j(x) = x^j$. Jeśli bowiem

$$w_{f,n}(x) = a_0 + a_1 x + \cdots + a_n x^n,$$

to również

$$w_{f,n}(x) = (\cdots (a_n x + a_{n-1})x + a_{n-2})x + \cdots + a_1)x + a_0,$$

co sugeruje zastosowanie następującego *schematu Hornera* do obliczenia $w_{f,n}(x)$:

```

v_n := a_n;
for j := n - 1 downto 0 do
    v_j := v_{j+1} * x + a_j.

```

Po wykonaniu tego algorytmu $w_{f,n}(x) = v_0$. Schemat Hornera wymaga wykonania tylko n mnożeń i n dodawań. Ma on również głębszy sens, zob. Ćw. 6.2.

Zauważmy jednak, że w przypadku bazy potęgowej macierz $(x_i^j)_{i,j=0}^n$ układu (6.1) jest pełna. Jest to tzw. *macierz Vandermonde'a*. Obliczenie współczynników wielomianu interpolacyjnego w bazie potęgowej bezpośrednio z tego układu, stosując jedną ze znanych nam już metod, kosztowałoby rzędu n^3 operacji arytmetycznych.

BAZA NEWTONA

Rozwiązaniem pośrednim, które łączy prostotę obliczenia współczynników z prostotą obliczenia wartości $w_{f,n}(x)$ jest wybór bazy Newtona,

$$\begin{aligned} p_0(x) &= 1, \\ p_j(x) &= (x - x_0)(x - x_1) \cdots (x - x_{j-1}), \quad 1 \leq j \leq n. \end{aligned}$$

W tym wypadku współczynniki rozwinięcia wielomianu interpolacyjnego będziemy oznaczać przez b_j ,

$$w_{f,n} = \sum_{j=0}^n b_j p_j.$$

Zwróćmy od razu uwagę na ważną własność bazy Newtona. Jeśli $w_{f,j} \in \Pi_j$ jest wielomianem interpolacyjnym dla funkcji f opartym na węzłach x_0, x_1, \dots, x_j , $0 \leq j \leq n$, to $w_{f,0} = b_0$ oraz

$$w_{f,j} = w_{f,j-1} + b_j p_j, \quad 1 \leq j \leq n. \quad (6.5)$$

Wartość $w_{f,n}(x)$ można obliczyć stosując prostą modyfikację algorytmu Hornera:

```

 $v_n := b_n;$ 
for  $j := n - 1$  downto  $0$  do
   $v_j := v_{j+1} * (x - x_j) + b_j.$ 

```

Ponadto układ równań (6.1) jest trójkątny dolny (tzn. $p_j(x_i) = 0$ dla $i < j$), co pozwala na obliczenie b_j algorytmem z Rozdziału 3.1 o koszcie n^2 ,

```

 $b_0 := f(x_0);$ 
for  $j := 1$  to  $n$  do
   $b_j := (f(x_j) - \sum_{s=0}^{j-1} b_s p_s(x_j)) / p_j(x_j).$ 

```

Zwykle jednak do obliczania współczynników b_j stosuje się nieco inny algorytm, który teraz przedstawimy.

6.3 Algorytm różnic dzielonych

Różnicę dzieloną funkcji f opartą na różnych węzłach t_0, t_1, \dots, t_s , gdzie $s \geq 1$, definiuje się indukcyjnie jako $f[x_0] = f(x_0)$,

$$f[t_0, t_1, \dots, t_s] = \frac{f[t_1, t_2, \dots, t_s] - f[t_0, t_1, \dots, t_{s-1}]}{t_s - t_0}. \quad (6.6)$$

Zachodzi następujące ważne twierdzenie.

Twierdzenie 6.1 *Współczynniki b_j wielomianu interpolacyjnego Newtona dla danej funkcji f dane są przez różnice dzielone f w węzłach x_0, x_1, \dots, x_j , tzn.*

$$b_j = f[x_0, x_1, \dots, x_j], \quad 0 \leq j \leq n.$$

Dowód. Dla $0 \leq i \leq j \leq n$, oznaczmy przez $w_{i,j}$ wielomian z Π_{j-i} interpolujący f w węzłach x_i, x_{i+1}, \dots, x_j . Wtedy ma miejsce następująca równość ($i < j$):

$$w_{i,j}(x) = \frac{(x - x_i)w_{i+1,j}(x) - (x - x_j)w_{i,j-1}(x)}{x_j - x_i}. \quad (6.7)$$

Aby ją pokazać wystarczy, że prawa strona tej równości, którą oznaczmy przez $v(x)$, przyjmuje wartości $f(x_s)$ dla $x = x_s$, $i \leq s \leq j$. Rzeczywiście, jeśli $i + 1 \leq s \leq j - 1$ to

$$v(x_s) = \frac{(x_s - x_i)f(x_s) - (x_s - x_j)f(x_s)}{x_j - x_i} = f(x_s).$$

Ponadto

$$v(x_i) = \frac{-(x_i - x_j)f(x_i)}{x_j - x_i} = f(x_i),$$

oraz podobnie $v(x_j) = f(x_j)$. Stąd v jest wielomianem z Π_{j-i} interpolującym f w węzłach x_s , $i \leq s \leq j$, czyli $w_{i,j} = v$.

Dalej postępujemy indukcyjnie ze względu na stopień n wielomianu interpolacyjnego. Dla $n = 0$ mamy oczywiście $b_0 = f(x_0)$. Niech $n \geq 1$. Ponieważ, jak łatwo zauważyć,

$$w_{0,n}(x) = w_{0,n-1}(x) + b_n p_n(x),$$

z założenia indukcyjnego mamy $b_j = f[x_0, \dots, x_j]$ dla $0 \leq j \leq n-1$. Aby pokazać podobną równość dla b_n , wykorzystamy równość (6.7) z $i = 0$ i $j = n$, która ma wtedy postać

$$w_{0,n}(x) = \frac{(x-x_0)w_{1,n}(x) - (x-x_n)w_{0,n-1}(x)}{x_n - x_0}.$$

Zauważmy teraz, że b_n jest współczynnikiem przy x^n w wielomianie $w_{0,n}$. Z założenia indukcyjnego wynika, że współczynniki przy x^{n-1} w wielomianach $w_{1,n}$ i $w_{0,n-1}$ są ilorazami różnicowymi opartymi odpowiednio na węzłach x_1, \dots, x_n i x_0, \dots, x_{n-1} . Stąd

$$b_n = \frac{f[x_1, \dots, x_n] - f[x_0, \dots, x_{n-1}]}{x_n - x_0} = f[x_0, x_1, \dots, x_n],$$

co kończy dowód. \square

Różnicę dzieloną $f[x_0, x_1, \dots, x_n]$ można łatwo obliczyć na podstawie wartości $f(x_j)$, $0 \leq j \leq n$, budując następującą tabelkę:

$$\begin{array}{ccccccc} x_0 & f[x_0] & & & & & \\ x_1 & f[x_1] & f[x_0, x_1] & & & & \\ x_2 & f[x_2] & f[x_1, x_2] & f[x_0, x_1, x_2] & & & \\ \vdots & \vdots & \vdots & \vdots & & \ddots & \\ x_n & f[x_n] & f[x_{n-1}, x_n] & f[x_{n-2}, x_{n-1}, x_n] & \cdots & f[x_0, x_1, \dots, x_n]. & \end{array}$$

Zauważmy przy tym, że “po drodze” obliczamy $f[x_i, x_{i+1}, \dots, x_j]$ dla wszystkich $0 \leq i < j \leq n$, a więc w szczególności również interesujące nas różnice dzielone $f[x_0, x_1, \dots, x_j]$. Stąd i z Twierdzenia 6.1 natychmiast wynika algorytm obliczania współczynników b_j wielomianu interpolacyjnego Newtona. Po wykonaniu następującego programu,

```
for j := 0 to n do b[j] := f(x[j]);
for j := 1 to n do
for k := n downto j do
    b[k] := (b[k] - b[k-1]) / (x[k] - x[k-j]),
```

współczynniki b_j zostaną umieszczone w tablicy $b[j]$.

6.4 Przypadek węzłów wielokrotnych

Uogólnieniem rozpatrzonego zadania interpolacji jest zadanie interpolacji *Hermite'a*. Zakładamy, że oprócz (różnych) węzłów x_j dane są również ich krotności n_j , $0 \leq j \leq k$, przy czym $\sum_{j=0}^k n_j = n + 1$. Należy skonstruować wielomian $w_{f,n} \in \Pi_n$ taki, że

$$w_{f,n}^{(i)}(x_j) = f^{(i)}(x_j) \quad \text{dla} \quad 0 \leq i \leq n_j - 1, \quad 0 \leq j \leq k.$$

Oczywiście zakładamy przy tym, że odpowiednie pochodne funkcji f istnieją.

Lemat 6.2 *Zadanie interpolacji Hermite'a ma jednoznaczne rozwiązanie.*

Dowód Istnienie i jednoznaczność rozwiązania można uzasadnić tak samo jak w przypadku węzłów jednokrotnych. Przedstawiając wielomian w dowolnej bazie otrzymujemy układ $n + 1$ równań z $n + 1$ niewiadomymi, który dla zerowej prawej strony ma jedynie rozwiązanie zerowe. Inaczej bowiem istniałby wielomian niezerowy stopnia nie większego niż n , który miałby zera o łącznej krotności większej niż n . \square

Nas oczywiście interesuje konstrukcja wielomianu $w_{f,n}$. W tym celu ustawimy węzły x_j w ciąg

$$(\bar{x}_0, \bar{x}_1, \dots, \bar{x}_n) = (\underbrace{x_0, \dots, x_0}_{n_0}, \underbrace{x_1, \dots, x_1}_{n_1}, \dots, \underbrace{x_k, \dots, x_k}_{n_k})$$

i zdefiniujemy uogólnioną bazę Newtona w Π_n jako

$$\begin{aligned} p_0(x) &= 1, \\ p_j(x) &= (x - \bar{x}_0)(x - \bar{x}_1) \cdots (x - \bar{x}_{j-1}), \quad 1 \leq j \leq n. \end{aligned}$$

Uogólnimy również pojęcie różnicy dzielonej na węzły powtarzające się kładąc

$$f[\bar{x}_i, \bar{x}_{i+1}, \dots, \bar{x}_j] = \frac{f^{(j-i)}(\bar{x}_i)}{(j-i)!}$$

dla $\bar{x}_i = \bar{x}_{i+1} = \cdots = \bar{x}_j$, oraz

$$f[\bar{x}_i, \bar{x}_{i+1}, \dots, \bar{x}_j] = \frac{f[\bar{x}_{i+1}, \dots, \bar{x}_j] - f[\bar{x}_i, \dots, \bar{x}_{j-1}]}{\bar{x}_j - \bar{x}_i}$$

dla $\bar{x}_i \neq \bar{x}_j$. Zauważmy, że przy tej definicji różnice $f[\bar{x}_i, \dots, \bar{x}_j]$ możemy łatwo obliczyć stosując schemat podobny do tego z przypadku węzłów jednokrotnych.

Twierdzenie 6.2 *Współczynniki b_j wielomianu interpolacyjnego Hermite'a w bazie Newtona,*

$$w_{f,n} = \sum_{j=0}^n b_j p_j,$$

dane są przez odpowiednie różnice dzielone, tzn.

$$b_j = f[\bar{x}_0, \bar{x}_1, \dots, \bar{x}_j], \quad 0 \leq j \leq n.$$

Dowód Dowód przeprowadzimy podobnie jak dla węzłów jednokrotnych. Niech $w_{i,j} \in \Pi_{j-i}$ oznacza wielomian interpolacyjny Hermite'a oparty na (być może powtarzających się) węzłach $\bar{x}_i, \bar{x}_{i+1}, \dots, \bar{x}_j$. To znaczy, $w_{i,j}$ interpoluje f w węzłach x_s takich, że x_s występuje w ciągu $\bar{x}_i, \dots, \bar{x}_j$, a jego krotność jest liczbą powtórzeń x_s w tym ciągu.

Zauważmy najpierw, że dla $\bar{x}_i \neq \bar{x}_j$ zachodzi znany nam już wzór (6.7),

$$w_{i,j}(x) = \frac{(x - \bar{x}_i)w_{i+1,j}(x) - (x - \bar{x}_j)w_{i,j-1}(x)}{\bar{x}_j - \bar{x}_i}. \quad (6.8)$$

Rzeczywiście, oznaczmy przez $v(x)$ prawą stronę powyższej równości. Dla k mniejszego od krotności danego węzła x_s w ciągu $\bar{x}_i, \dots, \bar{x}_j$, mamy $w_{i+1,j}^{(k-1)}(x_s) = w_{i,j-1}^{(k-1)}(x_s)$, a ponieważ

$$v^{(k)}(x) = \frac{k(w_{i+1,j}^{(k-1)}(x) - w_{i,j-1}^{(k-1)}(x))}{\bar{x}_j - \bar{x}_i} + \frac{(x - \bar{x}_i)w_{i+1,j}^{(k)}(x) - (x - \bar{x}_j)w_{i,j-1}^{(k)}(x)}{\bar{x}_j - \bar{x}_i},$$

to

$$v^{(k)}(x_s) = \frac{(x_s - \bar{x}_i)w_{i+1,j}^{(k)}(x_s) - (x_s - \bar{x}_j)w_{i,j-1}^{(k)}(x_s)}{\bar{x}_j - \bar{x}_i}.$$

Korzystając z tego wzoru sprawdzamy, że v spełnia warunki interpolacyjne, a stąd $w_{i,j} = v$.

Dalej postępujemy indukcyjnie ze względu na n . Dla $n = 0$ mamy $b_0 = f[x_0]$. Dla $n \geq 1$ wystarczy pokazać, że $b_n = f[\bar{x}_0, \bar{x}_1, \dots, \bar{x}_n]$. W tym celu rozpatrzmy dwa przypadki.

Jeśli $\bar{x}_0 = \bar{x}_n$ to mamy jeden węzeł x_0 o krotności $n + 1$. Wielomian interpolacyjny jest wtedy postaci

$$w_{f,n}(x) = \sum_{j=0}^n \frac{f^{(j)}(x_0)}{j!} (x - x_0)^j,$$

a stąd $b_n = f^{(n)}(x_0)/(n!) = f[\underbrace{x_0, \dots, x_0}_{n+1}]$. Jeśli zaś $\bar{x}_0 \neq \bar{x}_j$ to równość $b_n = f[\bar{x}_0, \bar{x}_1, \dots, \bar{x}_n]$ wynika z (6.8) po podstawieniu $i = 0$ i $j = n$, oraz z założenia indukcyjnego. \square

Uwagi i uzupełnienia

U. 6.1 Zbadamy uwarunkowanie zadania interpolacji Lagrange'a dla danych węzłów x_i ze względu na dane $y_j = f(x_j)$, $0 \leq j \leq n$. Zakładamy, że dziedzina D funkcji f jest zbiorem zwartym, np. $D = [a, b]$, gdzie $-\infty < a < b < +\infty$. Zdefiniujemy normę w przestrzeni wyników \mathbb{R}^{n+1} jako

$$\|(c_0, c_1, \dots, c_n)^T\| = \|w\| := \max_{x \in D} |w(x)|,$$

gdzie $w = \sum_{j=0}^n c_j \phi_j$, a $\{\phi_j\}_{j=0}^n$ jest jak wcześniej wybraną bazą w Π_n . Innymi słowy, norma wyniku to norma max otrzymanego wielomianu interpolacyjnego. Niech $\vec{z} = (z_0, \dots, z_n)^T$ będą zaburzonymi danymi $\vec{y} = (y_0, \dots, y_n)^T$ takimi, że $\|\vec{z} - \vec{y}\|_\infty \leq \varepsilon \|\vec{y}\|_\infty$. Wtedy różnica pomiędzy wielomianem interpolacyjnym w dla danych dokładnych \vec{y} a wielomianem interpolacyjnym \tilde{w} dla danych zaburzonych \vec{z} wynosi

$$w(x) - \tilde{w}(x) = \sum_{i=0}^n (y_i - z_i) l_i(x),$$

gdzie l_i są wielomianami Lagrange'a dla węzłów $\{x_i\}$. Stąd i z nierówności $|z_i - y_i| \leq \varepsilon \|\vec{y}\|_\infty \leq \varepsilon \|w\|$ dostajemy

$$\|w - \tilde{w}\| \leq \Lambda \varepsilon \|w\|, \quad \text{gdzie} \quad \Lambda = \max_{x \in D} \sum_{i=0}^n |l_i(x)|.$$

Zauważmy, że Λ nie zależy ani od wyboru bazy $\{\phi_i\}$, ani od danych \vec{y} , a jedynie od węzłów $\{x_i\}$ i dziedziny D . O Λ dowiemy się więcej w następnym rozdziale.

U. 6.2 Algorytm Hornera okazuje się optymalny. Każdy inny algorytm obliczający dokładną wartość wielomianu znając jego współczynniki wymaga wykonania co najmniej n mnożeń i n dodawań. Algorytm Hornera jest też numerycznie poprawny, zob. Ćw. 6.1.

U. 6.3 Mając obliczone współczynniki b_j wielomianu interpolacyjnego w postaci Newtona, można łatwo przejść do jego współczynników a_j w bazie potęgowej. Algorytm wynika bowiem z następującej zależności rekurencyjnej.

Dla $i = n, n-1, \dots, 0$, niech $a_j^{(i)}$, $i \leq j \leq n$, będą odpowiednimi współczynnikami w rozwinięciu

$$b_i + b_{i+1}(x - x_i) + \dots + b_n(x - x_i) \cdots (x - x_{n-1}) = a_i^{(i)} + a_{i+1}^{(i)}x + \dots + a_n^{(i)}x^{n-i}.$$

Wtedy $a_n^{(i)} = b_n$, a dla $i = n-1, n-2, \dots, 1, 0$ mamy

$$\begin{cases} a_n^{(i)} = a_n^{(i+1)}, \\ a_k^{(i)} = a_k^{(i+1)} - x_i a_{k+1}^{(i+1)}, \\ a_i^{(i)} = b_i - x_i a_{i+1}^{(i+1)}. \end{cases} \quad k = n-1, n-2, \dots, i+1,$$

Uwzględniając fakt, że poszukiwanymi współczynnikami są $a_j = a_j^{(0)}$, $0 \leq j \leq n$, algorytm może być zrealizowany następująco:

```
for j := 0 to n do a[j] := b[j];
for i := n-1 downto 0 do
for k := i to n-1 do
a[k] := a[k] - x[i] * a[k+1].
```

U. 6.4 Okazuje się, że przy realizacji w fl_ν algorytmu różnic dzielonych istotną rolę odgrywa porządek węzłów. Można pokazać, że algorytm liczenia $f[t_0, \dots, t_n]$ jest numerycznie poprawny ze względu na dane interpolacyjne $f^{(i)}(t_j)$, o ile węzły są uporządkowane nierosnąco lub niemalejąco, zob. Ćw. 6.6.

U. 6.5 Zauważmy, że pojęcie różnicy dzielonej formalnie zdefiniowaliśmy jedynie dla ciągu węzłów postaci $x_0, \dots, x_0, x_1, \dots, x_1, \dots, x_k, \dots, x_k$, gdzie x_j są parami różne. Tą definicję można rozszerzyć do dowolnego ciągu węzłów. Można bowiem powiedzieć, że $f[t_0, t_1, \dots, t_n]$ jest współczynnikiem przy x^n wielomianu $w_{t_0, \dots, t_n} \in \Pi_n$ interpolującego f w węzłach t_j (uwzględniając krotności). Równoważnie,

$$f[t_0, t_1, \dots, t_n] = \frac{w_{t_0, \dots, t_n}^{(n)}}{n!}.$$

Ćwiczenia

Ćw. 6.1 Pokaż numeryczną poprawność algorytmu Hornera obliczania wartości wielomianu ze względu na dane współczynniki a_j tego wielomianu.

Ćw. 6.2 Pokaż, że algorytm Hornera obliczania wartości $w(\xi)$ wielomianu danego w postaci potęgowej jest jednocześnie algorytmem dzielenia tego wielomianu przez jednomian $(x - \xi)$. Dokładniej, jeśli $w(x) = \sum_{j=0}^n a_j x^j$ to

$$w(x) = \left(\sum_{j=1}^n v_j x^{j-1} \right) (x - \xi) + v_0,$$

gdzie v_j są zdefiniowane tak jak w algorytmie Hornera.

Ćw. 6.3 Zaproponuj możliwie tani algorytm obliczający współczynniki wielomianu w bazie Newtona, dysponując współczynnikami tego wielomianu w bazie potęgowej.

Ćw. 6.4 Czy istnieje algorytm rozwiązujący układ równań liniowych z macierzą Vandermode'a o koszcie proporcjonalnym do n^2 ?

Ćw. 6.5 Pokaż, że dla różnych węzłów t_0, \dots, t_n mamy

$$f[t_0, t_1, \dots, t_n] = \sum_{j=0}^n \frac{f(t_j)}{(t_j - t_0) \cdots (t_j - t_{j-1})(t_j - t_{j+1}) \cdots (t_j - t_n)}.$$

Ćw. 6.6 Niech $t_0 < t_1 < \dots < t_n$. Pokaż, że realizując w \mathcal{F}_ν algorytm różnic dzielonych, zamiast dokładnej wartości $f[t_0, t_1, \dots, t_n]$ otrzymujemy $\mathcal{F}_\nu(f[t_0, t_1, \dots, t_n])$, która jest dokładną różnicą dzieloną dla danych $f(t_j)(1 + \varepsilon_j)$, gdzie $|\varepsilon_j| \leq K\nu$, $0 \leq j \leq n$.

Wskazówka Skorzystaj z Ćw. 6.5.

Ćw. 6.7 Uzasadnij, że iloraz różnicowy jest funkcją symetryczną, tzn. dla dowolnej permutacji t_{i_0}, \dots, t_{i_s} węzłów t_0, \dots, t_s mamy

$$f[t_{i_0}, \dots, t_{i_s}] = f[t_0, \dots, t_s].$$

Ćw. 6.8 Niech w będzie wielomianem stopnia dokładnie n . Pokaż, że dla ustalonych t_1, t_2, \dots, t_s funkcja

$$w_{t_1, \dots, t_s}(t) = w[t_1, t_2, \dots, t_s, t], \quad t \in \mathbb{R},$$

jest wielomianem stopnia dokładnie $n - s$ dla $s \leq n$, oraz jest zerem dla $s \geq n + 1$.

Ćw. 6.9 Pokaż, że jeśli funkcja $f \in C^n([a, b])$ to dla dowolnego t_n mamy

$$\lim_{t \rightarrow t_n} f[t_0, t_1, \dots, t_{n-1}, t] = f[t_0, t_1, \dots, t_{n-1}, t_n].$$

Ćw. 6.10 Pokaż, że jeśli funkcja $f \in C^n([a, b])$ to

$$\frac{\partial}{\partial x_n} f[x_0, x_1, \dots, x_{n-1}, x_n] = f[x_0, x_1, \dots, x_{n-1}, x_n, x_n].$$

Ćw. 6.11 Niech $x_0 < x_1 < \dots < x_n$. Wskaż wielomiany $p_i, q_i \in \Pi_{2n+1}$ dla $0 \leq i \leq n$ takie, że

$$p_i(x_j) = \delta_{i,j}, \quad p_i'(x_j) = 0, \quad q_i(x_j) = 0, \quad q_i'(x_j) = \delta_{i,j},$$

$0 \leq i, j \leq n$. Odpowiedź podaj ‘w języku’ wielomianów Lagrange’a l_i dla węzłów $\{x_i\}$.

Ćw. 6.12 Znajdź w postaci potęgowej wielomian $w_{f,5}$ interpolujący funkcję $f(x) = x^6$ w trzech dwukrotnych węzłach równych 0, 1 i 2 wykorzystując: (a) algorytm różnic dzielonych, (b) wielomiany p_i i q_i z Ćw. 6.10.

Rozdział 7

Interpolacja a aproksymacja

Jak zauważyliśmy wcześniej, interpolacja wielomianowa może być traktowana jako jeden ze sposobów aproksymacji funkcji, opartym na próbkowaniu. Naturalnym staje się więc pytanie o błąd takiej aproksymacji i jak daleko jej do optymalnej aproksymacji wielomianowej.

7.1 Błąd interpolacji

Niech x_0, x_1, \dots, x_n będą (niekoniecznie różnymi) węzłami należącymi do pewnego (być może nieskończonego) przedziału $D \subset \mathbb{R}$. Dla danej funkcji $f : D \rightarrow \mathbb{R}$, przez $w_{f,n}$ oznaczmy, tak jak w poprzednim rozdziale, wielomian interpolacyjny stopnia co najwyżej n interpolujący f w zadanych węzłach. W przypadku węzłów wielokrotnych jest to oczywiście wielomian interpolacyjny Hermite'a.

Lemat 7.1 *Dla dowolnego punktu $\bar{x} \in D$ błąd interpolacji w \bar{x} wyraża się wzorem*

$$f(\bar{x}) - w_{f,n}(\bar{x}) = (\bar{x} - x_0)(\bar{x} - x_1) \cdots (\bar{x} - x_n) f[x_0, x_1, \dots, x_n, \bar{x}]. \quad (7.1)$$

Jeśli ponadto $f \in C^{n+1}(D)$, czyli pochodna $f^{(n+1)}$ w D istnieje i jest ciągła, to

$$f(\bar{x}) - w_{f,n}(\bar{x}) = (\bar{x} - x_0)(\bar{x} - x_1) \cdots (\bar{x} - x_n) \frac{f^{(n+1)}(\xi)}{(n+1)!},$$

gdzie $\xi = \xi(\bar{x})$ jest pewnym punktem należącym do najmniejszego przedziału zawierającego punkty $x_0, x_1, \dots, x_n, \bar{x}$.

Dowód. Możemy założyć, że \bar{x} nie jest żadnym z węzłów x_j , $0 \leq j \leq n$. Niech $\bar{w}_{f,n} \in \Pi_{n+1}$ będzie wielomianem interpolacyjnym funkcji f opartym na węzłach x_0, \dots, x_n i dodatkowo na węźle \bar{x} . Mamy wtedy

$$\bar{w}_{f,n}(x) = w_{f,n}(x) + (x - x_0)(x - x_1) \cdots (x - x_n) f[x_0, x_1, \dots, x_n, \bar{x}],$$

a ponieważ z warunku interpolacyjnego $f(\bar{x}) = \bar{w}_{f,n}(\bar{x})$, to mamy też (7.1).

Aby pokazać drugą część lematu, rozpatrzmy funkcję $\psi : D \rightarrow \mathbb{R}$,

$$\psi(x) = f(x) - \bar{w}_{f,n}(x) = f(x) - w_{f,n}(x) - (x - x_0)(x - x_1) \cdots (x - x_n) f[x_0, \dots, x_n, \bar{x}].$$

Z warunków interpolacyjnych na $\bar{w}_{f,n} \in \Pi_{n+1}$ wynika, że funkcja ψ ma punkty zerowe o łącznej krotności co najmniej $n+2$. Stąd i z twierdzenia Rolle'a wnioskujemy, że ψ' ma zera o łącznej krotności

co najmniej $n + 1$, ψ'' ma zera o łącznej krotności co najmniej n , itd. W końcu funkcja $\psi^{(n+1)}$ zeruje się w co najmniej jednym punkcie $\xi = \xi(\bar{x})$ należącym do najmniejszego przedziału zawierającego $x_0, x_1, \dots, x_n, \bar{x}$. Wobec tego, że $w_{f,n}^{(n+1)} = 0$, a pochodna rzędu $n + 1$ wielomianu $(x - x_0) \cdots (x - x_n)$ wynosi $(n + 1)!$, mamy

$$0 = \psi^{(n+1)}(\xi) = f^{(n+1)}(\xi) - (n + 1)!f[x_0, \dots, x_n, \bar{x}].$$

Stąd

$$f[x_0, x_1, \dots, x_n, \bar{x}] = \frac{f^{(n+1)}(\xi)}{(n + 1)!},$$

co kończy dowód. \square

Zwykle interesuje nas nie tyle błąd w ustalonym punkcie $\bar{x} \in D$, co błąd na całym przedziale D . Zakładając, że przedział D jest domknięty, czyli

$$D = [a, b]$$

dla pewnych $-\infty < a < b < +\infty$, błąd ten będziemy mierzyć w normie *jednostajnej* (Czebyszewa) $\|\cdot\|_C$. Przypomnijmy, że dla funkcji ciągłej $f : [a, b] \rightarrow \mathbb{R}$ norma ta jest zdefiniowana jako

$$\|f\|_C = \max_{a \leq x \leq b} |f(x)|.$$

Natomiast jakość danego sposobu aproksymacji będziemy mierzyć poprzez błąd najgorszy w danej klasie funkcji. Naszym "papierkiem lakmusowym" będą klasy funkcji gładkich

$$C_M^r([a, b]) = \left\{ f \in C^r([a, b]) : \|f^{(r)}\|_C \leq M \right\}, \quad (7.2)$$

gdzie $0 < M < +\infty$. Oczywiście, $C^r([a, b])$ oznacza tradycyjnie przestrzeń funkcji r -krotnie różniczkowalnych w sposób ciągły, na przedziale $[a, b]$.

Mamy następujące twierdzenie na temat błędu najgorszego interpolacji.

Twierdzenie 7.1 *Załóżmy, że każdą funkcję $f \in C_M^{r+1}([a, b])$ aproksymujemy jej wielomianem interpolacyjnym $w_{f,r} \in \Pi_r$ opartym na $r + 1$ węzłach $x_0, \dots, x_r \in [a, b]$. Maksymalny błąd takiej aproksymacji wynosi*

$$\max_{f \in C_M^{r+1}([a, b])} \|f - w_{f,r}\|_C = \frac{M}{(r + 1)!} \max_{a \leq x \leq b} |(x - x_0) \cdots (x - x_r)|.$$

Dowód. Oszacowanie górne wynika bezpośrednio z Lematu 7.1, bowiem dla $f \in C_M^{r+1}([a, b])$ mamy

$$\begin{aligned} \|f - w_{f,r}\|_C &= \max_{a \leq x \leq b} |f(x) - w_{f,r}(x)| \\ &= \max_{a \leq x \leq b} |(x - x_0) \cdots (x - x_r)| \frac{|f^{(r+1)}(\xi(x))|}{(r + 1)!} \\ &\leq \frac{M}{(r + 1)!} \max_{x \in D} |(x - x_0) \cdots (x - x_r)|. \end{aligned}$$

Z drugiej strony zauważmy, że np. dla wielomianu

$$p(x) = \frac{M}{(r + 1)!} \prod_{i=0}^r (x - x_i) \quad (7.3)$$

mamy $p \in C_M^{r+1}([a, b])$ i $p^{(r+1)} = 1$, a stąd

$$\|p - w_{p,r}\|_C = \frac{M}{(r+1)!} \max_{a \leq x \leq b} |(x - x_0) \cdots (x - x_r)|,$$

co kończy dowód. \square

Maksymalny błąd interpolacji wielomianowej w istotny sposób zależy od wyboru węzłów x_j . Naturalne jest więc teraz następujące pytanie. W których punktach x_j przedziału $[a, b]$ należy obliczać wartości funkcji, aby ten błąd był minimalny? Problem sprowadza się oczywiście do minimalizacji wielkości

$$\max_{a \leq x \leq b} |(x - x_0) \cdots (x - x_r)| \quad (7.4)$$

ze względu na węzły x_j .

Twierdzenie 7.2 *Najmniejsza wartość wyrażenia (7.4) wynosi $2\left(\frac{b-a}{4}\right)^{r+1}$ i jest przyjmowana dla węzłów*

$$x_j^* = \frac{a+b}{2} - \frac{b-a}{2} \cos\left(\frac{j+1/2}{r+1} \pi\right), \quad 0 \leq j \leq r.$$

Stąd, dla interpolacji $w_{r,f}^* \in \Pi_r$ opartej na węzłach x_j^* mamy

$$\sup_{f \in C_M^{r+1}([a,b])} \|f - w_{r,f}^*\|_C = \frac{2M}{(r+1)!} \left(\frac{b-a}{4}\right)^{r+1}.$$

Dowód tego twierdzenia opiera się na własnościach pewnego ważnego ciągu wielomianów, który teraz przedstawimy.

7.2 Wielomiany Czebyszewa

Ciąg $\{T_k\}_{k \geq 0}$ wielomianów Czebyszewa (a właściwie wielomianów Czebyszewa pierwszego rodzaju, w odróżnieniu od wielomianów Czebyszewa drugiego rodzaju, które poznamy w Rozdziale 9) zdefiniowany jest indukcyjnie jako

$$\begin{aligned} T_0(x) &= 1, & T_1(x) &= x, \\ T_k(x) &= 2xT_{k-1}(x) - T_{k-2}(x), & \text{dla } k &\geq 2. \end{aligned}$$

Zauważmy, że T_k jest wielomianem stopnia dokładnie k o współczynniku przy x^k równym 2^{k-1} (dla $k \geq 1$). Ponadto wielomian T_k można dla $|x| \leq 1$ przedstawić w postaci

$$T_k(x) = \cos(k \arccos x). \quad (7.5)$$

Rzeczywiście, łatwo sprawdzić, że jest to prawdą dla $k = 0, 1$. Stosując podstawienie $\cos t = x$, $0 \leq t \leq \pi$, oraz wzór na sumę cosinusów otrzymujemy dla $k \geq 1$

$$\cos((k+1)t) = 2 \cos t \cos(kt) - \cos((k-1)t),$$

co jest równoważne formule rekurencyjnej dla T_{k+1} .

Ze wzoru (7.5) wynikają również inne ważne własności wielomianów Czebyszewa. Norma wielomianu Czebyszewa na $[-1, 1]$ wynosi

$$\|T_k\|_{C([-1,1])} := \max_{-1 \leq x \leq 1} |T_k(x)| = 1$$

i jest osiągnięta w $k + 1$ punktach tego przedziału równych

$$y_j = -\cos\left(\frac{j}{k}\pi\right), \quad 0 \leq j \leq k, \quad (7.6)$$

przy czym $T_k(y_j) = (-1)^j$. W końcu, k -ty wielomian Czebyszewa T_k ma dokładnie k pojedynczych zer w $[-1, 1]$ równych

$$z_j = -\cos\left(\frac{2j+1}{2r}\pi\right), \quad 0 \leq j \leq k-1.$$

Konsekwencją wymienionych własności jest następujący lemat, który jest kluczem do dowodu Twierdzenia 7.2. Przez $\bar{\Pi}_k$ oznaczmy klasę wielomianów stopnia k o współczynniku wiodącym równym 1, tzn.

$$\bar{\Pi}_k = \{w \in \Pi_k : w(x) = x^k + \dots\}.$$

Lemat 7.2 *Niech $k \geq 1$. W klasie $\bar{\Pi}_k$ minimalną normę jednostajną na przedziale $[-1, 1]$ ma wielomian $w^* = 2^{1-k}T_k$, tzn.*

$$\min_{w \in \bar{\Pi}_k} \|w\|_{C([-1,1])} = \|w^*\|_{C([-1,1])} = \frac{1}{2^{k-1}}.$$

Dowód. Zauważmy najpierw, że $w^* \in \bar{\Pi}_k$ oraz $\|w^*\|_{C([-1,1])} = 2^{1-k}$. Wystarczy więc pokazać, że norma każdego wielomianu z $\bar{\Pi}_k$ jest nie mniejsza niż 2^{1-k} .

Założmy, że jest przeciwnie, tzn. istnieje wielomian $w \in \bar{\Pi}_k$ taki, że

$$\|w\|_{C([-1,1])} < \frac{1}{2^{k-1}} = \|w^*\|_{C([-1,1])}.$$

Rozpatrzmy funkcję $\psi = w^* - w$. Ponieważ dla punktów “maksymalnych” zdefiniowanych w (7.6) mamy $w^*(y_{k-j}) = (-1)^j 2^{1-k}$ oraz $|w(y_{k-j})| < 2^{1-k}$, to

$$\psi(y_{k-j}) \begin{cases} > 0, & j\text{-parzyste,} \\ < 0, & j\text{-nieparzyste,} \end{cases} \quad 0 \leq j \leq k.$$

Stąd ψ ma co najmniej jedno zero w każdym z przedziałów (y_i, y_{i+1}) dla $0 \leq i \leq k-1$, czyli w sumie k zer. Z drugiej strony, ψ jest wielomianem stopnia co najwyżej $k-1$ (bo współczynniki przy x^k w wielomianach w^* i w redukują się), a więc $\psi = 0$ i $w^* = w$. \square

Dowód Twierdzenia 7.2 wynika teraz bezpośrednio z Lematu 7.2. Zauważmy bowiem, że wielomian $(x-x_0)(x-x_1)\cdots(x-x_r)$ jest w klasie $\bar{\Pi}_{r+1}$. Stąd dla $[a, b] = [-1, 1]$ optymalnymi węzłami są zera z_j wielomianu Czebyszewa, przy których

$$(x-z_0)(x-z_1)\cdots(x-z_r) = \frac{T_{r+1}(x)}{2^r}.$$

Jeśli przedział $[a, b]$ jest inny niż $[-1, 1]$, należy dokonać liniowej zamiany zmiennych tak, aby był przekształcony na $[-1, 1]$. Bezpośrednie sprawdzenie pokazuje, że w klasie $\overline{\Pi}_{r+1}$ minimalną normę Czebyszewa na przedziale $[a, b]$ ma wielomian

$$w_{a,b}^*(x) = \left(\frac{b-a}{2}\right)^{r+1} w^*\left(\frac{2x - (a+b)}{b-a}\right).$$

Stąd

$$\|w_{a,b}^*\|_C = \left(\frac{b-a}{2}\right)^{r+1} \frac{1}{2^r} = 2 \left(\frac{b-a}{4}\right)^{r+1}$$

i węzły x_j^* są optymalne. \square

7.3 Optymalna aproksymacja wielomianowa

Zdefiniujemy n -ty optymalny błąd aproksymacji wielomianowej dla danej funkcji $f \in C([a, b])$ jako

$$E_n(f) := \inf_{w \in \Pi_n} \|f - w\|_C. \quad (7.7)$$

Wobec tego, że przestrzeń wszystkich wielomianów dowolnego stopnia jest gęsta w $C([0, 1])$, co jest treścią znanego *twierdzenia Weierstrassa*, mamy

$$\lim_{n \rightarrow \infty} E_n(f) = 0.$$

Wielomian $w_{f,n}^* \in \Pi_n$ nazwiemy n -tym wielomianem optymalnym dla f jeśli $\|f - w_{f,n}^*\|_C = E_n(f)$.

Zauważmy, że wielomian optymalny istnieje. Rzeczywiście, ponieważ $E_n(f) \leq \|f - 0\|_C = \|f\|_C$ i z drugiej strony dla w o normie większej niż $2\|f\|_C$ mamy $\|f - w\|_C \geq \|w\|_C - \|f\|_C > \|f\|_C$, infimum w (7.7) może być równoważnie wzięte po domkniętej kuli o promieniu $2\|f\|$. Ponieważ $\dim \Pi_n = n + 1 < +\infty$, kula ta jest zwarta w Π_n i funkcja $w \mapsto \|f - w\|_C$, jako funkcja ciągła, przyjmuje w niej minimum.

Wielomian optymalny jest też jednoznacznie wyznaczony. Fakt ten wynika w szczególności z następującego twierdzenia, zwanego *twierdzeniem Czebyszewa o alternansie* i jest treścią Ćw. 7.10. Twierdzenie podaje pewną charakteryzację wielomianów optymalnych.

Twierdzenie 7.3 *Wielomian $w \in \Pi_n$ jest n -tym wielomianem optymalnym dla $f \in C([a, b])$ wtedy i tylko wtedy gdy istnieją $\sigma \in \{-1, +1\}$ oraz punkty $a \leq x_0 < x_1 < \dots < x_n < x_{n+1} \leq b$ (zwane alternanssem) takie, że*

$$f(x_i) - w(x_i) = \sigma(-1)^i \|f - w\|_C \quad \text{dla } 0 \leq i \leq n + 1. \quad (7.8)$$

Dowód. Załóżmy, że warunek (7.8) jest spełniony dla pewnego wielomianu $w \in \Pi_n$ i punktów x_i , $0 \leq i \leq n + 1$, przy czym, bez zmniejszenia ogólności, $\sigma = 1$. Jeśli w nie jest wielomianem optymalnym dla f to istnieje $\hat{w} \in \Pi_n$ taki, że $f(x_i) - \hat{w}(x_i) < \|f - w\|_C = f(x_i) - w(x_i)$ dla i parzystych, oraz $f(x_i) - \hat{w}(x_i) > -\|f - w\|_C = f(x_i) - w(x_i)$ dla i nieparzystych. Wtedy $(f - w) - (f - \hat{w}) = \hat{w} - w$ jest wielomianem w Π_n , który przyjmuje naprzemienne znaki w kolejnych punktach x_i , ma więc co najmniej jedno zero w każdym z otwartych przedziałów (x_i, x_{i+1}) , $0 \leq i \leq n$. Ale to oznacza, że $\hat{w} - w$ zeruje się w co najmniej $n + 1$ różnych punktach i w konsekwencji $\hat{w} = w$.

Teraz pokażemy, że jeśli w jest n -tym wielomianem optymalnym dla f to alternans istnieje. W tym celu, wybierzmy najpierw skończony ciąg punktów $\{x_i\}$ w następujący sposób. Jako x_1 bierzemy pierwszy punkt w $[a, b]$ spełniający $|(f - w)(x_1)| = \|f - w\|_C$. Możemy założyć bez zmniejszenia ogólności, że $(f - w)(x_1) = -\|f - w\|_C$. Jako x_2 bierzemy pierwszy punkt w $[x_1, b]$ spełniający $(f - w)(x_2) = \|f - w\|_C$, itd. Ogólnie, x_i jest pierwszym punktem w $[x_{i-1}, b]$ takim, że $(f - w)(x_i) = (-1)^i \|f - w\|_C$.

Jeśli w ten sposób wybraliśmy $n + 2$ punktów to są one poszukiwanym alternansem. Przypuśćmy więc, że zdołaliśmy wybrać tylko $k \leq n + 1$ punktów. Wtedy wybieramy dodatkowo y_i dla $1 \leq i \leq k - 1$ jako ostatni punkt w $[x_i, x_{i+1}]$ dla którego $(f - w)(y_i) = (-1)^i \|f - w\|_C$. Oczywiście $x_i \leq y_i < x_{i+1}$. Definiujemy wielomian

$$v(x) = (-1)^k a (x - z_2)(x - z_3) \cdots (x - z_k), \quad z_i = (y_{i-1} + x_i)/2,$$

gdzie $a > 0$ jest tak dobrane, aby $\|v\|_C < \|f - w\|_C$. Wielomian v jest w Π_n i ma tę własność, że $(f - w)(x)v(x) > 0$ dla wszystkich x takich, że $|(f - w)(x)| = \|f - w\|_C$. Niech

$$A = \{x \in [a, b] : (f - w)(x)v(x) \leq 0\}.$$

Jeśli A jest zbiorem pustym to $\|f - (w + v)\|_C = \|(f - w) - v\|_C < \|f - w\|_C$, czyli w nie jest optymalny. W przeciwnym wypadku, definiujemy

$$m = \max_{x \in A} |(f - w)(x)| < \|f - w\|_C$$

(gdzie ostatnia nierówność wynika ze zwartości A) oraz

$$v_1 = \left(1 - \frac{m}{\|f - w\|_C}\right)v.$$

Wtedy $|(f - w)(x) - v_1(x)| < \|f - w\|_C$ dla wszystkich $x \in [a, b]$ i w konsekwencji $\|f - (w + v_1)\|_C < \|f - w\|_C$. Rzeczywiście, jest to oczywiście prawdą dla $x \notin A$, a dla $x \in A$ mamy

$$|(f - w)(x) - v_1(x)| \leq |(f - w)(x)| + |v_1(x)| < m + \left(1 - \frac{m}{\|f - w\|_C}\right)\|f - w\|_C = \|f - w\|_C. \quad \square$$

Zadanie wyznaczenia wielomianu optymalnego, a dokładniej, jego współczynników w wybranej bazie, jest dużo trudniejsze niż interpolacja wielomianowa na ustalonych węzłach, bowiem wielomian optymalny $w_{f,n}^*$ zależy nieliniowo od funkcji f . Pomocą może tu służyć twierdzenie o alternansie. Na przykład, *algorytmy Remeza* znajdują ten wielomian metodą kolejnych przybliżeń alternansu, są one jednak dość kosztowne, zob. U. 7.2.

Wróćmy do interpolacji wielomianowej i zobaczmy jak daleko jest wielomian interpolacyjny od wielomianu optymalnego. W tym celu, rozpatrzmy dowolne węzły

$$a \leq x_{n,0} < x_{n,1} < \cdots < x_{n,n} \leq b \tag{7.9}$$

i odpowiadające im operatory interpolacyjne Lagrange'a $L_n : C([a, b]) \rightarrow \Pi_n$,

$$(L_n f)(x) = \sum_{i=0}^n f(x_{n,i}) l_{n,i}(x), \quad l_{n,i}(x) = \prod_{j \neq i} \frac{x - x_{n,j}}{x_{n,j} - x_{n,i}}.$$

Każdy z tych operatorów jest liniowy i ciągły. Łatwo sprawdzić, że jego norma wynosi

$$\|L_n\| := \sup_{0 \neq f \in C([a,b])} \frac{\|L_n f\|_C}{\|f\|_C} = \Lambda(x_{n,0}, x_{n,1}, \dots, x_{n,n}),$$

$$\text{gdzie} \quad \Lambda(x_0, x_1, \dots, x_n) := \max_{a \leq x \leq b} \sum_{i=0}^n |l_{n,i}(x)|.$$

Okazuje się, że wielkość normy $\|L_n\|$ jest kluczowa dla jakości interpolacji. Wykorzystując równość $L_n w = w$ dla $w \in \Pi_n$, mamy bowiem

$$\|f - L_n f\|_C \leq \|f - w_{n,f}^*\|_C + \|L_n w_{n,f}^* - L_n f\|_C \leq (1 + \|L_n\|) E_n(f).$$

Stąd, jeśli normy $\|L_n\|$ są wspólnie ograniczone to błąd interpolacji zbiega do zera co najmniej tak szybko jak n -ty błąd optymalny $E_n(f)$. Okazuje się że, niestety, niezależnie od wyboru punktów interpolacyjnych mamy $\lim_{n \rightarrow \infty} \|L_n\| = +\infty$. Co więcej, istnieją funkcje ciągłe f takie, że ciąg $\{\|L_n f\|\}_{n=0}^{\infty}$ jest nieograniczony.

Ale są też dobre wiadomości. Niech

$$\Lambda_n := \inf_{a \leq x_0 < \dots < x_n \leq b} \Lambda(x_0, \dots, x_n)$$

będzie minimalną normą operatora interpolacji opartej na $n + 1$ węzłach. Zauważmy, że Λ_n , zwana *stałą Lebesgue'a*, nie zależy od przedziału $[a, b]$, zob. Ćw. 7.12.

Lemat 7.3 *Mamy*

$$\lim_{n \rightarrow \infty} \frac{\Lambda_n}{\ln(n)} = \frac{2}{\pi}$$

Dodajmy, że optymalna asymptotyka z Lematu 7.3 jest osiągana, np. przez węzły Czebyszewa $\{x_i^*\}_{i=0}^n$ z Twierdzenia 7.2, albo przez ‘rozszerzone’ węzły Czebyszewa

$$\bar{x}_j^* = \frac{a+b}{2} - \frac{b-a}{2 \cos\left(\frac{\pi}{2(r+1)}\right)} \cos\left(\frac{2j+1}{2(r+1)} \pi\right), \quad 0 \leq j \leq r, \quad (7.10)$$

zawierające oba końce przedziału $[a, b]$. Natomiast złym pomysłem, przynajmniej dla ‘dużych’ n , jest interpolacja oparta na węzłach równoodległych, bowiem wtedy norma operatora interpolacji rośnie jak $2^{n+1}/(e n \ln(n))$. Jest to m.in. przyczyną tzw. *efektu Rungego*, zob. U. 7.3.

Podamy jeszcze jedną z wersji dobrze znanego i bardzo ważnego w teorii aproksymacji *twierdzenia Jacksona* uzależniającego błąd optymalnej aproksymacji wielomianowej od stopnia gładkości funkcji aproksymowanej. Twierdzenie to, jak i kilka wcześniej wypowiedzianych faktów, pozostawiamy bez dowodów, gdyż wykraczają one poza ramy tego kursu.

Twierdzenie 7.4 *Mamy*

$$\sup_{f \in C_M^{r+1}([a,b])} E_n(f) = C_r M n^{-(r+1)},$$

gdzie C_r nie zależy od n i M .

Wniosek 7.1 *Jeśli $f \in C^{r+1}([a, b])$ to błąd interpolacji tej funkcji, opartej na $n + 1$ węzłach Czebyszewa, zbiega do zera co najmniej tak szybko jak $\ln(n) n^{-(r+1)}$.*

7.4 Interpolacja kawałkami wielomianowa

Interpolacja wielomianowa daje teoretycznie dobre przybliżenie funkcji o ile funkcja jest odpowiednio regularna, a węzłów interpolacyjnych jest odpowiednio dużo i są odpowiednio wybrane. Jednak wraz ze wzrostem stopnia wielomianu interpolacyjnego obliczenie jego współczynników jest coraz droższe, a błędy zaokrągleń powodują dla dużych n duże błędy w ich obliczaniu. Tych trudności można łatwo uniknąć rezygnując z tradycyjnej interpolacji wielomianowej na rzecz *interpolacji kawałkami wielomianowej*, zachowując jednocześnie optymalną szybkość zbieżności $n^{-(r+1)}$ dla funkcji $f \in C^{r+1}([a, b])$.

Postąpimy zgodnie ze znaną zasadą “dziel i rządź”. Podzielimy przedział $[a, b]$ na k równych podprzedziałów $[t_{i-1}, t_i]$, $1 \leq i \leq k$, i zastosujemy na każdym z nich interpolację wielomianową stopnia r . Dokładniej, dla każdego i wybieramy (być może wielokrotne) węzły

$$t_{i-1} = x_{i,0} \leq x_{i,1} \leq \dots \leq x_{i,r} = t_i,$$

po czym aproksymujemy f na przedziale $[t_{i-1}, t_i]$ jej wielomianem interpolacyjnym z Π_r opartym na tych węzłach. Tak skonstruowaną kawałkami wielomianową aproksymację funkcji $f : [a, b] \rightarrow \mathbb{R}$ oznaczymy przez $\bar{w}_{f,r,k}$. Zauważmy, że $\bar{w}_{f,r,k}$ wykorzystuje na całym $[a, b]$ węzły o łącznej krotności

$$n \leq kr + 1 \quad (7.11)$$

oraz, że $\bar{w}_{f,r,k} \in C([a, b])$.

Twierdzenie 7.5 *Maksymalny błąd aproksymacji kawałkami wielomianowej w klasie $C_M^{r+1}([a, b])$ jest ograniczony przez*

$$\sup_{f \in C_M^{r+1}([a, b])} \|f - \bar{w}_{f,r,k}\|_C \leq \frac{M}{(r+1)!} \left(\frac{b-a}{k}\right)^{r+1} \leq D_r M \left(\frac{1}{n-1}\right)^{r+1},$$

$$\text{gdzie} \quad D_r = \frac{r^{r+1}}{(r+1)!} (b-a)^{r+1}.$$

Jeśli ponadto węzły $x_{i,j}$, $0 \leq j \leq r$, są na każdym przedziale $[t_{i-1}, t_i]$ rozszerzonymi węzłami Czebyszewa (7.10) to

$$\sup_{f \in C_M^{r+1}([a, b])} \|f - \bar{w}_{f,r,k}\|_C = \frac{2M}{(r+1)!} \left(\frac{b-a}{4k \cos\left(\frac{\pi}{2(r+1)}\right)}\right)^{r+1} = D_r^* M \left(\frac{1}{n-1}\right)^{r+1},$$

$$\text{gdzie} \quad D_r^* = \frac{D_r}{2^{2r+1} \cos\left(\frac{\pi}{2(r+1)}\right)} = \frac{2}{(r+1)!} \left(\frac{r}{4 \cos\left(\frac{\pi}{2(r+1)}\right)}\right)^{r+1} (b-a)^{r+1}.$$

Dowód. Niech $f \in C_M^{r+1}([a, b])$ i $x \in [t_{i-1}, t_i]$. Z Lematu 7.1 wynika, że

$$|f(x) - \bar{w}_f(x)| = \frac{|f^{(r+1)}(\xi(x))|}{(r+1)!} |(x - x_{i,0})(x - x_{i,1}) \dots (x - x_{i,r})|,$$

a ponieważ $|x - x_{i,j}| \leq t_i - t_{i-1} = \frac{1}{k}(b-a)$, to

$$|f(x) - \bar{w}_{f,r,k}(x)| \leq \frac{M}{(r+1)!} \left(\frac{b-a}{k}\right)^{r+1}.$$

Ta nierówność wraz z (7.11) dowodzi pierwszej części twierdzenia.

Dla dowodu drugiej części zauważmy, że rozszerzone węzły Czebyszewa na przedziale $[a, b]$ są jednocześnie zwykłymi węzłami Czebyszewa na przedziale $[a - h, b + h]$, gdzie

$$h = \frac{b - a}{2} \left(\frac{1}{\cos\left(\frac{\pi}{2(r+1)}\right)} - 1 \right),$$

którego długość wynosi $2h = (b - a) / \cos\left(\frac{\pi}{2(r+1)}\right)$. Dlatego nierówność wynika z Twierdzenia 7.2, a równość jest osiągnięta dla każdego wielomianu postaci $p(x) = \frac{M}{(n+1)!} x^{r+1} + \dots$ \square

Otrzymaliśmy, że interpolacja kawałkami wielomianami stopnia r korzystająca z n wartości funkcji daje w klasie $C_M^{r+1}([a, b])$ błąd rzędu $n^{-(r+1)}$. Zauważmy, że szybkość zbieżności nie zależy od doboru ‘wewnętrznych’ węzłów $x_{i,j}$, $0 \leq j \leq r$.

Można pokazać, że aproksymacja kawałkami wielomianowa jest optymalna. Dokładniej maksymalny błąd w klasie $C_M^{r+1}([a, b])$ dowolnej innej metody aproksymacji korzystającej z wartości funkcji i/lub jej pochodnych w n punktach dąży do zera nie szybciej niż $n^{-(r+1)}$; cf. U. 7.5.

Uwagi i uzupełnienia

U. 7.1 Twierdzenie 7.3 o alternansie ma ogólniejszy charakter, pozostaje bowiem prawdziwe również w przypadku aproksymacji względem podprzestrzeni V innych niż Π_n . Formalnie, v_f jest elementem optymalnym dla $f \in C([a, b])$ względem V gdy $v_f \in V$ i $\|f - v_f\|_C = \inf_{v \in V} \|f - v\|_C$.

Załóżmy, że podprzestrzeń liniowa $V \subset C([a, b])$ jest *przestrzenią Haara*. To znaczy, że każda niezerowa funkcja $f \in V$ znika w co najwyżej $n - 1$ różnych punktach, gdzie $n = \dim V$. Wtedy warunkiem koniecznym i dostatecznym na to, aby $v \in V$ był elementem optymalnym dla funkcji $f \in C([0, 1])$ jest istnienie alternansu o liczebności $\dim V + 1$. Ponadto, element optymalny jest wyznaczony jednoznacznie.

Oczywiście, Π_n są przestrzeniami Haara, bo $\dim \Pi_n = n + 1$ i każdy wielomian stopnia nie większego niż n zeruje się w co najwyżej n punktach.

U. 7.2 Jeśli znamy alternans $x_0 < \dots < x_n < x_{n+1}$ to współczynniki c_j wielomianu optymalnego funkcji f w danej bazie $\{\phi_j\}_{j=0}^n$ można wyliczyć rozwiązując układ $(n + 2) \times (n + 2)$ równań liniowych

$$f(x_i) - \sum_{j=0}^n c_j \phi_j(x_i) = (-1)^i \varepsilon, \quad 0 \leq i \leq n + 1, \quad (7.12)$$

gdzie niewiadomymi są błąd ε i c_j , $0 \leq j \leq n$. Macierz tego układu jest nieosobliwa (dlaczego?). *Algoritmy Remeza* wykorzystują ten fakt do konstrukcji przybliżeń elementów optymalnych. Startując z pewnego początkowego przybliżenia alternansu, w kolejnych krokach rozwiązuje się układ (7.12), a następnie wymienia jeden punkt (albo od razu kilka punktów) aktualnego przybliżenia alternansu na z taki, że $|f(z) - \sum_{j=0}^n c_j \phi_j(z)| > |\varepsilon|$. (Jeśli taki punkt nie istnieje to mamy alternans.) Pokazuje się, że przy zachowaniu odpowiednich reguł wymiany mamy zbieżność do alternansu funkcji f .

U. 7.3 Niech $f(x) = (1 + x^2)^{-1}$. Okazuje się, że jeśli $w_{f,n}$ jest wielomianem interpolacyjnym dla f opartym na węzłach równoodległych przedziału $[-5, 5]$, tzn. $x_j = -5 + 10j/n$, $0 \leq j \leq n$, to ciąg $w_{f,n}(x)$ jest zbieżny do $f(x)$ dla $|x| < 3.63\dots$ i rozbieżny dla $|x| > 3.63\dots$. To znaczy, że błąd interpolacji funkcji f na całym przedziale $[-5, 5]$ rośnie do $+\infty$ wraz ze wzrostem stopnia wielomianu interpolacyjnego. Zjawisko to znane jest pod nazwą *efektu Rungego*.

Z drugiej strony, jeśli interpolację oprzemy na węzłach Czebyszewa

$$x_j^* = -5 \cos \left(\frac{j+1/2}{n+1} \pi \right), \quad 0 \leq j \leq n,$$

to błąd interpolacji nie tylko zbiega do zera, ale zbiega szybciej niż n^{-r} dla każdego r . Wynika to z Wniosku 7.1 oraz faktu, że $f \in C^{r+1}([-5, 5])$ dla każdego r .

U. 7.4 Wielomiany Bernsteina,

$$(B_n f)(x) = \sum_{i=0}^n f \left(\frac{k}{n} \right) \binom{n}{k} x^i (1-x)^{n-i},$$

są wygodnym narzędziem do konstruktywnego dowodu twierdzenia Weierstrassa w $C([0, 1])$, ale mają słabe własności aproksymacyjne. Łatwo się przekonać, że już dla wielomianów kwadratowych, $p(x) = ax^2 + \dots$ gdzie $a \neq 0$, mamy $(B_n p)(x) - p(x) = \frac{a}{n} x(1-x)$, czyli $\|B_n p - p\|_{C([0,1])} = \frac{|a|}{4n}$ i zbieżność jest tylko liniowa.

U. 7.5 Pokażemy teraz, że aproksymacja funkcjami kawałkami wielomianowymi jest optymalna w $C_M^{r+1}([a, b])$ ze względu na rząd zbieżności błędu. Dokładniej pokażemy, że dla dowolnej aproksymacji $A(f)$ funkcji f wykorzystującej wartości f i/lub jej pochodnych w co najwyżej n różnych punktach, tzn. dla A postaci

$$A(f) = \varphi(f(x_1), \dots, f^{(r+1)}(x_1), \dots, f(x_n), \dots, f^{(r+1)}(x_n)),$$

mamy

$$\sup_{f \in C_M^{r+1}([a,b])} \|f - A(f)\|_C \geq C M (n+1)^{-(r+1)}, \quad (7.13)$$

gdzie $C > 0$ jest pewnym czynnikiem niezależnym od A , M i n .

W tym celu, wybierzmy dowolną niezerową funkcję $\Psi : \mathbb{R} \rightarrow \mathbb{R}$ taką, że $\Psi(x) = 0$ dla $x \notin (0, 1)$, Ψ jest $(r+1)$ razy różniczkowalna w sposób ciągły, oraz $\|\Psi^{(r+1)}\|_{C([0,1])} = M$. Niech s będzie takim indeksem, że $h_s := x_{s+1} - x_s \geq \frac{b-a}{n+1}$. Wtedy dla funkcji ψ_i , $i = 1, 2$, zdefiniowanych jako

$$\psi_1(x) = -\psi_2(x) = M h_s^{r+1} \Psi \left(\frac{x - x_s}{h_s} \right), \quad a \leq x \leq b,$$

mamy, że $\psi_i \in C_M^{r+1}([a, b])$ oraz informacja o ψ_i jest zerowa, tzn. $\psi_i^{(p)}(x_j) = 0$ dla $0 \leq p \leq r+1$, $1 \leq j \leq n$. Stąd maksymalny błąd metody na $[a, b]$ jest ograniczony z dołu przez

$$\begin{aligned} \max \{ \|\psi_1 - \varphi(0)\|_C, \|\psi_2 - \varphi(0)\|_C \} &\geq \frac{1}{2} \|\psi_1 - \psi_2\|_C = \|\psi_1\|_C \\ &= M h_s^{r+1} \|\Psi\|_{C([0,1])} \geq M \left(\frac{b-a}{n+1} \right)^{r+1} \|\Psi\|_{C([0,1])}. \end{aligned}$$

Nierówność (7.13) zachodzi więc z $C = (b-a)^{r+1} \|\Psi\|_{C([0,1])}$.

U. 7.6 Załóżmy, że funkcje z klasy $C_M^{r+1}([a, b])$ chcemy aproksymować z dokładnością $\varepsilon > 0$ w normie Czebyszewa używając możliwie mało obliczeń wartości funkcji. Z Twierdzenia 7.5 i U. 7.5 wynika, że musimy wtedy obliczać co najmniej rzędu $(1/\varepsilon)^{r+1}$ wartości funkcji. Co więcej, aproksymacja kawałkami wielomianowa daje błąd ε obliczając właśnie tyle wartości funkcji. Mówimy, że *złożoność informacyjna* zadania jest rzędu $(1/\varepsilon)^{r+1}$.

U. 7.7 Obok aproksymacji jednostajnej w przestrzeni $C([a, b])$ stosuje się też, m.in., *aproksymację średniokwadratową* w przestrzeni $\mathcal{L}_{2,\rho}(a, b)$. Jest to przestrzeń unitarna funkcji mierzalnych i takich, że całka ważona $\int_a^b |f(x)|^2 \rho(x) dx < +\infty$, przy czym *waga* ρ jest całkowalna i dodatnia p.w. na (a, b) , a iloczyn skalarny i norma zdefiniowane są jako

$$\langle f, g \rangle = \int_a^b f(x)g(x)\rho(x) dx, \quad \|f\| = \sqrt{\langle f, f \rangle} = \sqrt{\int_a^b |f(x)|^2 \rho(x) dx}.$$

Niech $\{\phi_j\}_{j=0}^n$ będzie daną bazą przestrzeni Π_n . Zadanie znalezienia współczynników wielomianu optymalnego $w_{f,n}^*$ w bazie $\{\phi_j\}_{j=0}^n$ dla funkcji $f \in \mathcal{L}_{2,\rho}(a, b)$ w sensie aproksymacji średniokwadratowej jest stosunkowo proste. Wiadomo bowiem, że w przestrzeniach unitarnych element optymalny jest rzutem prostokątnym f na Π_n . To zaś jest równoważne prostokątności różnicy $f - w_{f,n}^*$ do elementów bazowych ϕ_i . Rozpisując $w_{f,n}^* = \sum_{j=0}^n c_j \phi_j$ mamy

$$\left\langle f - \sum_{j=0}^n c_j \phi_j, \phi_i \right\rangle = 0$$

albo równoważnie

$$\sum_{j=0}^n c_j \langle \phi_j, \phi_i \rangle = \langle f, \phi_i \rangle \quad \text{dla } 0 \leq i \leq n.$$

Zadanie sprowadza się więc do rozwiązania układu równań z macierzą $G = (\langle \phi_j, \phi_i \rangle)_{i,j=0}^n$ i niewiadomymi współczynnikami c_j . Macierz G , zwana *macierzą Grama*, jest symetryczna i dodatnio określona, przy czym dodatnia określoność prosto wynika stąd, że dla dowolnego wektora $\vec{c} \neq \vec{0}$ mamy $\langle G\vec{c}, \vec{c} \rangle_2 = \|\sum_{j=0}^n c_j \phi_j\|_2^2 > 0$.

Oczywiście, jeśli $\{\phi_j\}_{j=0}^n$ jest bazą ortogonalną to G jest macierzą diagonalną i

$$w = \sum_{j=0}^n \frac{\langle f, \phi_j \rangle}{\langle \phi_j, \phi_j \rangle} \phi_j.$$

O bazach ortogonalnych w przestrzeniach Π_n i ich własnościach dowiemy się więcej w Rozdziale 10.

Ćwiczenia

Ćw. 7.1 Zauważ, że Twierdzenie 7.1 zachodzi w nieco szerszej niż $C_M^{r+1}([a, b])$ klasie funkcji $f \in C^r([a, b])$ takich, że $f^{(r)}$ spełnia warunek Lipschitza ze stałą M .

Ćw. 7.2 Wykaż, że optymalne węzły x_j^* w Twierdzeniu 7.2 są wyznaczone jednoznacznie.

Ćw. 7.3 Niech dane będą $\varepsilon > 0$ i funkcja $f : [a, b] \rightarrow \mathbb{R}$, która jest nieskończenie wiele razy różniczkowalna i wszystkie jej pochodne na $[a, b]$ są wspólnie ograniczone przez M . Napisz program znajdujący stopień n oraz współczynniki w bazie Newtona wielomianu $w_{f,n} \in \Pi_n$ interpolacyjnego f z błędem

$$\|f - w_{f,n}\|_C \leq \varepsilon.$$

Rozpatrz dwa przypadki: gdy węzły interpolacji są równoodległe, oraz gdy węzły są czebyszewowskie.

Ćw. 7.4 Pokaż, że dla $k \geq 1$ mamy

$$T_k(x) = \frac{(x + \sqrt{x^2 - 1})^k + (x - \sqrt{x^2 - 1})^k}{2},$$

gdzie T_k jest k -tym wielomianem Czebyszewa.

Ćw. 7.5 Wielomiany Czebyszewa $\{T_k\}_{k=0}^n$ tworzą bazę w przestrzeni Π_n . Podaj algorytm obliczający współczynniki danego wielomianu w bazie Czebyszewa na podstawie jego współczynników w bazie potęgowej, albo w bazie Newtona.

Ćw. 7.6 Podaj algorytm typu algorytmu Hornera obliczający wartość wielomianu $w(x)$ danego przez jego współczynniki w bazie Czebyszewa $\{T_k\}_{k \geq 0}$.

Ćw. 7.7 Niech w będzie wielomianem, którego rozwinięciem w bazie Czebyszewa jest $w(x) = \sum_{k=0}^n a_k T_k(x)$. Pokaż, że wielomian

$$w_1(x) = \sum_{k=0}^{n-1} a_k T_k(x)$$

najlepiej przybliża w w normie Czebyszewa na przedziale $[-1, 1]$ spośród wszystkich wielomianów stopnia co najwyżej $n-1$, oraz

$$\|w - w_1\|_C = |a_n|.$$

Czy $w_2(x) = \sum_{k=0}^{n-2} a_k T_k(x)$ najlepiej przybliża w w tej samej normie w przestrzeni Π_{n-2} ?

Ćw. 7.8 Załóżmy, że wielomian $w(x) = \sum_{k=0}^n a_k T_k(x)$ aproksymujemy wielomianem $w_i(x) = \sum_{k=0}^{n-i} a_k T_k(x)$. Uzasadnij, że

$$\|w - w_i\|_C \leq \sum_{k=n-i+1}^n |a_k|.$$

Kiedy w tej nierówności mamy równość?

Ćw. 7.9 Niech dany będzie przedział skończony $[a, b]$ nie zawierający zera, oraz liczba c . Niech

$$\tilde{\Pi}_k = \{w \in \Pi_k : w(0) = c\}.$$

Pokaż, że w klasie $\tilde{\Pi}_k$ najmniejszą normę Czebyszewa na $[a, b]$ ma wielomian

$$w(x) = c \frac{T_k\left(\frac{2x-(a+b)}{b-a}\right)}{T_k\left(\frac{a+b}{a-b}\right)} \quad \text{oraz} \quad \|w\|_C = \frac{|c|}{\left|T_k\left(\frac{a+b}{a-b}\right)\right|}.$$

Jakie jest rozwiązanie, gdy $0 \in [a, b]$?

Ćw. 7.10 Wykaż, że n -ty wielomian optymalny $w_{f,n}^*$ jest jednoznacznie wyznaczony.

Ćw. 7.11 Wykaż, że dla każdej funkcji $f \in C([a, b])$ można wskazać węzły $a < x_0 < \dots < x_n < b$ takie, że wielomian interpolacyjny funkcji f oparty na tych węzłach jest jednocześnie jej n -tym wielomianem optymalnym.

Ćw. 7.12 Uzasadnij, że stała Lebesgue'a Λ_n nie zależy od wyboru przedziału $[a, b]$.

Ćw. 7.13 Pokaż, że przestrzeń funkcji trygonometrycznych stopnia n zdefiniowana jako

$$\mathcal{V}_{2n+1} := \text{span}(1, \cos t, \sin t, \dots, \cos nt, \sin nt)$$

jest przestrzenią Haara w $C([a, b])$, o ile $0 < b - a < 2\pi$.

Ćw. 7.14 Pokaż, że

$$\hat{\mathcal{V}}_{n+1} := \text{span}(1, \cos t, \cos 2t, \dots, \cos nt)$$

jest przestrzenią Haara w $C([0, \pi])$.

Rozdział 8

Interpolacja funkcjami sklejanymi

Interpolacja kawałkami wielomianami interpolacyjnymi daje mały błąd aproksymacji, ma jednak również pewne wady. Zauważmy, że w punktach t_j podziału przedziału $[a, b]$ na podprzedziały funkcja aproksymująca $\bar{w}_{f,r,k}$ nie jest w ogólności różniczkowalna. Ta własność jest niepożądana w przypadku gdy wiemy, że funkcja którą aproksymujemy jest “gładka”. Wtedy możemy zastosować innego rodzaju interpolację, mianowicie interpolację splajnową, która wymusza zadaną regularność na całym przedziale $[a, b]$,

8.1 Co to są funkcje sklepane?

W ogólności, przez funkcję sklepaną rozumiemy każdą funkcję przedziałami wielomianową. Nas będą jednak interesować szczególne funkcje tego typu i dlatego termin *funkcje sklepane* zarezerwujemy dla funkcji przedziałami wielomianowych i posiadających dodatkowe własności, które teraz określimy.

Niech dany będzie przedział skończony $[a, b]$ i węzły

$$a = x_0 < x_1 < \dots < x_n = b,$$

przy czym $n \geq 1$.

Definicja 8.1 *Funkcję $s : \mathbb{R} \rightarrow \mathbb{R}$ nazywamy funkcją sklepaną rzędu r ($r \geq 1$) odpowiadającą węzłom x_j dla $0 \leq j \leq n$, jeśli spełnione są następujące dwa warunki:*

(i) *s jest wielomianem stopnia co najwyżej $2r - 1$ na każdym z przedziałów $[x_{j-1}, x_j]$, tzn. $s|_{[x_{j-1}, x_j]} \in \Pi_{2r-1}$ dla $1 \leq j \leq n$,*

(ii) *s jest $(2r - 2)$ -krotnie różniczkowalna w sposób ciągły na całej prostej, tzn. $s \in C^{2r-2}(\mathbb{R})$.*

Jeśli ponadto

(iii) *s jest wielomianem stopnia co najwyżej $r - 1$ poza (a, b) , tzn. $s|_{(-\infty, a]}, s|_{[b, +\infty)} \in \Pi_{r-1}$,*

to s jest naturalną funkcją sklepaną.

Klasę naturalnych funkcji sklepanych rzędu r opartych na węzłach x_j będziemy oznaczać przez $\mathcal{S}_r(x_0, \dots, x_n)$, albo po prostu \mathcal{S}_r , jeśli węzły są ustalone.

Na przykład, funkcją sklejaną rzędu pierwszego ($r = 1$) jest funkcja ciągła będąca wielomianem pierwszego stopnia na poszczególnych przedziałach $[x_{j-1}, x_j]$. Jest ona naturalna, gdy poza (a, b) jest funkcją stałą. Tego typu funkcje nazywamy *liniowymi funkcjami sklejanymi*.

Najważniejszymi z praktycznego punktu widzenia są jednak funkcje sklepane rzędu drugiego odpowiadające $r = 2$. Są to funkcje, które są na \mathbb{R} dwa razy różniczkowalne w sposób ciągły, a na każdym z podprzedziałów są wielomianami stopnia co najwyżej trzeciego. W tym przypadku mówimy o *kubicznych funkcjach sklepanych*. Funkcja sklejana kubiczna jest naturalna, gdy poza (a, b) jest wielomianem pierwszego stopnia.

8.2 Interpolacja i gładkość

Pokażemy najpierw ważny lemat, który okaże się kluczem do dowodu dalszych twierdzeń.

Niech $W^r(a, b)$ będzie klasą funkcji $f : [a, b] \rightarrow \mathbb{R}$ takich, że f jest $(r - 1)$ razy różniczkowalna na $[a, b]$ w sposób ciągły oraz $f^{(r)}(x)$ istnieje prawie wszędzie na $[a, b]$ i jest całkowalna z kwadratem, tzn.

$$W^r(a, b) = \left\{ f \in C^{r-1}([a, b]) : f^{(r)}(x) \text{ istnieje p.w. na } [a, b] \text{ oraz } f^{(r)} \in \mathcal{L}_2(a, b) \right\}.$$

Oczywiście każda funkcja sklejana rzędu r (niekoniecznie naturalna) należy do $W^r(a, b)$.

Lemat 8.1 Niech $f \in W^r(a, b)$ będzie funkcją zerującą się w węzłach, tzn.

$$f(x_j) = 0 \quad \text{dla} \quad 0 \leq j \leq n.$$

Wtedy dla dowolnej naturalnej funkcji sklepanej $s \in \mathcal{S}_r$ mamy

$$\int_a^b f^{(r)}(x)s^{(r)}(x) dx = 0.$$

Dowód Dla $r = 1$ funkcja s' jest przedziałami stała. Oznaczając przez a_j jej wartość na $[x_{j-1}, x_j]$ dostajemy

$$\int_a^b f'(x)s'(x) dx = \sum_{j=1}^n \int_{t_{j-1}}^{t_j} f'(x)a_j dx = \sum_{j=1}^n a_j(f(x_j) - f(x_{j-1})) = 0,$$

ponieważ f zeruje się w t_j .

Rozpatrzmy teraz przypadek $r \geq 2$. Ponieważ

$$(f^{(r-1)}s^{(r)})' = f^{(r)}s^{(r)} + f^{(r-1)}s^{(r+1)},$$

to

$$\int_a^b f^{(r)}(x)s^{(r)}(x) dx = f^{(r-1)}(x)s^{(r)}(x) \Big|_a^b - \int_a^b f^{(r-1)}(x)s^{(r+1)}(x) dx.$$

Wobec tego, że s jest poza przedziałem (a, b) wielomianem stopnia co najwyżej $r - 1$ oraz $s^{(r)}$ jest ciągła na \mathbb{R} , mamy $s^{(r)}(a) = 0 = s^{(r)}(b)$, a stąd

$$f^{(r-1)}(x)s^{(r)}(x) \Big|_a^b = 0.$$

Postępując podobnie, tzn. całkując przez części $r - 1$ razy otrzymujemy w końcu

$$\int_a^b f^{(r)}(x)s^{(r)}(x) dx = (-1)^{r-1} \int_a^b f'(x)s^{(2r-1)}(x) dx.$$

Funkcja $s^{(2r-1)}$ jest przedziałami stała, a więc możemy teraz zastosować ten sam argument jak dla $r = 1$, aby pokazać, że ostatnia całka jest równa zero. \square

Funkcje sklejjane chcielibyśmy zastosować do interpolacji funkcji. Ważne jest więc, aby odpowiednie zadanie interpolacyjne miało jednoznaczne rozwiązanie.

Twierdzenie 8.1 *Jeśli $n + 1 \geq r$ to dla dowolnej funkcji $f : [a, b] \rightarrow \mathbb{R}$ istnieje dokładnie jedna naturalna funkcja sklejjana $s_f \in \mathcal{S}_r$ interpolująca f w węzłach x_j , tzn. taka, że*

$$s_f(x_j) = f(x_j), \quad 0 \leq j \leq n.$$

Dowód. Pokażemy najpierw, że jedyną naturalną funkcją sklejjaną interpolującą dane zerowe jest funkcja zerowa. Rzeczywiście, jeśli $s(x_j) = 0$ dla $0 \leq j \leq n$, to podstawiając $f = s$ w Lemacie 8.1 otrzymujemy

$$\int_a^b (s^{(r)}(x))^2 dx = 0.$$

Stąd $s^{(r)}$ jest funkcją zerową, a więc s jest wielomianem stopnia co najwyżej $r - 1$ zerującym się w co najmniej $n + 1$ punktach x_j . Wobec tego, że $n + 1 > r - 1$, otrzymujemy $s \equiv 0$.

Zauważmy teraz, że problem znalezienia naturalnej funkcji sklejjanej s interpolującej f można sprowadzić do rozwiązania kwadratowego układu równań liniowych. Na każdym przedziale $[x_{i-1}, x_i]$, $1 \leq i \leq n$, jest ona postaci

$$s(x) = w_i(x) = \sum_{j=0}^{2r-1} a_{i,j} x^j,$$

dla pewnych współczynników $a_{i,j} \in \mathbb{R}$, a na $(-\infty, a]$ i $[b, \infty)$ mamy odpowiednio

$$s(x) = w_0(x) = \sum_{j=0}^{r-1} a_{0,j} x^j \quad \text{i} \quad s(x) = w_{n+1}(x) = \sum_{j=0}^{r-1} a_{n+1,j} x^j.$$

Aby wyznaczyć s , musimy więc znaleźć ogółem $2r(n + 1)$ współczynników $a_{i,j}$, przy czym są one związane $(2r - 1)(n + 1)$ warunkami jednorodnymi wynikającymi z gładkości,

$$w_i^{(k)}(x_i) = w_{i+1}^{(k)}(x_i)$$

dla $0 \leq i \leq n$ i $0 \leq k \leq 2r - 2$, oraz $n + 1$ niejednorodnymi warunkami interpolacyjnymi,

$$w_i(x_i) = f(x_i)$$

dla $0 \leq i \leq n$. Otrzymujemy więc układ $2r(n + 1)$ równań liniowych z $2r(n + 1)$ niewiadomymi $a_{i,j}$.

Naturalna funkcja sklejjana interpolująca f jest wyznaczona jednoznacznie wtedy i tylko wtedy, gdy układ ten ma jednoznaczne rozwiązanie. To zaś zachodzi, gdy zero jest jedynym rozwiązaniem układu jednorodnego. Rzeczywiście, układ jednorodny odpowiada zerowym warunkom interpolacyjnym, przy których, jak pokazaliśmy wcześniej, zerowa funkcja sklejjana (której odpowiada $a_{i,j} = 0$ dla wszystkich i, j) jest jedynym rozwiązaniem zadania interpolacyjnego. \square

Naturalne funkcje sklejjane możemy więc używać do interpolacji funkcji. Pokażemy teraz inną ich własność, która jest powodem dużego praktycznego zainteresowania funkcjami sklejjanymi.

Twierdzenie 8.2 Niech $f \in W^r(a, b)$ i niech $s_f \in \mathcal{S}_r$ będzie naturalną funkcją sklejaną rzędu r interpolującą f w węzłach x_j dla $0 \leq j \leq n$. Wtedy

$$\int_a^b (f^{(r)}(x))^2 dx \geq \int_a^b (s_f^{(r)}(x))^2 dx. \quad (8.1)$$

Dowód. Jeśli przedstawimy f w postaci $f = s_f + (f - s_f)$, to

$$\begin{aligned} \int_a^b (f^{(r)}(x))^2 dx &= \int_a^b (s_f^{(r)}(x))^2 dx + \int_a^b ((f - s_f)^{(r)}(x))^2 dx \\ &\quad + 2 \int_a^b s_f^{(r)}(x)(f - s_f)^{(r)}(x) dx. \end{aligned}$$

Funkcja $f - s_f$ jest w klasie $W^r(a, b)$ i zeruje się w węzłach x_j dla $0 \leq j \leq n$. Z Lematu 8.1 wynika więc, że $\int_a^b s_f^{(r)}(x)(f - s_f)^{(r)}(x) dx = 0$, a stąd (8.1). \square

Wartość całki $\int_a^b (f^{(r)}(x))^2 dx$ może być w ogólności uważana za pewną miarę gładkości funkcji. Nierówność (8.1) możemy więc interpretować w następujący sposób: naturalna funkcja sklejana jest w klasie $W^r(a, b)$ najgładszą funkcją spełniającą dane warunki interpolacyjne w węzłach x_j .

Konsekwencją Lematu 8.1 są również inne aproksymacyjne własności naturalnych funkcji sklepanych, zob. U. 8.2.

Jak już wspomnieliśmy, najczęściej używanymi są kubiczne funkcje sklepane. Dlatego poświęcimy im osobny podrozdział.

8.3 Kubiczne funkcje sklepane

Jeśli zdecydowaliśmy się na użycie kubicznych funkcji sklepanych to powstaje problem wyznaczenia $s_f \in \mathcal{S}_2$ interpolującej daną funkcję f , tzn. takiej, że $s_f(x_i) = f(x_i)$ dla $0 \leq i \leq n$. W tym celu, na każdym przedziale $[x_i, x_{i+1}]$ przedstawimy s_f w postaci jej rozwinięcia w szereg Taylora w punkcie x_i ,

$$s_f(x) = w_i(x) = a_i + b_i(x - x_i) + c_i \frac{(x - x_i)^2}{2} + d_i \frac{(x - x_i)^3}{6},$$

i podamy algorytm obliczania a_i, b_i, c_i, d_i dla $0 \leq i \leq n - 1$.

Warunki brzegowe i warunki ciągłości dla s_f'' w wewnętrznych punktach x_i przedziału $[a, b]$ dają nam $w_0''(x_0) = 0 = w_{n-1}''(x_n)$ oraz $w_i''(x_{i+1}) = w_{i+1}''(x_{i+1})$, czyli

$$\begin{aligned} c_0 &= 0, \\ c_i + d_i h_i &= c_{i+1}, \quad 0 \leq i \leq n - 2, \\ c_{n-1} + d_{n-1} h_{n-1} &= 0, \end{aligned}$$

gdzie $h_i = x_{i+1} - x_i$. Stąd, przyjmując dodatkowo $c_n = 0$, otrzymujemy

$$d_i = \frac{c_{i+1} - c_i}{h_i}, \quad 0 \leq i \leq n - 1. \quad (8.2)$$

Z warunków ciągłości dla s_f' w punktach x_i dostajemy z kolei

$$b_i + c_i h_i + \frac{1}{2} d_i h_i^2 = b_{i+1}, \quad 0 \leq i \leq n - 2,$$

Zauważmy, że macierz układu (8.9) jest trójdiagonalna i ma dominującą przekątną. Układ ten można więc rozwiązać kosztem proporcjonalnym do n używając algorytmu przegania z U. 3.6. Koszt znalezienia wszystkich współczynników naturalnej kubicznej funkcji sklejaney interpolującej f jest więc też proporcjonalny do n .

Na końcu oszacujemy jeszcze błąd interpolacji naturalnymi kubicznymi funkcjami sklejanymi na przedziale $[a, b]$. Będziemy zakładać, że f jest dwa razy różniczkowalna w sposób ciągły.

Twierdzenie 8.3 *Jeśli $f \in C_M^2([a, b])$ to*

$$\|f - s_f\|_{C([a,b])} \leq 5M \max_{1 \leq i \leq n} (x_i - x_{i-1})^2.$$

W szczególności, dla podziału równomiernego, $x_i = a + \frac{i}{n}(b-a)$ dla $0 \leq i \leq n$, mamy

$$\|f - s_f\|_{C([a,b])} \leq 5M \left(\frac{b-a}{n}\right)^2.$$

Dowód. Wykorzystamy obliczoną wcześniej postać interpolującej funkcji sklejaney s_f . Niech x będzie punktem w przedziale $[x_i, x_{i+1}]$. Wtedy

$$s_f(x) = f(x_i) + \left(\frac{f(x_{i+1}) - f(x_i)}{h_i} - h_i(c_{i+1}^* + 2c_i^*) \right) (x - x_i) + 3c_i^*(x - x_i)^2 + \frac{c_{i+1}^* - c_i^*}{h_i} (x - x_i)^3.$$

Z rozwinięcia f w szereg Taylora w punkcie x_i dostajemy $f(x) = f(x_i) + f'(x_i)(x - x_i) + \frac{1}{2}f''(\xi_1)(x - x_i)^2$ oraz $\frac{1}{h_i}(f(x_{i+1}) - f(x_i)) = f'(x_i) + \frac{1}{2}h_i f''(\xi_2)$ dla pewnych ξ_1, ξ_2 . Stąd

$$\begin{aligned} f(x) - s_f(x) &= \frac{f''(\xi_1)}{2}(x - x_i)^2 - \left(\frac{f''(\xi_2)}{2} - (c_{i+1}^* + 2c_i^*) \right) h_i (x - x_i) \\ &\quad - 3c_i^*(x - x_i)^2 - \frac{c_{i+1}^* - c_i^*}{h_i} (x - x_i)^3, \end{aligned}$$

czyli

$$|f(x) - s_f(x)| \leq (M + 2|c_{i+1}^*| + 6|c_i^*|) h_i^2 = (M + 8 \max_{1 \leq i \leq n-1} |c_i^*|) h_i^2. \quad (8.10)$$

Niech teraz $\max_{1 \leq i \leq n-1} |c_i^*| = |c_s^*|$. Przyjmując $u_1 = 0 = w_{n-1}$, z postaci układu (8.9) otrzymujemy

$$|c_s^*| \leq 2|c_s^*| - |c_s^*(u_s + w_s)| \leq |u_s c_{s-1}^* + 2c_s^* + w_s c_{s+1}^*| = |f[x_{s-1}, x_s, x_{s+1}]| \leq \left| \frac{f''(\xi_3)}{2} \right| \leq \frac{1}{2}M,$$

a stąd i z (8.10)

$$|f(x) - s_f(x)| \leq 5Mh_i^2,$$

co kończy dowód. \square

Uwagi i uzupełnienia

U. 8.1 Zamiast terminu funkcje sklelane używa się też często terminów *splajny* (ang. *spline*-sklejać), albo *funkcje gięte*.

U. 8.2 W Rozdziale 7 pokazaliśmy, że aproksymacja kawałkami wielomianowa stopnia r jest optymalna (co do rzędu zbieżności) w klasie $C_M^{r+1}([a, b])$, wśród wszystkich aproksymacji korzystających jedynie z informacji o wartościach funkcji w danej liczbie n punktów. Okazuje się, że podobne optymalne własności wykazują funkcje sklelane, ale w innych klasach funkcji.

Niech

$$W_M^r(a, b) = \left\{ f \in W^r(a, b) : \int_a^b (f^{(r)}(x))^2 dx \leq M \right\}.$$

Ustalmy węzły $a = x_0 < \dots < x_n = b$. Dla $f \in W_M^r(a, b)$, niech s_f będzie naturalną funkcją sklejaną interpolującą f w x_j dla $0 \leq j \leq n$, a a_f dowolną inną aproksymacją korzystającą jedynie z informacji o wartościach f w tych węzłach, tzn.

$$a_f = \phi(f(x_0), \dots, f(x_n)).$$

Załóżmy, że błąd aproksymacji mierzymy nie w normie Czebyszewa, ale w normie średniokwadratowej, zdefiniowanej jako

$$\|g\|_{\mathcal{L}_2(a,b)} = \sqrt{\int_a^b (g(x))^2 dx}.$$

Wtedy

$$\sup_{f \in W_M^r(a,b)} \|f - s_f\|_{\mathcal{L}_2(a,b)} \leq \sup_{f \in W_M^r(a,b)} \|f - a_f\|_{\mathcal{L}_2(a,b)}.$$

Aproksymacja naturalnymi funkcjami sklejanymi jest więc optymalna w klasie $W_M^r(a, b)$.

Można również pokazać, że interpolacja s_f^* naturalnymi funkcjami sklejanymi na węzłach równoodległych, $x_j = a + \frac{j}{n}(b-a)$ dla $0 \leq j \leq n$, jest optymalna co do rzędu w klasie $W_M^r(a, b)$, wśród wszystkich aproksymacji korzystających jedynie z informacji o wartościach funkcji w $n+1$ dowolnych punktach, oraz

$$\max_{f \in W_M^r(a,b)} \|f - s_f^*\|_{\mathcal{L}_2(a,b)} \asymp n^{-r}.$$

U. 8.3 Tak jak wielomiany, naturalne funkcje sklelane interpolujące dane funkcje można reprezentować przez ich współczynniki w różnych bazach. Do tego celu można na przykład użyć bazy kanonicznej K_j , $0 \leq j \leq n$, zdefiniowanej równościami

$$K_j(x_i) = \begin{cases} 0 & i \neq j, \\ 1 & i = j, \end{cases}$$

przy której $s_f(x) = \sum_{j=0}^n f(x_j)K_j(x)$. Baza kanoniczna jest jednak niewygodna w użyciu, bo funkcje K_j w ogólności nie zerują się na żadnym podprzedziale, a tym samym manipulowanie nimi jest utrudnione.

Częściej używa się bazy B-sklejanej B_j , $0 \leq j \leq n$. W przypadku funkcji kubicznych, $r = 2$, jest ona zdefiniowana przez następujące warunki:

$$\begin{aligned} B_j(x_j) &= 1, & \text{dla } 0 \leq j \leq n, \\ B_j(x) &= 0, & \text{dla } x \leq x_{j-2}, j \geq 2, \text{ oraz dla } x \geq x_{j+2}, j \leq n-2. \end{aligned}$$

Dla B_0 i B_1 dodatkowo żądamy, aby

$$B_0''(x_0) = 0 = B_1''(x_0), \quad B_1(x_0) = 0,$$

a dla B_{n-1} i B_n podobnie

$$B_{n-1}''(x_n) = 0 = B_n''(x_n), \quad B_{n-1}(x_{n-1}) = 0.$$

Wtedy B_j nie zerują się tylko na przedziale (x_{j-2}, x_{j+2}) , Wyznaczenie współczynników rozwinięcia w bazie $\{B_i\}_{i=0}^n$ funkcji sklejaney interpolującej f wymaga rozwiązania układu liniowego z macierzą trójdgonalną $\{B_j(x_i)\}_{i,j=0}^n$, a więc koszt obliczenia tych współczynników jest proporcjonalny do n .

U. 8.4 Oprócz naturalnych funkcji sklejaney często rozpatruje się też *okresowe funkcje sklejaney*. Są to funkcje $\tilde{s} : \mathbb{R} \rightarrow \mathbb{R}$ spełniające warunki (i), (ii) z Rozdziału 8.1, oraz warunek:

(iii)' $\tilde{s}^{(i)}$ jest dla $0 \leq i \leq r-1$ funkcją okresową o okresie $(b-a)$, tzn. $\tilde{s}^{(i)}(x) = \tilde{s}^{(i)}(x+b-a)$.

Klasę okresowych funkcji sklejaney rzędu r oznaczymy przez $\tilde{\mathcal{S}}_r$. Funkcje te mają podobne własności jak naturalne funkcje sklejaney. Dokładniej, niech

$$\tilde{W}^r(a, b) = \{ f \in W^r(a, b) : f^{(i)}(a) = f^{(i)}(b), 0 \leq i \leq r-1 \},$$

tzn. $\tilde{W}^r(a, b)$ jest klasą funkcji z $W^r(a, b)$, które można przedłużyć do funkcji, których wszystkie pochodne do rzędu $r-1$ włącznie są $(b-a)$ -okresowe na \mathbb{R} . Wtedy dla dowolnej funkcji $f \in \tilde{W}^r(a, b)$ zerującej się w węzłach x_j , oraz dla dowolnej $\tilde{s} \in \tilde{\mathcal{S}}_r$ mamy

$$\int_a^b f^{(r)}(x) \tilde{s}^{(r)}(x) dx = 0.$$

Jest to odpowiednik Lematu 8.1 w przypadku okresowym. W szczególności wynika z niego jednoznaczność rozwiązania zadania interpolacyjnego dla okresowych funkcji f (tzn. takich, że $f(a) = f(b)$), jak również odpowiednia własność minimalizacyjna okresowych funkcji sklejaney. Dokładniej, jeśli $f \in \tilde{W}^r(a, b)$ oraz $\tilde{s}_f \in \tilde{\mathcal{S}}_r$ interpoluje f w węzłach x_j , $0 \leq j \leq n$, to

$$\int_a^b (f^{(r)}(x))^2 dx \geq \int_a^b (\tilde{s}_f^{(r)}(x))^2 dx. \quad (8.11)$$

U. 8.5 Rozpatrzmy przestrzeń $W^r(a, b)$. Utożsamiając funkcje $f_1, f_2 \in W^r(a, b)$ takie, że $f_1(x) - f_2(x) \in \Pi_{r-1}$, zdefiniujemy w $W^r(a, b)$ normę

$$\|f\| = \sqrt{\int_a^b (f^{(r)}(x))^2 dx}.$$

Dla ustalonych węzłów $a = x_0 < \dots < x_n = b$, niech \mathcal{S}_r będzie podprzestrzenią $W^r(a, b)$ składającą się z naturalnych funkcji sklejaney rzędu r opartych na węzłach x_j dla $0 \leq j \leq n$. Oczywiście $\dim \mathcal{S}_r = n+1$, co wynika z jednoznaczności rozwiązania w \mathcal{S}_r zadania interpolacji. Okazuje się, że wtedy optymalną dla $f \in W^r(a, b)$ funkcją w podprzestrzeni \mathcal{S} jest naturalna funkcja sklejana s_f interpolująca f w węzłach x_j , tzn.

$$\|f - s_f\| = \min_{s \in \mathcal{S}_r} \|f - s\|.$$

Rzeczywiście, ponieważ norma w przestrzeni $W^r(a, b)$ generowana jest przez iloczyn skalarny

$$\langle f_1, f_2 \rangle = \int_a^b f_1^{(r)}(x) f_2^{(r)}(x) dx,$$

jest to przestrzeń unitarna. Znane twierdzenie mówi, że w przestrzeni unitarnej najbliższą danej f funkcją w dowolnej domkniętej podprzestrzeni V jest rzut prostopadły f na V , albo równoważnie, taka funkcja $v_f \in V$, że iloczyn skalarny

$$\langle f - v_f, v \rangle = 0 \quad \text{dla wszystkich } v \in V,$$

cf. U. 7.7. W naszym przypadku, ostatnia równość jest równoważna

$$\int_a^b (f - v_f)^{(r)}(x) s^{(r)}(x) dx = 0.$$

To zaś jest na mocy Lematu 8.1 prawdą gdy v_f interpoluje f w punktach x_j , czyli $v_f = s_f$.

Ćwiczenia

Ćw. 8.1 Dla $z \in \mathbb{R}$, niech

$$z_+ = \max(0, z) = \begin{cases} z, & z \geq 0, \\ 0, & z < 0. \end{cases}$$

Pokaż, że każdą funkcję sklejaną s rzędu r można przedstawić w postaci

$$s(x) = w(x) + \sum_{j=0}^n a_j \left((x - x_j)_+ \right)^{2r-1},$$

gdzie a_j są pewnymi współczynnikami rzeczywistymi, a $w \in \Pi_{2r-1}$. Ponadto, jeśli s jest naturalna, to $w \in \Pi_{r-1}$, a współczynniki a_j spełniają równania

$$\sum_{j=0}^n a_j x_j^i = 0 \quad \text{dla} \quad 0 \leq i \leq r-1.$$

Ćw. 8.2 Niech $h > 0$ i $c \in \mathbb{R}$. Wyznacz współczynniki kubicznej funkcji skleianej s opartej na pięciu węzłach $-2h, -h, 0, h, 2h$ i spełniającej dodatkowo następujące warunki interpolacyjne:

$$s(0) = c, \quad s^{(k)}(\pm 2h) = 0 \quad \text{dla} \quad k = 0, 1, 2.$$

Ćw. 8.3 Wykorzystując Ćw. 8.2 wykaż, że dla dowolnego układu węzłów $a \leq x_1 < \dots < x_n \leq b$ mamy

$$\max \{ \|f''\|_{C([a,b])} : f \in C_M^2([a,b]), f(x_i) = 0, 1 \leq i \leq n \} \geq \frac{M}{24} \left(\frac{b-a}{n+1} \right)^2.$$

Stąd i z U. 7.13 wywnioskuj, że dla każdej aproksymacji a_f korzystającej jedynie z wartości f w n punktach,

$$\max_{f \in C_M^2([a,b])} \|f - a_f\|_{C([a,b])} \geq \frac{M}{24} \left(\frac{b-a}{n+1} \right)^2.$$

Ćw. 8.4 Pokaż, że funkcje B_j , $0 \leq j \leq n$, zdefiniowane w U. 8.3 istnieją i tworzą bazę w przestrzeni \mathcal{S}_r .

Ćw. 8.5 Pokaż jednoznaczność rozwiązania zadania interpolacyjnego w przypadku okresowych funkcji sklejaných zdefiniowanych w U. 8.4, oraz własność minimalizacyjną (8.11).

Wskazówka. Zastosuj technikę dowodową podobną do tej z przypadku naturalnych funkcji sklejaných.

Ćw. 8.6 Pokaż, że ogólnie postawione zadanie aproksymacyjne z U. 8.5 ma rozwiązanie.

Rozdział 9

Całkowanie numeryczne

Zajmiemy się teraz zadaniem całkowania numerycznego. Polega ono na obliczeniu (a raczej przybliżeniu) całki oznaczonej

$$S(f) = \int_a^b f(x) dx,$$

gdzie $-\infty < a < b < +\infty$, a f należy do pewnej klasy F funkcji rzeczywistych określonych i całkownych w sensie Riemanna na całym przedziale $[a, b]$.

Będziemy zakładać, że mamy możliwość obliczania wartości funkcji f , a w niektórych przypadkach również jej pochodnych, o ile istnieją. Dokładna całka $S(f)$ będzie więc w ogólności przybliżana wartością $A(f)$, która zależy tylko od wartości f i ew. jej pochodnych w skończonej liczbie punktów.

9.1 Co to są kwadratury?

Kwadraturami nazywamy funkcjonały liniowe $Q_n : F \rightarrow \mathbb{R}$ postaci

$$Q_n(f) = \sum_{i=0}^n a_i f(x_i),$$

albo ogólniej

$$Q_n(f) = \sum_{i=0}^k \sum_{j=0}^{n_i-1} a_{i,j} f^{(j)}(x_i), \quad (9.1)$$

gdzie x_i są punktami z $[a, b]$, współczynniki a_i i $a_{i,j}$ są rzeczywiste, oraz $n+1 = \sum_{i=0}^k n_i$. Zauważmy, że obliczenia kwadratur są dopuszczalne w naszym modelu obliczeniowym, mogą więc służyć jako sposób przybliżania całki.

Jeden z możliwych sposobów konstrukcji kwadratur jest następujący. Najpierw wybieramy węzły x_j (pojedyncze lub wielokrotne), budujemy wielomian interpolacyjny odpowiadający tym węzłom, a następnie całkujemy go. Ponieważ wielomian interpolacyjny zależy tylko od danej informacji o f , otrzymana w ten sposób wartość też będzie zależeć tylko od tej informacji, a w konsekwencji funkcjonał wynikowy będzie postaci (9.1). Są to tzw. kwadratury interpolacyjne.

Definicja 9.1 Kwadraturę Q_n^I opartą na węzłach o łącznej krotności $n+1$ nazywamy kwadraturą interpolacyjną, jeśli

$$Q_n^I(f) = \int_a^b w_{f,n}(x) dx,$$

gdzie $w_{f,n}$ jest wielomianem interpolacyjnym funkcji f stopnia co najwyżej n , opartym na tych węzłach.

Współczynniki kwadratur interpolacyjnych można łatwo wyliczyć. Rozpatrzmy dla uproszczenia przypadek, gdy węzły są jednokrotne. Zapisując wielomian interpolacyjny w postaci jego rozwinięcia w bazie kanonicznej Lagrange'a l_i (cf. (6.3)), otrzymujemy

$$Q_n^I(f) = \int_a^b \sum_{i=0}^n f(x_i) l_i(x) dx = \sum_{i=0}^n f(x_i) \int_a^b l_i(x) dx,$$

a stąd i z postaci l_i ,

$$a_i = \int_a^b \frac{(x - x_0) \cdots (x - x_{i-1})(x - x_{i+1}) \cdots (x - x_n)}{(x_i - x_0) \cdots (x_i - x_{i-1})(x_i - x_{i+1}) \cdots (x_i - x_n)} dx, \quad 0 \leq i \leq n.$$

Podamy teraz kilka przykładów.

Kwadratura prostokątów jest oparta na jednym węźle $x_0 = \frac{1}{2}(a + b)$,

$$R(f) = (b - a)f\left(\frac{a + b}{2}\right).$$

Kwadratura trapezów jest oparta na jednokrotnych węzłach $x_0 = a$, $x_1 = b$ i jest równa polu odpowiedniego trapezu,

$$T(f) = \frac{b - a}{2}(f(a) + f(b)).$$

Kwadratura parabol (Simpsona) jest oparta na jednokrotnych węzłach $x_0 = a$, $x_1 = \frac{1}{2}(a + b)$, $x_2 = b$, i jest równa polu pod parabolą interpolującą f w tych węzłach,

$$P(f) = \frac{b - a}{6} \left(f(a) + 4f\left(\frac{a + b}{2}\right) + f(b) \right).$$

Zauważmy, że kwadratury trapezów i parabol są oparte na węzłach jednokrotnych i równoodległych, przy czym $x_0 = a$ i $x_n = b$. Ogólnie, kwadratury interpolacyjne oparte na węzłach równoodległych, $x_i = a + \frac{i}{n}(b - a)$ dla $0 \leq i \leq n$, nazywamy *kwadraturami Newtona–Cotesa*.

9.2 Błąd kwadratur interpolacyjnych

Zajmiemy się teraz błędem kwadratur interpolacyjnych. Przypomnijmy, że $C_M^r([a, b])$ oznacza klasę funkcji r razy różniczkowalnych w sposób ciągły i takich, że $|f^{(r)}(x)| \leq M$ dla $x \in [a, b]$.

Twierdzenie 9.1 *Niech Q_r^I będzie kwadraturą interpolacyjną opartą na (jednokrotnych lub wielokrotnych) węzłach x_i dla $0 \leq i \leq r$. Jeśli $f \in C_M^{r+1}([a, b])$ to*

$$|S(f) - Q_r^I(f)| \leq \frac{M}{(r + 1)!} \int_a^b |(x - x_0)(x - x_1) \cdots (x - x_r)| dx.$$

W szczególności,

$$|S(f) - Q_r^I(f)| \leq \frac{M}{(r + 1)!} (b - a)^{r+2}. \quad (9.2)$$

Dowód. Teza wynika bezpośrednio ze znanego nam wzoru na błąd interpolacji wielomianowej z Lematu 7.1, który implikuje, że

$$S(f) - Q_r^I(f) = \int_a^b (x - x_0)(x - x_1) \cdots (x - x_n) f[x_0, x_1, \dots, x_r, x] dx,$$

oraz z faktu, że występujące tu różnice dzielone są ograniczone z góry przez $\frac{M}{(r+1)!}$. \square

Twierdzenie 9.1 rodzi pytanie o minimalną wartość całki

$$\int_a^b |(x - u_0)(x - u_1) \cdots (x - u_r)| dx, \quad (9.3)$$

ze względu na wybór węzłów u_j . Odpowiedź daje następujące twierdzenie.

Twierdzenie 9.2 *Najmniejsza wartość całki (9.3) wynosi $4\left(\frac{b-a}{4}\right)^{r+2}$ i jest przyjmowana dla węzłów*

$$u_j^* = \frac{a+b}{2} - \frac{b-a}{2} \cos\left(\frac{j+1}{r+2}\pi\right), \quad 0 \leq j \leq r.$$

Stąd, jeśli $f \in F_M^r([a, b])$ to dla kwadratury interpolacyjnej Q_r^* opartej na węzłach u_j^* mamy

$$|S(f) - Q_r^*(f)| \leq \frac{4M}{(r+1)!} \left(\frac{b-a}{4}\right)^{r+2}.$$

Aby udowodnić to twierdzenie, potrzebujemy kilka dodatkowych faktów.

Ciąg $\{U_k\}_{k \geq 0}$ wielomianów Czebyszewa drugiego rodzaju zdefiniowany jest jako (cf. (7.5))

$$\begin{aligned} U_0(x) &= 1, & U_1(x) &= 2x, \\ U_k(x) &= 2xU_{k-1}(x) - U_{k-2}(x), & \text{dla } k &\geq 2. \end{aligned}$$

Zauważmy, że U_k jest wielomianem stopnia dokładnie k o współczynniku przy x^k równym 2^k . Ponadto, wielomian U_k można dla $|x| \leq 1$ przedstawić w postaci

$$U_k(x) = \frac{\sin((k+1)\arccos x)}{\sin(\arccos x)} = \frac{\sin(k+1)t}{\sin t}, \quad \text{gdzie } x = \cos t. \quad (9.4)$$

Łatwo bowiem sprawdzić, stosując wzór na sumę sinusów, że tak zdefiniowane funkcje spełniają definicję U_k . Ze wzoru (9.4) wynika również, że $\|U_k\|_{C([-1,1])} = U_k(1) = k+1$ oraz, że wielomian U_k ma dokładnie k pojedynczych zer w $[-1, 1]$ równych

$$u_j = -\cos\left(\frac{j}{k+1}\pi\right), \quad 1 \leq j \leq k.$$

Przypomnijmy klasę $\bar{\Pi}_k$ wielomianów stopnia k o współczynniku wiodącym równym 1, tzn.

$$\bar{\Pi}_k = \{w \in \Pi_k : w(x) = x^k + \dots\}.$$

Lemat 9.1 *Niech $k \geq 1$. W klasie $\bar{\Pi}_k$ minimalną normę pierwszą na przedziale $(-1, 1)$ ma wielomian $v^* = 2^{-k}U_k$, tzn.*

$$\min_{v \in \bar{\Pi}_k} \|v\|_{L^1(-1,1)} = \|v^*\|_{L^1(-1,1)} = \frac{1}{2^{k-1}}.$$

Dowód. Teza wynika bezpośrednio z faktu, że

$$I_j := \int_{-1}^1 U_j(x) \operatorname{sgn}(U_k(x)) dx = \begin{cases} 0 & \text{dla } 0 \leq j < k, \\ 2 & \text{dla } j = k. \end{cases} \quad (9.5)$$

Rzeczywiście, wtedy dla dowolnego wielomianu $w = \frac{1}{2^k}U_k + \sum_{i=0}^{k-1} a_i U_i \in \bar{\Pi}_k$ mamy

$$\begin{aligned} \|w\|_{L^1(-1,1)} &= \int_{-1}^1 |w(x)| dx \geq \int_{-1}^1 w(x) \operatorname{sgn}(U_k(x)) dx \\ &= \frac{1}{2^k} \int_{-1}^1 U_k(x) \operatorname{sgn}(U_k(x)) dx = \frac{1}{2^k} \int_{-1}^1 |U_k(x)| dx = \frac{1}{2^{k-1}}. \end{aligned}$$

Aby pokazać (9.5), dokonujemy zamiany zmiennych $x = \cos t$ i dostajemy

$$\begin{aligned} I_j &= \int_0^\pi \frac{\sin(j+1)t}{\sin t} \operatorname{sgn}\left(\frac{\sin(k+1)t}{\sin t}\right) \sin t dt \\ &= \sum_{s=0}^k (-1)^s \int_{\frac{s\pi}{k+1}}^{\frac{(s+1)\pi}{k+1}} \sin(j+1)t dt \\ &= \frac{1}{j+1} \sum_{s=0}^k (-1)^{s+1} \left(\cos\left(\frac{(j+1)(s+1)\pi}{k+1}\right) - \cos\left(\frac{(j+1)s\pi}{k+1}\right) \right). \end{aligned}$$

Dalej, oznaczając $\phi = \frac{j+1}{k+1}\pi$ i stosując wzór na różnicę cosinusów mamy

$$(j+1) I_j = \sum_{s=0}^k (-1)^{s+1} (\cos(s+1)\phi - \cos s\phi) = 2 \sin \phi \sum_{s=0}^k (-1)^s \sin\left(s + \frac{1}{2}\right)\phi.$$

Jeśli teraz $j = k$ to $\phi = \pi$ i po prawej stronie dostajemy $2 \sum_{s=0}^k (-1)^{2s} = 2(k+1)$, czyli $I_k = 2$. Jeśli zaś $j < k$ to wykorzystując wzór $\sin x = \frac{1}{2i}(e^{ix} - e^{-ix})$ (gdzie $i = \sqrt{-1}$) mamy

$$\begin{aligned} \sum_{s=0}^k (-1)^s \sin\left(s + \frac{1}{2}\right)\phi &= \frac{1}{2i} \left(e^{\frac{1}{2}i\phi} \sum_{s=0}^k (-e^{i\phi})^s - e^{-\frac{1}{2}i\phi} \sum_{s=0}^k (-e^{-i\phi})^s \right) \\ &= \frac{1}{2i} \left(e^{\frac{1}{2}i\phi} \frac{1 - (-e^{i\phi})^{k+1}}{1 + e^{i\phi}} - e^{-\frac{1}{2}i\phi} \frac{1 - (-e^{-i\phi})^{k+1}}{1 + e^{-i\phi}} \right) \\ &= \frac{1}{2i} \left(1 - (-1)^{k+1} e^{i(j+1)\pi} \right) \left(\frac{e^{\frac{1}{2}i\phi}}{1 + e^{i\phi}} - \frac{e^{-\frac{1}{2}i\phi}}{1 + e^{-i\phi}} \right) = 0, \end{aligned}$$

ponieważ co najmniej wyrażenie w ostatnich nawiasach się zeruje. \square

Dowód Twierdzenia 9.2 wynika teraz bezpośrednio z Lematu 9.1. Dla $[a, b] = [-1, 1]$ optymalnymi węzłami są zera u_j wielomianu U_{r+1} , przy których $(x - u_0)(x - u_1) \cdots (x - u_r) = \frac{1}{2^{r+1}}U_{r+1}(x)$. Jeśli przedział $[a, b]$ jest inny niż $[-1, 1]$, wystarczy dokonać liniowej zamiany zmiennych, cf. dowód Twierdzenia 7.2. \square

W szczególnych przypadkach, takich jak kwadratury trapezów T czy parabol P , możemy otrzymać innego rodzaju formuły na błąd.

Twierdzenie 9.3 (i) Jeśli $f \in C^2([a, b])$ to dla kwadratury trapezów mamy

$$S(f) - T(f) = -\frac{(b-a)^3}{12} f^{(2)}(\xi_1).$$

(ii) Jeśli $f \in C^4([a, b])$ to dla kwadratury parabol mamy

$$S(f) - P(f) = -\frac{(b-a)^5}{2880} f^{(4)}(\xi_2).$$

$(\xi_1, \xi_2 \in [a, b])$.

Dowód. (i) Ze wzoru (9.2)

$$S(f) - T(f) = \int_a^b (x-a)(x-b)f[a, b, x] dx.$$

Ponieważ funkcja $x \mapsto f(a, b, x)$ jest ciągła, a wielomian $(x-a)(x-b)$ przyjmuje jedynie wartości nieujemne, można zastosować twierdzenie o wartości średniej dla całki, aby otrzymać

$$S(f) - T(f) = f[a, b, c] \int_a^b (x-a)(x-b) dx = -\frac{f^{(2)}(\xi_1)}{2!} \frac{(b-a)^3}{6},$$

dla pewnych $c, \xi_1 \in [a, b]$.

(ii) Niech $w_{f,2} \in \Pi_2$ i $w_{f,3} \in \Pi_3$ będą wielomianami interpolacyjnymi funkcji f odpowiednio dla węzłów $a, b, \frac{1}{2}(a+b)$ oraz $a, b, \frac{1}{2}(a+b), \frac{1}{2}(a+b)$. Wtedy

$$w_{f,3}(x) = w_{f,2}(x) + f\left[a, b, \frac{a+b}{2}, \frac{a+b}{2}\right] (x-a)\left(x - \frac{a+b}{2}\right)(x-b).$$

Wobec $\int_a^b (x-a)\left(x - \frac{1}{2}(a+b)\right)(x-b) dx = 0$ mamy

$$P(f) = \int_a^b w_{f,2}(x) dx = \int_a^b w_{f,3}(x) dx.$$

Stąd i ze wzoru na błąd interpolacji Hermite'a otrzymujemy

$$\begin{aligned} S(f) - P(f) &= \int_a^b (f - w_{f,3})(x) dx \\ &= \int_a^b (x-a)\left(x - \frac{a+b}{2}\right)^2 (x-b) f\left[a, b, \frac{a+b}{2}, \frac{a+b}{2}, x\right] dx. \end{aligned}$$

Ponieważ wielomian $(x-a)\left(x - \frac{1}{2}(a+b)\right)^2(x-b)$ jest niedodatni na $[a, b]$, możemy znów zastosować twierdzenie o wartości średniej. Mamy

$$\begin{aligned} S(f) - P(f) &= f\left[a, b, \frac{a+b}{2}, \frac{a+b}{2}, c\right] \int_a^b (x-a)\left(x - \frac{a+b}{2}\right)^2 (x-b) dx \\ &= -\frac{f^{(4)}(\xi_2)}{4!} \frac{(b-a)^5}{120}, \end{aligned}$$

co kończy dowód. \square

9.3 Kwadratury złożone

Podobnie jak w przypadku zadania interpolacji chcielibyśmy, aby błąd kwadratur malał do zera, gdy liczba węzłów rośnie do nieskończoności. Można to osiągnąć stosując np. *kwadratury złożone*. Są to kwadratury, które powstają przez scałkowanie funkcji kawałkami wielomianowej interpolującej f .

Prostym przykładem kwadratury złożonej jest suma Riemanna,

$$\bar{Q}(f) = \sum_{i=0}^n (t_{i+1} - t_i) f(x_i),$$

gdzie $a = t_0 < t_1 < \dots < t_{n+1} = b$ oraz $x_i \in [t_i, t_{i+1}]$. Jeśli średnica podziału, $\max_{0 \leq i \leq n} (t_i - t_{i-1})$, maleje do zera to $\lim_{n \rightarrow \infty} \bar{Q}(f) = S(f)$.

Będziemy rozpatrywać kwadratury złożone postaci

$$\bar{Q}_{r,k}(f) = \int_a^b \bar{w}_{f,r,k}(x) dx,$$

gdzie $\bar{w}_{f,r,k}$ jest kawałkami wielomianem z Rozdziału 7.4. Dla danego k kładziemy $t_i = a + \frac{i}{k}(b-a)$, gdzie $0 \leq i \leq k$, a następnie dla każdego i wybieramy dowolne węzły $x_{i,j} \in [t_{i-1}, t_i]$, $0 \leq j \leq r$. Wtedy \bar{w}_f jest na każdym przedziale wielomianem interpolacyjnym funkcji f stopnia co najwyżej r opartym na węzłach $x_{i,j}$. Kwadratura $\bar{Q}_{r,k}$ korzysta z węzłów o łącznej krotności $n \leq k(r+1)$.

Twierdzenie 9.4 *Błąd kwadratury złożonej $\bar{Q}_{r,k}(f)$ dla funkcji $f \in C_M^{r+1}([a, b])$ jest ograniczony przez*

$$|S(f) - \bar{Q}_{r,k}(f)| \leq \frac{M}{(r+1)!} \frac{(b-a)^{r+2}}{k^{r+1}} \leq C \left(\frac{1}{n}\right)^{r+1}, \quad C = \frac{M(r+1)^{r+1}(b-a)^{r+2}}{(r+1)!}.$$

Dowód. Teza jest bezpośrednim wnioskiem z Twierdzenia 9.1. Mamy bowiem

$$\begin{aligned} |S(f) - \bar{Q}_{r,k}(f)| &\leq \sum_{i=1}^k \int_{t_{i-1}}^{t_i} |f(x) - \bar{w}_f(x)| dx \\ &\leq \sum_{i=1}^k \left(\frac{b-a}{k}\right)^{r+2} \frac{M}{(r+1)!} = \frac{(b-a)^{r+2}}{k^{r+1}} \frac{M}{(r+1)!}. \quad \square \end{aligned}$$

W klasie $C_M^{r+1}([a, b])$, błąd kwadratur złożonych jest rzędu $n^{-(r+1)}$, czyli jest taki sam jak błąd interpolacji kawałkami wielomianowej. I tak jak przy interpolacji można pokazać, że błąd każdej innej metody całkowania korzystającej jedynie z wartości funkcji lub jej pochodnych w n punktach nie może w klasie $C_M^{r+1}([a, b])$ maleć szybciej niż $n^{-(r+1)}$, zob. U. 9.4. Podane kwadratury złożone mają więc optymalny rząd zbieżności.

Zajmiemy się teraz błędem szczególnych kwadratur złożonych, mianowicie złożonych kwadratur trapezów \bar{T}_k i parabol \bar{P}_k . Powstają one przez zastosowanie na każdym przedziale $[t_{i-1}, t_i]$ odpowiednio kwadratur trapezów T i parabol P . Jak łatwo się przekonać,

$$\bar{T}_k(f) = \frac{b-a}{k} \left(\frac{f(a) + f(b)}{2} + \sum_{j=1}^{k-1} f\left(a + \frac{j}{k}(b-a)\right) \right),$$

oraz

$$\bar{P}_k(f) = \frac{b-a}{3k} \left(\frac{f(a) + f(b)}{2} + \sum_{j=1}^{k-1} f\left(a + \frac{j}{k}(b-a)\right) + 2 \sum_{j=1}^k f\left(a + \frac{2j-1}{2k}(b-a)\right) \right).$$

Twierdzenie 9.5 (i) Jeśli $f \in C^2([a, b])$ to

$$S(f) - \bar{T}_k(f) = -\frac{(b-a)^3}{12k^2} f^{(2)}(\xi_1).$$

(ii) Jeśli $f \in C^4([a, b])$ to

$$S(f) - \bar{P}_k(f) = -\frac{(b-a)^5}{2880k^4} f^{(4)}(\xi_2).$$

Dowód. Dla kwadratury trapezów mamy

$$S(f) - \bar{T}_k(f) = -\sum_{i=1}^k \frac{(b-a)^3}{12k^3} f^{(2)}(\alpha_i) = -\frac{(b-a)^3}{12k^2} f^{(2)}(\xi_1),$$

a dla kwadratury parabol podobnie

$$S(f) - \bar{P}_k(f) = -\sum_{i=1}^k \frac{(b-a)^5}{2880k^5} f^{(4)}(\beta_i) = -\frac{(b-a)^5}{2880k^4} f^{(4)}(\xi_2). \quad \square$$

Kwadratura parabol ma więc optymalny rząd zbieżności nie tylko w $C_M^3([a, b])$, ale też w $C_M^4([a, b])$.

9.4 Przyspieszanie zbieżności kwadratur

W praktyce często stosuje się obliczanie kwadratur poprzez zagęszczanie podziału przedziału $[a, b]$. Na przykład, dla złożonej kwadratury trapezów zachodzi następujący wygodny wzór rekurencyjny:

$$\bar{T}_{2k}(f) = \frac{1}{2} \left(\bar{T}_k(f) + \frac{b-a}{k} \sum_{i=1}^k f\left(a + \frac{2i-1}{2k}(b-a)\right) \right). \quad (9.6)$$

Pozwala on obliczyć $\bar{T}_{2k}(f)$ na podstawie $\bar{T}_k(f)$ poprzez “doliczenie” wartości funkcji w punktach “gęstszej” siatki. W ten sposób możemy obserwować zachowanie się kolejnych przybliżeń $\bar{T}_{2^s}(f)$ całki $S(f)$ dla $s \geq 0$. Jest to szczególnie istotne wtedy, gdy nie mamy informacji o $\|f''\|_{C([a,b])}$, a przez to nie potrafimy oszacować liczby węzłów n , dla której osiągniemy pożądaną dokładność, cf. U. 9.2.

Jeśli funkcja jest więcej niż dwa razy różniczkowalna to użycie złożonych kwadratur trapezów zdaje się tracić sens. Wtedy istnieją przecież kwadratury, których błąd maleje do zera szybciej niż n^{-2} . Okazuje się jednak, że kwadratury \bar{T}_k mogą być podstawą dla prostej rekurencyjnej konstrukcji innych kwadratur posiadających już optymalną zbieżność. Konstrukcja ta bazuje na następującym ważnym lemacie.

Lemat 9.2 (Formuła Eulera-Maclaurina)

Dla funkcji $f \in C^{2m+2}([a, b])$, błąd złożonej kwadratury trapezów \bar{T}_k wyraża się wzorem

$$S(f) - \bar{T}_k(f) = \sum_{i=1}^m c_i h^{2i} \left(f^{(2i-1)}(b) - f^{(2i-1)}(a) \right) + c_{m+1} h^{2m+2} (b-a) f^{(2m+2)}(\xi_{m,k}),$$

gdzie $h = \frac{1}{k}(b-a)$, $\xi_{m,k} \in [a, b]$, a c_i są pewnymi stałymi liczbowymi.

Mamy $c_1 = -\frac{1}{12}$, $c_2 = \frac{1}{720}$ i ogólnie

$$c_i = \frac{B_{2i}(0)}{(2i)!} \quad \text{dla } i \geq 1,$$

gdzie B_i są wielomianami Bernoulliego (zob. U. 9.5). \square

Dowód, który tutaj pominiemy, oparty jest na pewnych własnościach wielomianów Bernoulliego. Formułę Eulera-Maclaurina można przepisać w postaci

$$S(f) - \bar{T}_k(f) = \sum_{i=1}^m c_i^{(0)}(f) k^{-2i} + c_{m+1,k}^{(0)}(f) k^{-(2m+2)},$$

gdzie $c_i^{(0)}(f) = c_i(b-a)^{2i}(f^{(2i-1)}(b) - f^{(2i-1)}(a))$, $1 \leq i \leq m$, oraz $c_{m+1,k}^{(0)}(f) = c_{m+1}(b-a)^{2m+2} f^{(2m+2)}(\xi_{m+1,k})$. Zauważmy przy tym, że jeśli $f \in F_M^{2m+1}([a, b])$ to współczynniki $c_{m+1,k}^{(0)}(f)$ są wspólnie ograniczone przez $c_{m+1}(b-a)^{2m+2} M$.

Definiując teraz kwadraturę

$$\bar{T}_k^1(f) = \frac{4\bar{T}_{2k}(f) - \bar{T}_k(f)}{3},$$

dla $f \in C^4([a, b])$ mamy

$$\begin{aligned} S(f) - \bar{T}_k^1(f) &= \frac{4(S(f) - \bar{T}_{2k}(f)) - (S(f) - \bar{T}_k(f))}{3} \\ &= \frac{4}{3} \left(\frac{c_1^{(0)}(f)}{4k^2} + \frac{c_{2,2k}^{(0)}(f)}{4^2 k^4} \right) - \frac{1}{3} \left(\frac{c_1^{(0)}(f)}{k^2} + \frac{c_{2,k}^{(0)}(f)}{k^4} \right) = \frac{c_{2,k}^{(1)}(f)}{k^4}, \end{aligned}$$

gdzie $c_{2,k}^{(1)}(f) = \frac{1}{12} c_{2,2k}^{(0)}(f) - \frac{1}{3} c_{2,k}^{(0)}(f)$ i jest wspólnie ograniczone dla $f \in F_M^3([a, b])$. Kwadratura T_k^1 ma więc optymalny w $F_M^3([a, b])$ rząd zbieżności k^{-4} . Proces ten można kontynuować dalej tworząc kolejne kwadratury o coraz to wyższym rzędzie zbieżności. Dokładniej, położmy $\bar{T}_k^0(f) = \bar{T}_k(f)$ oraz

$$\bar{T}_k^s(f) = \frac{4^s \bar{T}_{2k}^{s-1}(f) - \bar{T}_k^{s-1}(f)}{4^s - 1} \quad \text{dla } s \geq 1. \quad (9.7)$$

Wtedy, dla $f \in C_M^{2m+2}([a, b])$, rząd zbieżności kwadratury \bar{T}_k^m wynosi $k^{-(2m+2)}$. Rzeczywiście, sprawdziliśmy, że jest to prawdą dla $m = 0, 1$. Niech $m \geq 2$. Postępując indukcyjnie ze względu na $s = 1, 2, \dots, m$ mamy

$$\begin{aligned} S(f) - \bar{T}_k^s(f) &= \frac{4^s(S(f) - \bar{T}_{2k}^{s-1}(f)) - (S(f) - \bar{T}_k^{s-1}(f))}{4^s - 1} \\ &= \left(4^s \left(\sum_{i=s}^m c_i^{(s-1)}(f) (2k)^{-2i} + c_{m+1,2k}^{(s-1)}(f) (2k)^{-(2m+2)} \right) \right. \\ &\quad \left. - \left(\sum_{i=s}^m c_i^{(s-1)}(f) k^{-2i} + c_{m+1,k}^{(s-1)}(f) k^{-(2m+2)} \right) \right) \frac{1}{4^s - 1} \\ &= \sum_{i=s+1}^m c_i^{(s)}(f) k^{-2i} + c_{m+1,k}^{(s)}(f) k^{-(2m+2)}, \end{aligned}$$

ponieważ współczynniki przy k^{-2s} redukują się. $c_i^{(s)}(f)$ są tutaj pewnymi nowymi stałymi, a $c_{m+1,k}^{(s)}(f)$ jest w klasie $C_M^{2m+2}([a, b])$ ograniczona przez stałą niezależną od f . Ostatecznie, dla $s = m$ mamy więc

$$S(f) - \bar{T}_k^m(f) = c_{m+1,k}^{(m)}(f) k^{-(2m+2)}$$

i w klasie $C_M^{2m+2}([a, b])$

$$|S(f) - \bar{T}_k^m(f)| \leq c_m k^{-(2m+2)}$$

dla pewnej c_m niezależnej od f (ale zależnej od M).

Zauważmy jeszcze, że \bar{T}_k^m wykorzystuje $n = k2^m + 1$ wartości funkcji f w punktach równoodległych na $[a, b]$ co oznacza, że w terminach n rząd zbieżności wynosi też $n^{-(2m+2)}$, a więc jest optymalny w klasie $C_M^{2m+2}([a, b])$.

Kwadratury \bar{T}_k^s nazywane są *kwadraturami Romberga*. Dla danej funkcji f , można je łatwo konstruować budując następującą tablicę trójkątną:

$$\begin{array}{ccccccc}
 \bar{T}_1^0(f) & & & & & & \\
 \bar{T}_2^0(f) & \bar{T}_1^1(f) & & & & & \\
 \bar{T}_4^0(f) & \bar{T}_2^1(f) & \bar{T}_1^2(f) & & & & \\
 \bar{T}_8^0(f) & \bar{T}_4^1(f) & \bar{T}_2^2(f) & \bar{T}_1^3(f) & & & \\
 \vdots & \vdots & \vdots & \vdots & \ddots & & \\
 \bar{T}_{2^s}^0(f) & \bar{T}_{2^{s-1}}^1(f) & \bar{T}_{2^{s-2}}^2(f) & \bar{T}_{2^{s-3}}^3(f) & \cdots & \bar{T}_1^s(f), &
 \end{array} \tag{9.8}$$

w której pierwsza kolumna jest tworzona indukcyjnie zgodnie ze wzorem (9.6), a kolejne zgodnie ze wzorem (9.7).

Uwagi i uzupełnienia

U. 9.1 Twierdzenie 9.1 podaje jedynie górne ograniczenie na błąd najgorszy kwadratury w klasie $C_M^{r+1}([a, b])$. Dokładne wyrażenia na ten błąd dla dowolnej kwadratury, podobne do tych z Twierdzenia 9.3, wymagają użycia tzw. jąder Peano i zostaną pokazane w uwadze U. 10.2 do Rozdziału 10.

U. 9.2 Jeśli $f \in C^2([a, b])$ i $f'(a) \neq f'(b)$ to dla dużych k mamy

$$S(f) - \bar{T}_k(f) \approx c(f)k^{-2},$$

gdzie $c(f) = -\frac{1}{12}(b-a)^2(f'(b) - f'(a))$. Stąd

$$\begin{aligned}
 \bar{T}_k(f) - \bar{T}_{k/2}(f) &= (S(f) - \bar{T}_{k/2}(f)) - (S(f) - \bar{T}_k(f)) \\
 &\approx \frac{c(f)}{k^2} - \frac{c(f)}{(k/2)^2} = -\frac{3c(f)}{k^2} \approx -3(S(f) - \bar{T}_k(f)),
 \end{aligned}$$

a więc

$$S(f) - \bar{T}_k(f) \approx \frac{1}{3}(\bar{T}_{k/2}(f) - \bar{T}_k(f)),$$

gdy $k \rightarrow +\infty$. Ostatnia przybliżona równość może sugerować kryterium kończenia obliczeń, gdy chcemy otrzymać przybliżenie całki $S(f)$ z zadaną dokładnością $\varepsilon > 0$. Mianowicie, obliczając kolejne kwadratury $\bar{T}_{2^s}(f)$ dla $s = 0, 1, \dots$, sprawdzamy jednocześnie, czy

$$|\bar{T}_{2^{s-1}}(f) - \bar{T}_{2^s}(f)| \leq 3\varepsilon.$$

Jeśli ten warunek jest spełniony dla kilku kolejnych s to z dużym prawdopodobieństwem możemy stwierdzić, że $|S(f) - \bar{T}_{2^s}(f)| \leq \varepsilon$. Oczywiście, podobne kryteria kończenia obliczeń mogą być konstruowane dla innych ciągów kwadratur, jeśli tylko znamy teoretyczne zachowanie się błędów.

U. 9.3 W rozpatrywanych w tym rozdziale kwadraturach złożonych zakładaliśmy równomierny podział przedziału całkowania na k podprzedziałów. Kwadratury *adaptacyjne* produkują podział nierównomierny. (Słowo ‘adaptacyjne’ odnosi się tu do faktu, że kolejne punkty, w których obliczana jest wartość funkcji wybierane

są w zależności od poprzednio otrzymanych wartości.) Ponieważ błąd w podprzedziale zależy od odpowiedniej pochodnej funkcji w tym podprzedziale, pomysł polega na tym aby stosować gęsty podział tam, gdzie ta pochodna jest duża i rzadki podział tam, gdzie jest ona mała. W ten sposób można istotnie zmniejszyć stałą asymptotyczną (ale nie rząd zbieżności, cf. U. 9.4) w wyrażeniu na błąd. Dodatkową zaletą kwadratur adaptacyjnych jest to, że umożliwiają automatyczne całkowanie funkcji z zadany błądem.

Zobaczmy jak podstawowa idea kwadratur adaptacyjnych może być zrealizowana na przykładzie *adaptacyjnej kwadratury Simpsona* Q^{ada} . Dla $a \leq u < v \leq b$, niech $P_1(f; u, v)$ i $P_2(f; u, v)$ oznaczają odpowiednio zwykłą kwadraturę parabol dla funkcji f opartą na punktach $u, v, \frac{1}{2}(u+v)$ i kwadraturę złożoną parabol odpowiadającą podziałowi przedziału $[u, v]$ na dwie równe części. Kwadratura Q^{ada} bierze jako parametr $\varepsilon > 0$ i dokonuje rekurencyjnego podziału przedziału całkowania na podprzedziały $[x_{i-1}, x_i]$, gdzie $a = x_0 < x_1 < \dots < x_m = b$, aż do spełnienia warunku $|P_1(f; x_{i-1}, x_i) - P_2(f; x_{i-1}, x_i)| \leq 15\varepsilon \frac{x_i - x_{i-1}}{b-a}$ dla wszystkich i . Proces ten można zapisać następująco.

```

function QUADA( $f, a, b, \varepsilon$ );
begin
   $P1 := \text{Simpson}(f, a, b)$ ;
   $P2 := \text{Simpson}(f, a, \frac{a+b}{2}) + \text{Simpson}(f, \frac{a+b}{2}, b)$ ;
  if  $|P1 - P2| \leq 15\varepsilon$  then return  $P2$  else
    return QUADA( $f, a, \frac{a+b}{2}, \frac{\varepsilon}{2}$ ) + QUADA( $f, \frac{a+b}{2}, b, \frac{\varepsilon}{2}$ )
end.

```

Okazuje się, że jeśli $f^{(4)}$ jest w całym przedziale $[a, b]$ stale nieujemna albo stale niedodatnia to QUADA zwraca wartość, która asymptotycznie, gdy $\varepsilon \rightarrow 0^+$, jest ε -przybliżeniem całki. Uzasadnienie tego faktu zasadniczo opiera się na następującym rozumowaniu. Oznaczmy $I(f; u, v) = \int_u^v f(x) dx$. Niech $f \in C^4([a, b])$ i $\xi \in [a, b]$ będzie takie, że $f^{(4)}(\xi) \neq 0$. Załóżmy, że $u \leq \xi \leq v$ oraz $|v-u|$ zbiega do zera. Wtedy, wykorzystując formuły na błąd kwadratur parabol z Twierdzeń 9.3 i 9.5 mamy

$$\begin{aligned}
 P_2(f; u, v) - P_1(f; u, v) &= (I(f; u, v) - P_1(f; u, v)) - (I(f; u, v) - P_2(f; u, v)) \\
 &= \left(-\frac{(v-u)^5}{2880} f^{(4)}(\xi_1) \right) - \left(-\frac{(v-u)^5}{2^4 \cdot 2880} f^{(4)}(\xi_2) \right) \approx -\frac{(v-u)^5}{2^4 \cdot 2880} 15 f^{(4)}(\xi) \\
 &\approx 15(I(f; u, v) - P_2(f; u, v)),
 \end{aligned}$$

a stąd

$$\begin{aligned}
 |I(f; a, b) - Q^{\text{ada}}(f; \varepsilon)| &\leq \sum_{i=1}^m |I(f; x_{i-1}, x_i) - P_2(f; x_{i-1}, x_i)| \\
 &\approx \frac{1}{15} \sum_{i=1}^m |P_1(f; x_{i-1}, x_i) - P_2(f; x_{i-1}, x_i)| \leq \frac{1}{15} \sum_{i=1}^m 15\varepsilon \frac{x_i - x_{i-1}}{b-a} = \varepsilon.
 \end{aligned}$$

Można pokazać, że dla funkcji spełniających $0 \neq f^{(4)} \geq 0$ błąd kwadratury adaptacyjnej jest asymptotycznie proporcjonalny do $(b-a) \left(\int_a^b (f^{(4)}(x))^{1/4} dx \right)^4 m^{-4}$, gdzie m jest liczbą podprzedziałów, podczas gdy analogiczny błąd dla kwadratury niedaptacyjnej \bar{P}_m wynosi $(b-a)^4 \left(\int_a^b f^{(4)}(x) dx \right) m^{-4}$, cf. Ćw. 9.13 i 9.14.

U. 9.4 Pokażemy, że jeśli dowolny algorytm A daje przybliżoną wartość całki $S(f) = \int_a^b f(x) dx$ na podstawie jedynie wartości lub pochodnych funkcji f w n różnych punktach, tzn.

$$A(f) = \phi(f(x_1), \dots, f^{(r+1)}(x_1), \dots, f(x_n), \dots, f^{(r+1)}(x_n))$$

dla pewnych $x_i \in [a, b]$ (które mogą być w ogólności wybrane adaptacyjnie), to błąd takiego przybliżenia w klasie $C_M^{r+1}([a, b])$,

$$\sup_{f \in C_M^{r+1}([a, b])} |S(f) - A(f)| \geq C_1 M n^{-(r+1)}, \quad (9.9)$$

gdzie C_1 jest pewnym czynnikiem dodatnim niezależnym od A i n .

W tym celu, podobnie jak w U. 7.5, wybierzmy dowolną nieujemną funkcję $\psi : \mathbb{R} \rightarrow \mathbb{R}$ taką, że $\psi \in C_1^{r+1}(\mathbb{R})$ oraz $\psi(x) = 0$ dla $x \notin [0, 1]$. Oznaczmy $x_0 = a$, $x_{n+1} = b$, $h_i = x_{i+1} - x_i$, $0 \leq i \leq n$, oraz zdefiniujmy funkcje

$$\psi_i(x) = M h_i^{r+1} \psi\left(\frac{x - x_i}{h_i}\right).$$

(Jeśli x_i są wybierane adaptacyjnie to bierzemy te, które są wybierane dla funkcji zerowej.) Niech $g_1(x) = -g_2(x) = \sum_{i=0}^n \psi_i(x)$. Wtedy dla $l = 1, 2$ mamy $g_l \in C_M^{r+1}([a, b])$ oraz $g_l^{(p)}(x_j) = 0$ dla $0 \leq p \leq r + 1$, $1 \leq j \leq n$. Stąd $\phi_0 = \underbrace{\phi(0, \dots, 0)}_{n(r+2)}$ jest aproksymacją dla całek obu funkcji g_l i

$$\begin{aligned} \max\{|S(g_1) - \phi_0|, |S(g_2) - \phi_0|\} &\geq \frac{1}{2} |S(g_1) - S(g_2)| = S(g_1) = \sum_{i=0}^n \int_{x_{i-1}}^{x_i} \psi_i(x) dx \\ &= M \sum_{i=0}^n h_i^{r+1} \int_{x_{i-1}}^{x_i} \psi\left(\frac{x - x_i}{h_i}\right) dx = M \left(\int_0^1 \psi(x) dx \right) \left(\sum_{i=0}^n h_i^{r+2} \right). \end{aligned}$$

Zauważmy jeszcze, że suma $\sum_{i=0}^n h_i^{r+2}$ jest minimalna dla $h_i = \frac{1}{n+1}(b - a)$, a stąd

$$S(g_1) \geq M \frac{(b - a)^{r+2}}{(n + 1)^{r+1}} \int_0^1 \psi(x) dx.$$

Nierówność (9.9) zachodzi więc z $C_1 = (2(b - a))^{r+1} \int_0^1 \psi(x) dx$.

U. 9.5 *Wielomiany Bernoulliego* B_n , występujące w sformułowaniu Lematu 9.2, są jednoznacznie zdefiniowane zależnością

$$\sum_{k=0}^n \binom{n+1}{k} B_k(x) = (n+1)x^n, \quad n = 1, 2, \dots$$

W szczególności, mamy $B_0(x) = 1$, $B_1(x) = x - \frac{1}{2}$, $B_2(x) = x^2 - x + \frac{1}{6}$, $B_3(x) = x^3 - \frac{3}{2}x^2 + \frac{1}{2}x$.

Ćwiczenia

Ćw. 9.1 Pokaż, że w kwadraturze Newtona-Cotesa opartej na $n+1$ równoodległych węzłach $x_i = a + \frac{i}{n}(b-a)$, współczynniki a_i dane są wzorami

$$a_i = \frac{(-1)^{n-i}(b-a)}{n!i!(n-i)!} \int_0^n x(x-1)\cdots(x-(i-1))(x-(i+1))\cdots(x-n) dx.$$

Ćw. 9.2 Zauważ, że Twierdzenie 9.1 zachodzi dla szerszej niż $C_M^{r+1}([a, b])$ klasy funkcji $f \in C^r([a, b])$ takich, że $f^{(r)}$ spełnia warunek Lipschitza ze stałą M .

Ćw. 9.3 Wykaż, że optymalne węzły u_j^* w Twierdzeniu 7.2 są jednoznacznie wyznaczone.

Ćw. 9.4 Pokaż, że jeśli $f \in C^2([a, b])$ to dla kwadratury prostokątów, czyli dla $R(f) = (b-a)f(\frac{1}{2}(a+b))$ mamy

$$S(f) - R(f) = \frac{(b-a)^3}{24} f^{(2)}(\xi_0) \quad (\xi_0 \in [a, b])$$

Błąd w klasie $C_M^2([a, b])$ jest więc dla kwadratury prostokątów dwa razy mniejszy niż dla kwadratury trapezów. Wywnioskuj stąd, że złożona kwadratura prostokątów,

$$\bar{Q}_{0,k}(f) = \frac{b-a}{k} \sum_{i=1}^k f\left(\frac{2i-1}{2k}\right),$$

ma optymalny rząd zbieżności nie tylko w klasie $C_M^1([a, b])$, ale też w klasie $C_M^2([a, b])$.

Ćw. 9.5 Niech

$$Q_1^I(f) = \frac{b-a}{2} \left(f\left(\frac{2a+b}{3}\right) + f\left(\frac{a+2b}{3}\right) \right).$$

Pokaż nierówność

$$\sup_{f \in C_M^2([a,b])} |S(f) - Q_1^I(f)| \leq \frac{11}{324} M(b-a)^3.$$

i porównaj z błędem kwadratury interpolacyjnej opartej na węzłach u_0^* i u_1^* z Twierdzenia 9.2.

Ćw. 9.6 Uzasadnij, że jeśli kwadratura interpolacyjna Q_r^I oparta jest na węzłach Czebyszewa

$$x_i = \frac{a+b}{2} + \frac{b-a}{2} \cos\left(\frac{2i+1}{2r+2}\pi\right), \quad 0 \leq i \leq r,$$

to dla $f \in C_M^{r+1}([a, b])$ błąd

$$|S(f) - Q_r^I(f)| \leq \frac{8M}{(r+1)!} \left(\frac{b-a}{4}\right)^{r+2},$$

a dla odpowiadającej węzłom Czebyszewa kwadratury złożonej

$$|S(f) - \bar{Q}_{r,k}^I(f)| \leq \frac{8M}{(r+1)!} \left(\frac{b-a}{4}\right)^{r+2} \left(\frac{1}{k}\right)^{r+1}.$$

Porównaj z wynikiem z Twierdzenia 9.2.

Ćw. 9.7 Rozpatrzmy kwadraturę

$$\bar{Q}_k(f) = \bar{T}_k(f) - \frac{(b-a)^2}{12k^2} (f'(b) - f'(a)).$$

Pokaż, nie korzystając z formuły Eulera-Maclaurina, że jeśli $f \in C_M^4([a, b])$ to błąd $|S(f) - \bar{Q}_k(f)|$ zbiega do zera co najmniej tak szybko jak k^{-4} , gdy $k \rightarrow +\infty$. W szczególności, dla funkcji spełniających $f'(a) = f'(b)$ rząd zbieżności złożonej kwadratury trapezów $\bar{T}_k(f)$ wynosi k^{-4} .

Ćw. 9.8 Opracuj ekonomiczny algorytm obliczania ciągu kolejnych złożonych kwadratur parabol $\bar{P}_{2^s}(f)$ dla $s = 0, 1, 2, \dots$

Ćw. 9.9 Opracuj kryterium kończenia obliczeń analogiczne do tego z U. 9.2, dla ciągu złożonych kwadratur parabol $\bar{P}_{2^s}(f)$, gdzie $s = 0, 1, \dots$

Ćw. 9.10 Pokaż, że drugą kolumnę tabeli (9.8) kwadratur Romberga tworzą złożone kwadratury parabol, tzn.

$$\bar{P}_k(f) = \frac{4\bar{T}_{2k}(f) - \bar{T}_k(f)}{3}.$$

Ćw. 9.11 Opracuj program obliczający wartość $\bar{T}_1^s(f)$ kwadratury Romberga, korzystając ze wzorów (9.6) i budując tabelę (9.7).

Ćw. 9.12 Opracuj kwadraturę adaptacyjną odpowiadającą adaptacyjnej kwadraturze Simpsona z U. 9.3, ale bazującej na kwadraturze trapezów.

Ćw. 9.13 Wykaż, że jeśli $f \in C^4([a, b])$ i $\int_a^b f^{(4)}(x) dx \neq 0$, to dla złożonej kwadratury parabol \bar{P}_k mamy

$$\lim_{k \rightarrow \infty} (S(f) - \bar{P}_k(f)) k^4 = \frac{(b-a)^4}{2880} \left(\int_a^b f^{(4)}(x) dx \right).$$

Ćw. 9.14 Wykaż, że jeśli $f \in C^4([a, b])$ i $f^{(4)} \geq 0$ to stałe asymptotyczne dla kwadratur adaptacyjnej i nieadaptacyjnej z U. 9.3 spełniają

$$(b-a) \left(\int_a^b (f^{(4)}(x))^{1/4} dx \right)^4 \leq (b-a)^4 \left(\int_a^b (f^{(4)}(x)) dx \right),$$

oraz że stosunek prawej strony tej nierówności do lewej strony może być dowolnie duży.

Rozdział 10

Całkowanie a aproksymacja

W poprzednim rozdziale badaliśmy kwadratury ze względu na kryterium błędu, w pewnych klasach funkcji. W tym rozdziale zajmiemy się badaniem kwadratur ze względu na inne, bardziej klasyczne kryterium, którym jest rząd kwadratury. Będziemy przy tym rozpatrywać nieco ogólniejsze zadanie całkowania z wagą, czyli

$$S(f) = S_\rho(f) = \int_a^b f(x)\rho(x) dx,$$

gdzie waga ρ jest prawie wszędzie dodatnia i całkowna,

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

Dopuszczymy również możliwość nieskończonych przedziałów całkowania, a więc $-\infty \leq a < b \leq +\infty$.

10.1 Rząd kwadratury

Zacznijmy od definicji.

Definicja 10.1 *Rzędem kwadratury Q_n nazywamy taką liczbę r , że*

(i) *kwadratura Q_n daje dokładną wartość całki dla wielomianów stopnia mniejszego niż r ,*

$$Q_n(w) = S_\rho(w) \quad \text{dla } w \in \Pi_{r-1},$$

(ii) *istnieje wielomian w^* stopnia r dla którego kwadratura nie jest równa całce, tzn.*

$$Q_n(w^*) \neq S_\rho(w^*).$$

Rząd kwadratury Q_n oznaczmy przez $\text{rz}(Q_n)$. Dla przykładu, załóżmy, że $\rho = 1$ i przedział $[a, b]$ jest skończony. Wtedy rząd kwadratury prostokątów R wynosi 2, tak jak rząd kwadratury trapezów T , natomiast rząd kwadratury parabol P wynosi 4. Rzeczywiście, to wynika bezpośrednio ze wzorów na błędy tych kwadratur danych w Ćw. 9.4 i Twierdzeniu 9.3. Ogólniej, mamy następujące twierdzenie, które przytaczamy bez dowodu.

Twierdzenie 10.1 *Niech waga $\rho = 1$ i niech przedział $[a, b]$ będzie skończony. Rząd kwadratury Newtona-Cotesa Q_n^{NC} , tzn. opartej na jednokrotnych węzłach równoodległych, $x_i = a + \frac{i}{n}(b-a)$ dla $0 \leq i \leq n$, wynosi*

$$\text{rz}(Q_n^{NC}) = \begin{cases} n+1 & \text{dla } n \text{ nieparzystych,} \\ n+2 & \text{dla } n \text{ parzystych.} \end{cases} \quad \square$$

Będą nas interesować kwadratury o jak najwyższym rzędzie. Chcielibyśmy wiedzieć, jaki jest maksymalny rząd kwadratur korzystających z węzłów o ustalonej krotności i jak takie kwadratury konstruować. Nie jest to tylko akademickie pytanie, bowiem istnieją ściśle związki pomiędzy rzędami a błędami kwadratur. W U. 10.3 pokazy jest taki związek w kontekście kwadratur złożonych. Tu zacytujemy inne twierdzenie.

Twierdzenie 10.2 *Rozpatrzmy ciąg kwadratur*

$$Q_n(f) = \sum_{i=0}^n a_{n,i} f(x_{n,i}) \quad \text{dla } n = 0, 1, 2, \dots,$$

aproksymujących całkę ważoną $S_\rho(f)$ na skończonym przedziale $[a, b]$. Jeśli

(i) $\lim_{n \rightarrow \infty} \text{rz}(Q_n) = +\infty,$

(ii) *istnieje $A < +\infty$ takie, że dla wszystkich n jest $\sum_{i=0}^n |a_{n,i}| \leq A,$*

to dla każdej funkcji $f \in C([a, b])$ mamy

$$\lim_{n \rightarrow \infty} Q_n(f) = S_\rho(f).$$

Dowód. Weźmy $\epsilon > 0$. Niech w_ϵ będzie wielomianem takim, że $\|f - w_\epsilon\|_C \leq \epsilon$. Wobec (i), dla wszystkich dostatecznie dużych n , tzn. spełniających $\text{rz}(Q_n) > \deg(w_\epsilon)$, mamy $S(w_\epsilon) = Q_n(w_\epsilon)$. Stąd

$$\begin{aligned} |S(f) - Q_n(f)| &\leq |S(f) - S(w_\epsilon)| + |S(w_\epsilon) - Q_n(w_\epsilon)| + |Q_n(f) - Q_n(w_\epsilon)| \\ &= \left| \int_a^b (f - w_\epsilon)(x) \rho(x) dx \right| + \left| \sum_{i=0}^n a_{n,i} (f - w_\epsilon)(x_{n,i}) \right| \\ &\leq \epsilon \int_a^b \rho(x) dx + \epsilon \sum_{i=0}^n |a_{n,i}| \leq \epsilon \left(\int_a^b \rho(x) dx + A \right), \end{aligned}$$

co wobec dowolności ϵ dowodzi tezy twierdzenia. \square

Częściową odpowiedź na pytanie o maksymalny rząd kwadratur daje następujący lemat.

Lemat 10.1 *Niech Q_n będzie kwadraturą opartą na węzłach o łącznej krotności $n + 1$.*

(i) *Jeśli $\text{rz}(Q_n) \geq n + 1$ to Q_n jest kwadraturą interpolacyjną.*

(ii) *Jeśli Q_n jest kwadraturą interpolacyjną to*

$$n + 1 \leq \text{rz}(Q_n) \leq 2(n + 1).$$

Dowód. (i) Dla dowolnej funkcji f , niech w_f będzie wielomianem interpolacyjnym dla f stopnia co najwyżej n , opartym na tej samej informacji o f , z jakiej korzysta kwadratura Q_n . Wtedy mamy $Q_n(f) = Q_n(w_f)$, a ponieważ $\text{rz}(Q_n) \geq n + 1$ to również $Q_n(w_f) = S_\rho(w_f)$. Stąd

$$Q_n(f) = S_\rho(w_f),$$

a to oznacza, że Q_n jest interpolacyjna.

(ii) Jeśli Q_n jest interpolacyjna to dla każdego $w \in \Pi_n$ mamy $Q_n(w) = S_\rho(w)$, bo w jest dla siebie wielomianem interpolacyjnym. Stąd $\text{rz}(Q_n) \geq n + 1$.

Dla dowodu drugiej nierówności załóżmy, że Q_n korzysta z węzłów x_0, x_1, \dots, x_n (być może wielokrotnych). Wtedy dla wielomianu

$$w_1(x) = (x - x_0)^2(x - x_1)^2 \cdots (x - x_n)^2$$

mamy $w_1 \in \Pi_{2(n+1)}$ oraz $S_\rho(w_1) > 0$, bo w_1 jest prawie wszędzie dodatni. Z drugiej strony, $Q_n(w_1) = 0$, bo informacja o w_1 jest zerowa. Stąd

$$Q_n(w_1) \neq S_\rho(w_1)$$

i w konsekwencji $\text{rz}(Q_n) \leq 2(n+1)$. \square

Lemat 10.1 mówi nam, że w poszukiwaniu kwadratur o najwyższym rzędzie możemy ograniczyć się do kwadratur interpolacyjnych. Ponadto, maksymalny rząd jest na pewno nie mniejszy niż $n+1$ i na pewno nie większy niż $2(n+1)$. Aby jednak odpowiedzieć na pytanie jaki jest rzeczywisty maksymalny rząd, musimy odwołać się do pewnych faktów z teorii aproksymacji w przestrzeniach unitarnych, a ściślej, do własności ciągów wielomianów ortogonalnych.

10.2 Ciągi wielomianów ortogonalnych

Waga ρ definiuje nam przestrzeń $\mathcal{L}_{2,\rho}(a, b)$ funkcji o wartościach rzeczywistych, określonych i całkowalnych na przedziale (a, b) i takich, że $\int_a^b f^2(x)\rho(x) dx < +\infty$. W $\mathcal{L}_{2,\rho}(a, b)$ wprowadzimy iloczyn skalarny

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

i generowaną przez ten iloczyn normę

$$\|f\| = \sqrt{\langle f, f \rangle} = \sqrt{\int_a^b f^2(x)\rho(x) dx}.$$

W ten sposób, $\mathcal{L}_{2,\rho}(a, b)$ jest przestrzenią unitarną i można mówić o ortogonalności (prostopadłości) elementów w tej przestrzeni.

Definicja 10.2 Ciąg wielomianów p_k , $k = 0, 1, 2, \dots$, nazywamy ciągiem wielomianów ortogonalnych w przestrzeni $\mathcal{L}_{2,\rho}(a, b)$ jeśli

(i) wielomian p_k jest stopnia dokładnie k ,

$$\deg p_k = k \quad \text{dla} \quad k \geq 0,$$

(ii) wielomiany te są wzajemnie ortogonalne w przestrzeni $\mathcal{L}_{2,\rho}(a, b)$, tzn.

$$\langle p_i, p_j \rangle = \int_a^b p_i(x)p_j(x)\rho(x) dx = 0 \quad \text{dla} \quad i \neq j.$$

Pokażemy, że ta definicja ma sens.

Lemat 10.2 Ciąg wielomianów ortogonalnych $\{p_k\}_{k \geq 0}$ w przestrzeni unitarnej $\mathcal{L}_{2,\rho}(a, b)$ istnieje. Jeśli dodatkowo założymy, że $p_k(x) = x^k + \dots$, tzn. współczynnik przy najwyższej potędze wielomianu p_k wynosi 1, to ciąg taki jest wyznaczony jednoznacznie.

Dowód. Wielomiany p_k można skonstruować stosując proces ortogonalizacji Grama-Schmidta, tzn. ortogonalizując dowolny ciąg wielomianów $\{w_k\}_{k \geq 0}$ spełniający $\deg w_k = k$. Rzeczywiście, weźmy np. $w_k(x) = x^k$. Jeśli położymy $p_0 = w_0$ i dalej indukcyjnie

$$p_k = w_k - \sum_{j=0}^{k-1} \frac{\langle w_k, p_j \rangle}{\langle p_j, p_j \rangle} p_j, \quad k \geq 1,$$

to, jak łatwo bezpośrednio sprawdzić, $\{p_k\}_{k \geq 0}$ będzie ciągiem wielomianów ortogonalnych i takich, że $p_k(x) = x^k + \dots$

Aby pokazać jednoznaczność założymy, że istnieje inny ciąg $\{q_k\}_{k \geq 0}$ wielomianów ortogonalnych, w którym współczynniki przy najwyższych potęgach wynoszą 1. Ponieważ $\deg p_k = k$, wielomiany p_0, p_1, \dots, p_k tworzą bazę ortogonalną w Π_k (ze względu na iloczyn skalarny $\langle \cdot, \cdot \rangle$), a stąd

$$q_k = \sum_{j=0}^k \frac{\langle q_k, p_j \rangle}{\langle p_j, p_j \rangle} p_j.$$

Z drugiej strony, dla $0 \leq j \leq k-1$ mamy $\langle q_k, p_j \rangle = 0$, bo ortogonalność q_k do wielomianów bazowych p_j , $0 \leq j \leq k-1$, jest równoważna ortogonalności do całej przestrzeni Π_{k-1} , a w szczególności do p_j . Stąd

$$q_k = \frac{\langle q_k, p_k \rangle}{\langle p_k, p_k \rangle} p_k,$$

a ponieważ współczynniki przy x^k w wielomianach q_k i p_k są takie same, to $q_k = p_k$. \square

Pokazaliśmy, że ciąg wielomianów ortogonalnych jest wyznaczony jednoznacznie z dokładnością do współczynników przy najwyższych potęgach x^k . Podamy teraz kilka przykładów takich ciągów.

Wielomiany Legendre'a P_k .

Jest to najbardziej naturalnie zdefiniowany ciąg wielomianów ortogonalnych, bowiem w tym przypadku $[a, b] = [-1, 1]$ i waga jest jednostkowa, $\rho = 1$. Mamy

$$P_k(x) = \frac{1}{2^k k!} \frac{d^k}{dx^k} (x^2 - 1)^k,$$

albo

$$\begin{aligned} P_0(x) &= 1, & P_1(x) &= x, \\ P_k(x) &= \frac{2k-1}{k} x P_{k-1}(x) - \frac{k-1}{k} P_{k-2}(x), & k &\geq 2. \end{aligned}$$

Wielomiany Czebyszewa pierwszego rodzaju T_k .

Ten ciąg wielomianów poznaliśmy już w Rozdziale 7.2. Są one ortogonalne w przedziale $[-1, 1]$ z wagą $\rho(x) = (1 - x^2)^{-1/2}$. Przypomnijmy, że

$$\begin{aligned} T_0(x) &= 1, & T_1(x) &= x, \\ T_k(x) &= 2xT_{k-1}(x) - T_{k-2}(x), & k &\geq 2. \end{aligned}$$

Wielomiany Czebyszewa drugiego rodzaju U_k

Ten ciąg wielomianów poznaliśmy w Rozdziale 9.2. Są one ortogonalne w przedziale $[-1, 1]$ z wagą $\rho(x) = \sqrt{1-x^2}$. Przypomnijmy, że

$$\begin{aligned} U_0(x) &= 1, & U_1(x) &= 2x, \\ U_k(x) &= 2xU_{k-1}(x) - U_{k-2}(x), & k &\geq 2. \end{aligned}$$

Wielomiany Laguerre'a L_k

Jest to przykład wielomianów określonych na półprostej, $(a, b) = (0, +\infty)$, przy czym waga $\rho(x) = e^{-x}$. Mamy

$$L_k(x) = \frac{e^x}{k!} \frac{d^k}{dx^k} (x^k e^{-x}),$$

albo

$$\begin{aligned} L_0(x) &= 1, & L_1(x) &= 1-x, \\ L_k(x) &= \frac{2k-1-x}{k} L_{k-1}(x) - \frac{k-1}{k} L_{k-2}(x), & k &\geq 2. \end{aligned}$$

Wielomiany Hermite'a H_k

To z kolei przykład wielomianów ortogonalnych na całej prostej rzeczywistej, a więc $(a, b) = (-\infty, +\infty)$, przy czym waga $\rho(x) = e^{-x^2}$. Wielomiany Hermite'a dane są wzorami

$$H_k(x) = (-1)^k e^{x^2} \frac{d^k}{dx^k} e^{-x^2},$$

albo

$$\begin{aligned} H_0(x) &= 1, & H_1(x) &= 2x, \\ H_k(x) &= 2xH_{k-1}(x) - (2k-2)H_{k-2}(x), & k &\geq 2. \end{aligned}$$

Zauważmy, że we wszystkich tych przykładach kolejny wielomian ortogonalny zależy tylko od dwóch poprzednich. Nie jest to przypadek, ale ogólna reguła. Wielomiany ortogonalne spełniają bowiem formułę trójczłonową, cf. U. 10.1.

Ciągi wielomianów ortogonalnych mają również szereg innych ciekawych własności, co powoduje, że często używane są jako narzędzie do aproksymacji funkcji wielomianami, cf. U. 7.7. Teraz pokażemy tylko jedną bardzo ważną ich własność, a dokładniej, własność zer tych wielomianów, która będzie nam potrzebna do rozwiązania problemu kwadratur maksymalnego rzędu.

Lemat 10.3 *Wielomian ortogonalny p_k ma dokładnie k jednokrotnych zer w przedziale (a, b) .*

Dowód. Niech y_1, y_2, \dots, y_s będą wszystkimi punktami z przedziału (a, b) , w których wielomian p_k zmienia znak. Gdyby lemat nie był prawdziwy to mielibyśmy $s < k$. Wtedy, kładąc

$$w(x) = (x - y_1)(x - y_2) \cdots (x - y_s),$$

mamy $\langle w, p_k \rangle = 0$, bo $w \in \Pi_s$ i $s < k$. Z drugiej strony,

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

bo funkcja podcałkowa jest albo stale dodatnia, albo stale ujemna (z wyjątkiem zbioru miary zero). Ta sprzeczność kończy dowód. \square

10.3 Kwadratury Gaussa

Dla dowolnej n , niech $x_0^*, x_1^*, \dots, x_n^*$ będą (różnymi) zerami $(n+1)$ -szego wielomianu p_{n+1} w ciągu wielomianów ortogonalnych w przestrzeni unitarnej $\mathcal{L}_{2,\rho}(a,b)$, tzn.

$$p_{n+1}(x) = (x - x_0^*)(x - x_1^*) \cdots (x - x_n^*). \quad (10.1)$$

Ponieważ x_j^* leżą w (a,b) , możemy mówić o kwadraturze interpolacyjnej opartej na tych węzłach.

Definicja 10.3 Kwadraturę interpolacyjną Q_n^{GS} opartą na zerach wielomianu ortogonalnego p_{n+1} nazywamy kwadraturą Gaussa.

Okazuje się, że właśnie kwadratury Gaussa mają najwyższy rząd. Dokładniej, mamy następujące twierdzenie.

Twierdzenie 10.3 Kwadratura Gaussa Q_n^{GS} ma najwyższy rząd spośród wszystkich kwadratur opartych na węzłach o łącznej krotności $n+1$, oraz

$$\text{rz}(Q_n^{GS}) = 2n + 2.$$

Dowód. Wobec Lematu 10.1(ii) wystarczy pokazać, że kwadratura Q_n^{GS} jest dokładna dla każdego wielomianu stopnia nie większego niż $2n+1$. Niech $f \in \Pi_{2n+1}$. Niech $w_f \in \Pi_n$ będzie wielomianem interpolującym f w węzłach-zerach x_0^*, \dots, x_n^* wielomianu p_{n+1} . Jeśli $\deg f \leq n$ to oczywiście $w_f = f$ i $Q_n^{GS}(f) = S_\rho(w_f) = S_\rho(f)$. Jeśli zaś

$$n+1 \leq \deg f \leq 2n+1,$$

to $f - w_f$ jest wielomianem stopnia tego samego co f i zeruje się w x_j^* dla $0 \leq j \leq n$. Stąd

$$f(x) - w_f(x) = (x - x_0^*)(x - x_1^*) \cdots (x - x_n^*) g(x),$$

gdzie g jest wielomianem,

$$\deg g = \deg f - (n+1) \leq (2n+1) - (n+1) = n.$$

Korzystając z (10.1) i faktu, że p_{n+1} jest prostopadły do Π_n , ostatecznie otrzymujemy

$$\begin{aligned} S_\rho(f) - Q_n^{GS}(f) &= \int_a^b (f(x) - w_f(x)) \rho(x) dx \\ &= \int_a^b (x - x_0^*) \cdots (x - x_n^*) g(x) \rho(x) dx \\ &= \int_a^b p_{n+1}(x) g(x) \rho(x) dx = \langle p_{n+1}, g \rangle = 0. \quad \square \end{aligned}$$

Zajmiemy się teraz błędem kwadratur Gaussa. Pokażemy, że również ze względu na błąd mają one dobre własności.

Twierdzenie 10.4 Jeśli $f \in C^{2n+2}([a,b])$ to błąd kwadratury Gaussa Q_n^{GS} wyraża się wzorem

$$S_\rho(f) - Q_n^{GS}(f) = \|p_{n+1}\|^2 \frac{f^{(2n+2)}(\xi)}{(2n+2)!},$$

gdzie $\xi \in [a,b]$. Stąd, w szczególności,

$$\sup_{f \in C_M^{2m+2}([a,b])} |S_\rho(f) - Q_n^{GS}(f)| = \frac{M \|p_{n+1}\|^2}{(2n+2)!}.$$

Dowód. Niech $w_f \in \Pi_n$ będzie wielomianem interpolującym f w zerach x_j^* wielomianu p_{n+1} . Niech $\tilde{w}_f \in \Pi_{2n+1}$ będzie z kolei wielomianem (Hermite'a) interpolującym f w dwukrotnych węzłach x_j^* , tzn. takim, że $\tilde{w}_f(x_j^*) = f(x_j^*)$ i $\tilde{w}'_f(x_j^*) = f'(x_j^*)$, $0 \leq j \leq n$. Ponieważ $\text{rz}(Q_n^{GS}) = 2n + 2$ to $Q_n^{GS}(\tilde{w}_f) = S_\rho(\tilde{w}_f)$, a stąd i ze wzoru na błąd interpolacji Hermite'a mamy

$$\begin{aligned} S_\rho(f) - Q_n^{GS}(f) &= S_\rho(f) - Q_n^{GS}(w_f) = S_\rho(f) - Q_n^{GS}(\tilde{w}_f) \\ &= S_\rho(f) - S_\rho(\tilde{w}_f) = \int_a^b (f(x) - \tilde{w}_f(x))\rho(x) dx \\ &= \int_a^b (x - x_0^*)^2 \cdots (x - x_n^*)^2 f[x_0^*, \dots, x_n^*, x]\rho(x) dx \\ &= \int_a^b p_{n+1}^2(x)\rho(x)f[x_0^*, \dots, x_n^*, x] dx. \end{aligned}$$

Ponieważ funkcja $p_{n+1}^2(x)\rho(x)$ jest prawie wszędzie dodatnia, możemy teraz zastosować twierdzenie o wartości średniej, aby ostatecznie otrzymać

$$S_\rho(f) - Q_n^{GS}(f) = \frac{f^{(2n+2)}(\xi)}{(2n+2)!} \int_a^b p_{n+1}^2(x)\rho(x) dx = \frac{f^{(2n+2)}(\xi)}{(2n+2)!} \|p_{n+1}\|^2,$$

co kończy dowód. \square

Na końcu, zwrócimy jeszcze uwagę na inną, bardzo ważną własność kwadratur Gaussa, a mianowicie, że ich współczynniki są dodatnie. Rzeczywiście, zapisując

$$Q_n^{GS}(f) = \sum_{j=0}^n a_j f(x_j)$$

i podstawiając

$$f_j(x) = (x - x_0^*)^2 \cdots (x - x_{j-1}^*)^2 (x - x_{j+1}^*)^2 \cdots (x - x_n^*)^2$$

mamy, że $f_j \in \Pi_{2n}$ i f_j jest prawie wszędzie dodatnia. Stąd

$$0 < S_\rho(f_j) = Q_n^{GS}(f_j) = a_j f_j(x_j^*)$$

i $a_j > 0$, bo $f_j(x_j^*) > 0$.

Z przedstwionych własności wynika, że kwadratury Gaussa spełniają założenia Twierdzenia 10.2 z $A = \int_a^b \rho(x) dx$ (bo suma współczynników kwadratury jest całką z funkcji $f = 1$), mamy więc zbieżność $Q_n^{GS}(f)$ do całki $S_\rho(f)$ dla dowolnej funkcji ciągłej f .

Uwagi i uzupełnienia

U. 10.1 Pokażemy teraz, że wielomiany ortogonalne $\{p_k\}_{k \geq 0}$ w danej przestrzeni $\mathcal{L}_{2,\rho}(a,b)$ spełniają następującą formułę trójczłonową. Załóżmy dla uproszczenia, że współczynnik przy x^k w wielomianie p_k jest dla każdego k jedynką. Wtedy istnieją liczby β_k (dla $k \geq 1$) i $\gamma_k > 0$ (dla $k \geq 2$) takie, że

$$\begin{aligned} p_0(x) &= 1, \\ p_1(x) &= (x - \beta_1), \\ p_k(x) &= (x - \beta_k)p_{k-1}(x) - \gamma_k p_{k-2}(x), \quad k \geq 2. \end{aligned} \tag{10.2}$$

Aby to pokazać zauważmy, że p_k można dla $k \geq 1$ przedstawić w postaci rozwinięcia

$$p_k(x) = (x - c_{k-1})p_{k-1}(x) + \sum_{j=0}^{k-2} c_j p_j(x).$$

Mnożąc skalarnie obie strony tego równania przez p_s dla $0 \leq s \leq k-3$, otrzymujemy

$$0 = \langle p_k(x), p_s(x) \rangle = \langle (x - c_{k-1})p_{k-1}(x), p_s(x) \rangle + c_s \langle p_s(x), p_s(x) \rangle.$$

Wobec tego, że $(x - c_{k-1})p_{k-1}(x)$ jest wielomianem stopnia mniejszego niż $k-1$, mamy

$$\langle (x - c_{k-1})p_{k-1}(x), p_s(x) \rangle = \langle p_{k-1}(x), (x - c_{k-1})p_s(x) \rangle = 0,$$

a stąd $c_s \langle p_s(x), p_s(x) \rangle = 0$ i $c_s = 0$. Możemy więc napisać, że dla $k=1$ mamy $p_1(x) = (x - \beta_1)$, a dla $k \geq 2$,

$$p_k(x) = (x - \beta_k)p_{k-1}(x) - \gamma_k p_{k-2}(x), \quad (10.3)$$

gdzie $\beta_k = c_{k-1}$ i $\gamma_k = c_{k-2}$. Aby jawnie wyznaczyć β_k i γ_k , pomnożymy skalarnie obie strony równania (10.3) kolejno przez p_{k-1} i p_{k-2} . Otrzymujemy

$$\begin{aligned} 0 &= \langle p_k(x), p_{k-1}(x) \rangle \\ &= \langle (x - \beta_k)p_{k-1}(x), p_{k-1}(x) \rangle - \gamma_k \langle p_{k-2}(x), p_{k-1}(x) \rangle \\ &= \langle x p_{k-1}(x), p_{k-1}(x) \rangle - \beta_k \langle p_{k-1}(x), p_{k-1}(x) \rangle, \end{aligned}$$

czyli

$$\beta_k = \frac{\langle x p_{k-1}(x), p_{k-1}(x) \rangle}{\langle p_{k-1}(x), p_{k-1}(x) \rangle},$$

oraz

$$\begin{aligned} 0 &= \langle p_k(x), p_{k-2}(x) \rangle \\ &= \langle (x - \beta_k)p_{k-1}(x), p_{k-2}(x) \rangle - \gamma_k \langle p_{k-2}(x), p_{k-2}(x) \rangle \\ &= \langle p_{k-1}(x), x p_{k-2}(x) \rangle - \gamma_k \langle p_{k-2}(x), p_{k-2}(x) \rangle, \end{aligned}$$

a stąd i z równości $\langle p_{k-1}(x), x p_{k-2}(x) \rangle = \langle p_{k-1}(x), p_{k-1}(x) \rangle$,

$$\gamma_k = \frac{\langle p_{k-1}(x), p_{k-1}(x) \rangle}{\langle p_{k-2}(x), p_{k-2}(x) \rangle}.$$

Zauważmy, że z formuły trójczłonowej wynika w szczególności algorytm wyznaczenia ciągu wielomianów ortogonalnych. Wystarczy bowiem wyznaczać kolejne współczynniki β_k i γ_k (obliczając odpowiednie iloczyny skalarne $\langle x p_k(x), p_k(x) \rangle$ i $\langle p_k(x), p_k(x) \rangle$) i stosować wzór rekurencyjny (10.2). Dodajmy jeszcze, że w obliczeniach numerycznych najlepiej jest przechowywać informację o ciągu $\{p_k\}_{k \geq 0}$ w postaci liczb β_k i γ_k .

U. 10.2 Załóżmy, że całkę $S_\rho(f) = \int_a^b f(x)\rho(x) dx$ dla funkcji $f \in C^r([a, b])$ aproksymujemy przy pomocy kwadratury $Q_n(f) = \sum_{i=0}^n a_i f(x_i)$, której rząd $\text{rz}(Q_n) \geq r$. Rozwińmy f w szereg Taylora z resztą całkową,

$$f(x) = (\mathcal{T}_r f)(x) + \int_a^b \frac{(x-t)_+^{r-1}}{(r-1)!} f^{(r)}(t) dt, \quad \text{gdzie} \quad (\mathcal{T}_r f)(x) = \sum_{k=0}^{r-1} \frac{(x-a)^k}{k!} f^{(k)}(a).$$

Ponieważ $S(\mathcal{T}_r f) = Q_n(\mathcal{T}_r f)$, błąd kwadratury można wyrazić w postaci

$$\begin{aligned} S(f) - Q_n(f) &= \int_a^b \left(\int_a^b \frac{(x-t)_+^{r-1}}{(r-1)!} f^{(r)}(t) dt \right) \rho(x) dx - \sum_{i=0}^n a_i \int_a^b \frac{(x_i-t)_+^{r-1}}{(r-1)!} f^{(r)}(t) dt \\ &= \int_a^b \left(\int_a^b \frac{(x-t)_+^{r-1}}{(r-1)!} \rho(x) dx - \sum_{i=0}^n a_i \frac{(x_i-t)_+^{r-1}}{(r-1)!} \right) f^{(r)}(t) dt. \end{aligned}$$

Oznaczając $\psi_{t,r}(x) = \frac{(x-t)_+^{r-1}}{(r-1)!}$ mamy, że

$$S(f) - Q_n(f) = \int_a^b K_{n,r}(t) f^{(r)}(t) dt, \quad (10.4)$$

gdzie wartość $K_{n,r}(t) = S(\psi_{t,r}) - Q_n(\psi_{t,r})$ jest błędem kwadratury dla ‘funkcji testowej’ $\psi_{t,r}$. Funkcję $K_{n,r}$ nazywamy *jądrem Peano* funkcjonału $S - Q_n$.

Wyrażenie (10.4) pozwala w wygodny sposób wyrazić błędy kwadratur. Na przykład, jeśli $K_{n,r}$ nie zmienia znaku w $[a, b]$, co ma miejsce w przypadku kwadratur Newtona-Cotesa Q_n^{NC} , to

$$S(f) - Q_n(f) = \left(\int_a^b K_{n,r}(t) dt \right) f^{(r)}(\xi), \quad \xi \in [a, b].$$

Natomiast dla dowolnej kwadratury mamy

$$\sup_{f \in C_M^r([a,b])} |S(f) - Q_n(f)| = \left(\int_a^b |K_{n,r}(t)| dt \right) M. \quad (10.5)$$

Rzeczywiście, równość mamy dla funkcji g , której pochodna $g^{(r)}$ przyjmuje $+M$ na przedziałach gdzie $K_{n,r}$ jest nieujemna i $-M$ na pozostałych przedziałach. Co prawda, taka funkcja nie należy do $C_M^r([a, b])$, ale dla dowolnego małego $\epsilon > 0$ można znaleźć funkcję f taką, że $f \in C_M^r([a, b])$ oraz

$$\|(f - g)^{(r)}\|_{\mathcal{L}_1} = \int_a^b |(f - g)^{(r)}(x)| dx \leq \epsilon.$$

Wtedy

$$\begin{aligned} |S(f) - Q_n(f)| &= \left| \int_a^b K_{n,r}(t) f^{(r)}(t) dt \right| \geq \left| \int_a^b K_{n,r}(t) g^{(r)}(t) dt \right| - \left| \int_a^b K_{n,r}(t) (f - g)^{(r)}(t) dt \right| \\ &\geq \left(\int_a^b |K_{n,r}(t)| dt \right) M - \epsilon \operatorname{ess\,sup}_{a \leq t \leq b} |K_{n,r}(t)|, \end{aligned}$$

co dowodzi (10.5) wobec dowolności ϵ .

U. 10.3 Niech $Q(f) = \sum_{i=0}^n a_i f(t_i)$ będzie kwadraturą o rzędzie $\operatorname{rz}(Q) \geq r$ dla aproksymacji całki $\int_0^1 f(x) dx$. Niech K będzie jądrem Peano tej kwadratury dla regularności r . Wtedy kwadratura

$$Q_{u,v}(f) = (v - u) \sum_{i=0}^n a_i f(x_i), \quad \text{gdzie } x_i = u + t_i(v - u),$$

aproksymująca całkę $S_{u,v}(f) = \int_u^v f(x) dx$, ma rząd $\operatorname{rz}(Q_{u,v}) = \operatorname{rz}(Q) \geq r$, a jej jądro Peano wynosi

$$K_{u,v}(x) = (v - u)^r K\left(\frac{x - u}{v - u}\right), \quad (10.6)$$

cf. Ćw. 10.13. Stąd, wobec (10.5), dla $f \in C^r([u, v])$ mamy

$$|S_{u,v}(f) - Q_{u,v}(f)| \leq \|K_{u,v}\|_{\mathcal{L}_1(u,v)} \|f^{(r)}\|_{C([u,v])} = (v - u)^{r+1} C \|f^{(r)}\|_{C([u,v])}, \quad C = \|K\|_{\mathcal{L}_1(0,1)}.$$

Rozpatrzmy teraz kwadraturę złożoną \bar{Q}_k dla całki $S(f) = \int_a^b f(x) dx$, opartą na równomiernym podziale przedziału $[a, b]$ na k podprzedziałów $[x_{i-1}, x_i]$, gdzie w każdym podprzedziale stosujemy Q_{x_{i-1}, x_i} . Z ostatniej nierówności wynika, że wtedy

$$|S(f) - \bar{Q}_k(f)| \leq (b - a)^{r+1} C \|f^{(r)}\|_{C([a,b])} k^{-r}.$$

Stąd dostajemy ważny wniosek, że mamy zbieżność k^{-r} o ile rząd kwadratury Q użytej jako podstawy wynosi co najmniej r . Przypomnijmy, że w tym przypadku, czyli dla wagi $\rho = 1$, rząd jest maksymalny i wynosi $2n + 2$ gdy Q jest kwadraturą Legendre’a na przedziale $[0, 1]$.

Ćwiczenia

Ćw. 10.1 Uzasadnij, że: (a) jeśli kwadratura jest dokładna dla dowolnych $n + 1$ wielomianów tworzących bazę w Π_n , to jest ona rzędu co najmniej $n + 1$, oraz (b) jeśli kwadratura jest rzędu $n + 1$ to jest ona niedokładna dla każdego wielomianu stopnia dokładnie $n + 1$.

Ćw. 10.2 Ile wynosi maksymalny rząd kwadratur opartych na $n + 1$ węzłach, przy czym każdy węzeł jest dwukrotny?

Ćw. 10.3 Pokaż, że wielomiany Czebyszewa pierwszego rodzaju $\{T_k\}_{k \geq 0}$ tworzą układ ortogonalny w przestrzeni $\mathcal{L}_{2,\rho}(-1, 1)$, gdzie waga $\rho(x) = (1 - x^2)^{-1/2}$, tzn. iloczyn skalarny

$$\langle T_i, T_j \rangle = \int_{-1}^1 \frac{T_i(x)T_j(x)}{\sqrt{1-x^2}} dx = \begin{cases} \pi & i = j = 0, \\ \pi/2 & i = j \neq 0, \\ 0 & i \neq j. \end{cases}$$

Ćw. 10.4 Pokaż, że wielomiany Czebyszewa drugiego rodzaju $\{U_k\}_{k \geq 0}$ tworzą układ ortogonalny w przestrzeni $\mathcal{L}_{2,\rho}(-1, 1)$, gdzie waga $\rho(x) = (1 - x^2)^{1/2}$, tzn. iloczyn skalarny

$$\langle U_i, U_j \rangle := \int_{-1}^1 U_i(x)U_j(x)\sqrt{1-x^2} dx = \begin{cases} \pi/2 & i = j, \\ 0 & i \neq j. \end{cases}$$

Ćw. 10.5 Rozpatrzmy ciąg wielomianów ortogonalnych $\{p_n\}_{n=0}^\infty$ w przestrzeni $\mathcal{L}_{2,\rho}(-1, 1)$. Wykaż, że jeśli waga ρ jest parzysta, tzn. $\rho(x) = \rho(-x)$, to p_{2n} są wielomianami parzystymi, a p_{2n+1} są wielomianami nieparzystymi dla $n = 0, 1, 2, \dots$.

Ćw. 10.6 Niech x_j dla $0 \leq j \leq n$ będą zerami $(n + 1)$ -szego wielomianu ortogonalnego w przestrzeni $\mathcal{L}_{2,\rho}(a, b)$. Niech dalej l_j będą odpowiadającymi tym węzłom wielomianami Lagrange'a. Wykaż, że dla $f, g \in \Pi_n$ iloczyn skalarny w $\mathcal{L}_{2,\rho}(a, b)$ można wyrazić jako

$$\langle f, g \rangle_{\mathcal{L}_{2,\rho}} = \int_{-1}^1 f(x)g(x)\rho(x) dx = \sum_{j=0}^n w_j f(x_j)g(x_j), \quad \text{gdzie } w_j = \langle l_j, l_j \rangle_{\mathcal{L}_{2,\rho}} = \|l_j\|_{\mathcal{L}_{2,\rho}}^2.$$

Ćw. 10.7 Załóżmy, że dane są liczby β_k i γ_k definiujące ciąg wielomianów ortogonalnych przez formułę trójczłonową. Zaproponuj ekonomiczny (tzn. o koszcie proporcjonalnym do n) algorytm obliczania wartości n -tego wielomianu ortogonalnego w danym punkcie x , wykorzystujący formułę trójczłonową.

Ćw. 10.8 Przeprowadzając ortogonalizację Grama-Schmidta bazy potęgowej $\{1, x, x^2, x^3\}$ znajdź wielomiany ortogonalne Legendre'a stopnia 0, 1, 2, 3, tzn. wielomiany ortogonalne na przedziale $[-1, 1]$ z wagą $\rho = 1$. Następnie wskaż zera tych wielomianów, czyli węzły odpowiednich kwadratur Gaussa-Legendre'a.

Ćw. 10.9 Rozpatrzmy całkowanie z wagą $S_\rho(f) = \int_a^b f(x)\rho(x) dx$, gdzie $-\infty < a < b < +\infty$. Ile wynosi maksymalny rząd kwadratury opartej na $n + 1$ węzłach zawierających końce przedziału całkowania,

$$a = x_0 \leq x_1 \leq \dots \leq x_{n-1} \leq x_n = b,$$

dla aproksymacji całki $S_\rho(f)$?

Ćw. 10.10 Wykaż, że całka

$$\int_a^b (x - x_0)^2 (x - x_1)^2 \dots (x - x_n)^2 \rho(x) dx,$$

gdzie ρ jest pewną wagą, jest najmniejsza wtedy i tylko wtedy gdy za x_0, \dots, x_n weźmiemy zera $(n + 1)$ -szego wielomianu ortogonalnego w przestrzeni $\mathcal{L}_{2,\rho}(a, b)$.

Ćw. 10.11 Uzasadnij, że kwadratura prostokątów jest kwadraturą Legendre'a, natomiast żadna z kwadratur Newtona-Cotesa Q_n^{NC} nie jest kwadraturą Gaussa dla jakiejkolwiek wagi ρ .

Ćw. 10.12 Dla wielomianów Legendre'a P_n prawdziwa jest równość

$$\|P_n\|_{\mathcal{L}_2(-1,1)} = \int_{-1}^1 P_n^2(x) dx = \frac{2}{2n+1}.$$

Niech x_j dla $0 \leq j \leq n$ będą zerami $(n+1)$ -szego wielomianu Legendre'a. Pokaż, że

$$\int_{-1}^1 (x-x_0)^2(x-x_1)^2 \cdots (x-x_n)^2 dx = \frac{2^{2n+3}}{2n+3} \left(\frac{1 \cdot 2 \cdots n \cdot (n+1)}{(n+2)(n+3) \cdots (2n+2)} \right)^2.$$

Ćw. 10.13 Wykaż prawdziwość równości (10.6) z U. 10.3.

Ćw. 10.14 Wykaż, że jądra Peano z U. 10.2 kwadratur prostokątów R , trapezów T i parabol P dla aproksymacji całki $\int_0^1 f(x) dx$ są symetryczne względem $\frac{1}{2}$ i dla $t \in [0, \frac{1}{2}]$ wynoszą odpowiednio

$$K_{0,2}^R(t) = \frac{1}{2}t^2, \quad K_{1,2}^T(t) = -\frac{1}{2}t(1-t), \quad K_{2,4}^P(t) = -\frac{1}{24}t^3\left(\frac{2}{3}-t\right).$$

Wywnioskuj stąd formuły na błędy tych kwadratur z Ćw. 9.4 i Twierdzenia 9.3.

Ćw. 10.15 Niech Q_a^I będzie kwadraturą interpolacyjną opartą na dwóch jednokrotnych węzłach $\pm a$, gdzie $a > 0$, dla aproksymacji całki $\int_{-1}^1 f(x) dx$. Wykaż, że maksymalny błąd w klasie $C_1^2([-1, 1])$ wynosi

$$\sup_{f \in C_1^2([-1,1])} |S(f) - Q_a^I(f)| = \begin{cases} \left(\frac{1}{3} - a^2\right), & 0 < a \leq \frac{1}{2}, \\ \left(\frac{1}{3} - a^2\right) + \frac{4}{3}(2a-1)^{\frac{3}{2}}, & \frac{1}{2} < a \leq 1. \end{cases}$$

Czy maksymalny błąd jest najmniejszy gdy kwadratura oparta jest na zerach wielomianu Czebyszewa U_2 albo wielomianu Legendre'a P_2 ?

Ćw. 10.16 Wykaż, że w klasie funkcji $f \in C^1([-1, 1])$ takich, że $f(\pm \frac{3}{4}) = 0$ i f' spełnia warunek Lipschitza ze stałą 1, całka $\int_{-1}^1 f(x) dx$ jest maksymalizowana przez funkcję

$$f^*(x) = \begin{cases} \frac{1}{2}x^2 + \sqrt{2}x + \left(\frac{3\sqrt{2}}{4} - \frac{9}{32}\right) & -1 \leq x \leq -\frac{1}{2}\sqrt{2}, \\ -\frac{1}{2}x^2 + \frac{9}{32}, & -\frac{1}{2}\sqrt{2} \leq x \leq \frac{1}{2}\sqrt{2}, \\ \frac{1}{2}x^2 - \sqrt{2}x + \left(\frac{3\sqrt{2}}{4} - \frac{9}{32}\right), & \frac{1}{2}\sqrt{2} \leq x \leq 1. \end{cases}$$

Rozdział 11

Iteracje dla równań liniowych

Algorytmy rozwiązywania układów równań liniowych postaci

$$A\vec{x} = \vec{b},$$

gdzie A jest nieosobliwą macierzą rzeczywistą $n \times n$, a \vec{b} jest wektorem rzeczywistym w \mathbb{R}^n , które rozpatrywaliśmy w Rozdziałach 3, 4 i 5, należą do grupy algorytmów *dokładnych* albo *bezpośrednich*. To znaczy, że po wykonaniu skończonej liczby dopuszczalnych operacji elementarnych dostajemy w arytmetyce idealnej dokładne rozwiązanie

$$\vec{x}^* = A^{-1}\vec{b}.$$

W tym rozdziale zajmujemy się *algorytmami iteracyjnymi* rozwiązywania układów równań liniowych. Polegają one na tym, że, startując z pewnego przybliżenia początkowego \vec{x}_0 , konstruuje się ciąg kolejnych przybliżeń

$$\vec{x}_k = \Phi_k(A, \vec{b}; \vec{x}_0), \quad k = 1, 2, \dots,$$

które w granicy osiągają rozwiązanie dokładne,

$$\lim_{k \rightarrow \infty} \vec{x}_k = \vec{x}^*.$$

11.1 Kiedy stosujemy iteracje?

Jasne jest, że algorytmy iteracyjne stosujemy wtedy, gdy są one konkurencyjne w stosunku do algorytmów bezpośrednich. Dlatego przekształcenia Φ_k należy wybierać tak, aby kolejne przybliżenia można było łatwo obliczać i jednocześnie kolejne błędy $\|\vec{x}_k - \vec{x}^*\|$ szybko zbiegały do zera.

Zwykle zakłada się również, że dokładne rozwiązanie \vec{x}^* jest punktem stałym przekształcenia $\Phi_k(A, \vec{b}; \cdot)$. Wtedy kolejne błędy spełniają zależność

$$\vec{x}_k - \vec{x}^* = \Phi_k(A, \vec{b}; \vec{x}_0) - \Phi_k(A, \vec{b}; \vec{x}^*).$$

Jeśli teraz $\Phi_k(A, \vec{b}; \cdot)$ są lipschitzowskie ze stałymi $m_k < +\infty$, tzn. dla pewnej normy wektorowej $\|\cdot\|$ mamy

$$\|\Phi_k(\vec{x}) - \Phi_k(\vec{y})\| \leq m_k \|\vec{x} - \vec{y}\| \quad \text{dla } \vec{x}, \vec{y} \in \mathbb{R}^n$$

to

$$\|\vec{x}_k - \vec{x}^*\| \leq m_k \|\vec{x}_0 - \vec{x}^*\|.$$

Warunek $\lim_{k \rightarrow \infty} m_k = 0$ jest więc dostateczny na to, aby metoda była zbieżna dla dowolnego przybliżenia początkowego \vec{x}_0 , przy czym szybkość zbieżności zależy od tego, jak szybko m_k maleją do zera. Dla większości stosowanych metod Φ_k jest funkcją liniową błędu początkowego, tzn.

$$\Phi_k(A, \vec{b}; \vec{x}_0 - \vec{x}^*) = M_k(\vec{x}_0 - \vec{x}^*),$$

gdzie M_k jest pewną macierzą. Wtedy jako m_k można przyjąć dowolną normę tej macierzy,

$$m_k = \|M_k\| = \sup_{\|\vec{x}\|=1} \|M_k \vec{x}\|.$$

Dla ilustracji, rozpatrzmy ogólną metodę *iteracji prostej*, w której

$$\vec{x}_k = B\vec{x}_{k-1} + \vec{c}, \quad (11.1)$$

dla pewnej macierzy B formatu $n \times n$ i wektora $\vec{c} \in \mathbb{R}^n$. W tym przypadku

$$\vec{x}_k - \vec{x}^* = B^k(\vec{x}_0 - \vec{x}^*),$$

a stąd i z nierówności $\|B^k\| \leq \|B\|^k$, mamy

$$\|\vec{x}_k - \vec{x}^*\| \leq \|B\|^k \|\vec{x}_0 - \vec{x}^*\|.$$

Warunkiem dostatecznym zbieżności iteracji prostych jest więc $\|B\| < 1$. Mówimy, że metoda jest zbieżna liniowo z ilorazem $\|B\|$.

Przykład 11.1 Rozkładając macierz $A = (a_{i,j})_{i,j=1}^n$ na sumę

$$A = D + C,$$

gdzie D jest macierzą diagonalną składającą się z wyrazów stojących na głównej przekątnej macierzy A , układ $A\vec{x} = \vec{b}$ jest równoważny układowi

$$D\vec{x} = -C\vec{x} + \vec{b},$$

a stąd (o ile na przekątnej macierzy A nie mamy zera) otrzymujemy metodę iteracyjną

$$\vec{x}_k = B\vec{x}_{k-1} + \vec{c},$$

gdzie $B = -D^{-1}C$ i $\vec{c} = D^{-1}\vec{b}$, zwaną *metodą Jacobiego*.

W metodzie Jacobiego warunek dostateczny zbieżności, $\|B\| < 1$, jest spełniony wtedy, gdy macierz A ma dominującą przekątną, tzn. gdy

$$2|a_{i,i}| > \sum_{j=1}^n |a_{i,j}|, \quad 1 \leq i \leq n. \quad (11.2)$$

Rzeczywiście, ponieważ wyraz (i, j) macierzy $D^{-1}C$ wynosi 0 dla $i = j$ i $a_{i,j}/a_{i,i}$ dla $i \neq j$, to

$$\|D^{-1}C\|_\infty = \max_{1 \leq i \leq n} \sum_{j=1, j \neq i}^n \frac{|a_{i,j}|}{|a_{i,i}|} = \max_{1 \leq i \leq n} \sum_{j=1}^n \frac{|a_{i,j}|}{|a_{i,i}|} - 1 < 1,$$

przy czym ostatnia nierówność wynika z (11.2).

Inne przykłady iteracji prostych podane są w U. 11.3 i Ćw. 11.2.

Zastanówmy się teraz nad złożonością metod iteracyjnych. Ponieważ możemy jedynie znaleźć pewne przybliżenie rozwiązania dokładnego \vec{x}^* , przez złożoność metody będziemy rozumieli koszt kombinatoryczny obliczenia \vec{x}_k zadaną dokładnością $\varepsilon > 0$. Dla uproszczenia założymy, że metoda jest zbieżna liniowo z ilorazem m . Zauważmy, że aby zredukować błąd początkowy do $\varepsilon > 0$, wystarczy wykonać $k = k(\varepsilon)$ iteracji, gdzie k spełnia

$$m^k \|\vec{x}_0 - \vec{x}^*\| \leq \varepsilon,$$

czyli

$$k \geq \frac{\log(1/\varepsilon) - \log(1/\|\vec{x}_0 - \vec{x}^*\|)}{\log(1/m)}.$$

Liczba ta zależy więc w istotny sposób od błędu początkowego i (przede wszystkim) od stałej Lipschitza m , natomiast zależność od dokładności ε i wymiaru n układu jest dużo mniej istotna. Zakładając, że koszt jednej iteracji wynosi $c = c(n)$ (zwykle $c(n)$ jest tym mniejszy, im mniejsza jest liczba niezerowych elementów macierzy A), złożoność metody jest proporcjonalna do

$$c(n) \frac{\log(1/\varepsilon)}{\log(1/m)}.$$

Stąd oczywisty wniosek, że metody iteracyjne warto stosować zamiast metod bezpośrednich w przypadku gdy:

- wymiar n układu $A\vec{x} = \vec{b}$ jest “duży”, oraz
- macierz A układu jest “rozrzedzona”, tzn. ma stosunkowo niewielką liczbę elementów niezerowych, np. proporcjonalną do n .

Układy o tych własnościach powstają często przy numerycznym rozwiązywaniu równań różniczkowych cząstkowych.

Zaletą metod iteracyjnych jest również ich prostota, przez co są one łatwe do zaprogramowania.

11.2 Metoda Czebyszewa

Zauważyliśmy, że ze względu na szybkość zbieżności metody iteracyjnej ważne jest, aby stałe lipschitzowskie m_k odwzorowań Φ_k miały jak najszybciej. *Metoda Czebyszewa*, którą przedstawimy w tym rozdziale, jest właśnie próbą minimalizacji tych stałych.

Przy opisie metody skorzystamy z następujących dwóch faktów, które teraz przypomnimy i sformułujemy jako lematy.

Lemat 11.1 *Jeśli macierz A jest symetryczna, to istnieje ortonormalna w \mathbb{R}^n baza jej wektorów własnych $\vec{\xi}_j$, $1 \leq j \leq n$, tzn.*

$$\langle \vec{\xi}_i, \vec{\xi}_j \rangle_2 = \vec{\xi}_i^T \vec{\xi}_j = \begin{cases} 0 & i \neq j, \\ 1 & i = j, \end{cases}$$

a odpowiadające im wartości własne λ_j , $A\vec{\xi}_j = \lambda_j\vec{\xi}_j$, są rzeczywiste. Jeśli ponadto A jest dodatnio określona to λ_j są dodatnie.

Lemat 11.2 Niech macierz A będzie symetryczna i niech

$$|\lambda_1| \geq |\lambda_2| \geq \dots \geq |\lambda_n|$$

będą jej, być może wielokrotnymi, wartościami własnymi. Wtedy

$$\|A\|_2 = \sup_{\vec{x} \neq \vec{0}} \frac{\|A\vec{x}\|_2}{\|\vec{x}\|_2} = |\lambda_1|.$$

Jeśli ponadto A jest nieosobliwa, tzn. $|\lambda_n| > 0$, to

$$\|A^{-1}\|_2 = \frac{1}{|\lambda_n|}.$$

Lemat 11.1 jest znanym faktem z algebry liniowej, więc dowód pominiemy. Dowód Lematu 11.2 podany jest w U. 11.1.

Zakładamy, że macierz A wyjściowego układu równań jest symetryczna i dodatnio określona, tzn.

$$A = A^T > 0,$$

a jej wartości własne λ_j leżą w znanym przedziale $[a, b]$, gdzie

$$0 < a \leq \lambda_n \leq \dots \leq \lambda_1 = \|A\|_2 \leq b < +\infty.$$

W metodzie Czebyszewa kolejne przybliżenia \vec{x}_k konstruujemy tak, aby była spełniona równość

$$(\vec{x}_k - \vec{x}^*) = W_k(A)(\vec{x}_0 - \vec{x}^*), \quad (11.3)$$

gdzie $W_k(A)$ jest macierzą, która wielomianowo zależy od A ,

$$W_k(A) = \sum_{j=0}^k a_j^{(k)} A^j,$$

i ma możliwie małą normę. (Dalej będziemy używać tego samego symbolu W_k do oznaczenia wielomianu zmiennej rzeczywistej t , tzn. $W_k(t) = \sum_{j=0}^k a_j^{(k)} t^j$.) Wobec (11.3) możemy napisać, że

$$\vec{x}_k = W_k(A)\vec{x}_0 + (I - W_k(A))\vec{x}^*.$$

Aby zapewnić konstruowalność przybliżenia \vec{x}_k , musimy umieć obliczać $(I - W_k(A))\vec{x}^*$. Ponieważ

$$\begin{aligned} (I - W_k(A))\vec{x}^* &= - \sum_{j=1}^k a_j^{(k)} A^j \vec{x}^* + (1 - a_0^{(k)})\vec{x}^* \\ &= - \sum_{j=0}^{k-1} a_{j+1}^{(k)} A^j \vec{x}^* + (1 - a_0^{(k)})\vec{x}^*, \end{aligned}$$

dla konstruowalności \vec{x}_k wystarczy założyć, że $a_0^{(k)} = 1$, co odpowiada warunkowi

$$W_k(0) = 1 \quad \text{dla wszystkich } k. \quad (11.4)$$

Z równości (11.3) mamy

$$\|\vec{x}_k - \vec{x}^*\|_2 \leq \|W_k(A)\|_2 \|\vec{x}_0 - \vec{x}^*\|_2.$$

Idea metody polega teraz na wyborze wielomianów W_k tak, aby norma $\|W_k(A)\|_2$ była jak najmniejsza. W tym celu zauważmy, że macierz $W_k(A)$ jest symetryczna, a jej wartości własne wynoszą $W_k(\lambda_j)$, gdzie λ_j są wartościami własnymi macierzy A , cf. Ćw. 11.1. Stąd

$$\|W_k(A)\|_2 \leq \max_{a \leq t \leq b} |W_k(t)| = \|W_k\|_{C([a,b])}.$$

Rozwiązanie problemu minimalizacji ostatniego maksimum ze względu na wszystkie wielomiany $W_k \in \Pi_k$ takie, że $W_k(0) = 1$, jest podane w Rozdziale 7, Ćw. 7.9. Przypomnijmy, że optymalny wielomian dany jest wzorem

$$W_k^*(t) = \frac{T_k(h(t))}{T_k(h(0))},$$

gdzie T_k jest k -tym wielomianem Czebyszewa, a

$$h(t) = \frac{2t - (b+a)}{b-a}.$$

Przypomnijmy również, że wielomiany Czebyszewa spełniają formułę trójczłonową, $T_0(t) = 1$, $T_1(t) = t$, oraz $T_k(t) = 2tT_{k-1}(t) - T_{k-2}(t)$ dla $k \geq 2$. Pozwala to na rekurencyjną konstrukcję kolejnych \vec{x}_k w zależności od \vec{x}_{k-1} i \vec{x}_{k-2} . Dokładniej, dla pierwszego przybliżenia mamy

$$W_1^*(A) = \frac{T_1(h(A))}{T_1(h(0))} = \frac{\left(\frac{2A - (b+a)I}{b-a}\right)}{\left(-\frac{b+a}{b-a}\right)} = I - \frac{2}{b+a}A,$$

a stąd

$$\vec{x}_1 = W_1^*(A)\vec{x}_0 + (I - W_1(A))\vec{x}^* = \vec{x}_0 + \frac{2}{b+a}(\vec{b} - A\vec{x}_0) = \vec{x}_0 + \frac{2}{b+a}\vec{r}_0,$$

gdzie $\vec{r}_0 = \vec{b} - A\vec{x}_0$ jest początkowym residuum.

Dla $k \geq 2$ wykorzystamy formułę trójczłonową dla wielomianów Czebyszewa. Oznaczając $t_j = T_j(h(0))$ mamy

$$\begin{aligned} W_k^*(A) &= \frac{T_k(h(A))}{t_k} = \frac{2h(A)T_{k-1}(h(A)) - T_{k-2}(h(A))}{t_k} \\ &= 2\frac{t_{k-1}}{t_k}h(A)\frac{T_{k-1}(h(A))}{t_{k-1}} - \frac{t_{k-2}}{t_k}\frac{T_{k-2}(h(A))}{t_{k-2}} \\ &= 2\frac{t_{k-1}}{t_k}h(A)W_{k-1}^*(A) - \frac{t_{k-2}}{t_k}W_{k-2}^*(A). \end{aligned}$$

Stąd

$$\begin{aligned} \vec{x}_k - \vec{x}^* &= W_k^*(A)(\vec{x}_0 - \vec{x}^*) \\ &= 2\frac{t_{k-1}}{t_k}h(A)W_{k-1}^*(A)(\vec{x}_0 - \vec{x}^*) - \frac{t_{k-2}}{t_k}W_{k-2}^*(A)(\vec{x}_0 - \vec{x}^*) \\ &= 2\frac{t_{k-1}}{t_k}h(A)(\vec{x}_{k-1} - \vec{x}^*) - \frac{t_{k-2}}{t_k}(\vec{x}_{k-2} - \vec{x}^*) \\ &= 2\frac{t_{k-1}}{t_k}\left(\frac{2}{b-a}A - \frac{b+a}{b-a}I\right)(\vec{x}_{k-1} - \vec{x}^*) - \frac{t_{k-2}}{t_k}(\vec{x}_{k-2} - \vec{x}^*) \\ &= -\frac{4}{b-a}\frac{t_{k-1}}{t_k}\vec{r}_{k-1} - 2\frac{t_{k-1}}{t_k}\frac{b+a}{b-a}\vec{x}_{k-1} - \frac{t_{k-2}}{t_k}\vec{x}_{k-2} + \left(2\frac{t_{k-1}}{t_k}\frac{b+a}{b-a} + \frac{t_{k-2}}{t_k}\right)\vec{x}^*. \end{aligned}$$

Wykorzystując wynikającą z formuły trójczłonowej równość

$$t_k = -2 \frac{b-a}{b+a} t_{k-1} - t_{k-2}$$

otrzymujemy ostatecznie

$$\vec{x}_k = (1 + \alpha_k)\vec{x}_{k-1} - \alpha_k\vec{x}_{k-2} - \beta_k\vec{r}_{k-1},$$

gdzie $\alpha_k = \frac{t_{k-2}}{t_k}$ oraz $\beta_k = \frac{4t_{k-1}}{t_k(b+a)}$. Algorytm wynikający z metody Czebyszewa można więc zapisać następująco.

```

 $\vec{x}_0 := \{\text{dowolne}\}; \quad \vec{r}_0 := \vec{b} - A\vec{x}_0;$ 
 $\vec{x}_1 := \vec{x}_0 + 2\vec{r}_0/(b+a); \quad \vec{r}_1 := \vec{b} - A\vec{x}_1;$ 
 $t_0 := 1; \quad t_1 := -(b+a)/(b-a);$ 
 $\beta := -4/(b-a);$ 
for  $j = 2, 3, \dots$  do
begin
 $t_k := 2t_1t_{k-1} - t_{k-2};$ 
 $\alpha_k := t_{k-2}/t_k;$ 
 $\vec{x}_k := (1 + \alpha_k)\vec{x}_{k-1} - \alpha_k\vec{x}_{k-2} + \beta(t_{k-1}/t_k)\vec{r}_{k-1};$ 
 $\vec{r}_k := \vec{b} - A\vec{x}_k$ 
end.

```

Zauważmy, że opisana metoda Czebyszewa jest *metodą dwupunktową*, bowiem do konstrukcji kolejnego przybliżenia \vec{x}_k wykorzystuje się dwa poprzednie przybliżenia \vec{x}_{k-1} i \vec{x}_{k-2} .

Zastanowimy się jeszcze nad szybkością zbieżności metody Czebyszewa. Wobec

$$\|W_k(A)\|_2 = \frac{\|T_k(h(\cdot))\|_{C([a,b])}}{|T_k(h(0))|} = \frac{1}{T_k\left(\frac{b+a}{b-a}\right)}$$

mamy

$$\|\vec{x}_k - \vec{x}^*\|_2 \leq \frac{\|\vec{x}_0 - \vec{x}^*\|}{T_k\left(\frac{b+a}{b-a}\right)}.$$

Aby oszacować $T_k^{-1}\left(\frac{b+a}{b-a}\right)$, wykorzystamy jawną formułę na $T_k(t)$ z Ćw. 7.4. Oznaczając $t = \frac{b+a}{b-a}$ mamy dla dużych k

$$\begin{aligned} \frac{1}{T_k(t)} &= \frac{2}{(t + \sqrt{t^2 - 1})^k + (t - \sqrt{t^2 - 1})^k} \approx \frac{2}{(t + \sqrt{t^2 - 1})^k} \\ &= \frac{2}{\left(\frac{b-a}{b+a} + \sqrt{\frac{(b+a)^2 - (b-a)^2}{(b-a)^2}}\right)^k} = \frac{2(b-a)^k}{(\sqrt{b} + \sqrt{a})^{2k}} = 2 \left(\frac{\sqrt{b} - \sqrt{a}}{\sqrt{b} + \sqrt{a}}\right)^k, \end{aligned}$$

a stąd w przybliżeniu

$$\|\vec{x}_k - \vec{x}^*\|_2 \leq 2 \left(\frac{\sqrt{\frac{b}{a}} - 1}{\sqrt{\frac{b}{a}} + 1}\right)^k \|\vec{x}_0 - \vec{x}^*\|_2.$$

Metoda Czebyszewa jest więc zbieżna co najmniej liniowo z ilorazem $(\sqrt{\frac{b}{a}} - 1) / (\sqrt{\frac{b}{a}} + 1)$. Zauważmy jeszcze, że jeśli a i b są odpowiednio najmniejszą i największą wartością własną macierzy A , to $b = \|A\|_2$, $1/a = \|A^{-1}\|_2$, a stąd $b/a = \|A\|_2 \|A^{-1}\|_2 = \text{cond}(A)$ i metoda jest zbieżna z ilorazem

$$\frac{\sqrt{\text{cond}(A)} - 1}{\sqrt{\text{cond}(A)} + 1}.$$

Po raz kolejny widzimy tu ważną rolę uwarunkowania macierzy. Im uwarunkowanie jest gorsze tym gorsza zbieżność metody Czebyszewa.

11.3 Metoda najszybszego spadku

Tak jak poprzednio, będziemy zakładać, że macierz A układu jest symetryczna i dodatnio określona. Wtedy zachodzi następujący fakt.

Lemat 11.3 *Rozwiązanie \vec{x}^* nieosobliwego układu równań liniowych $A\vec{x} = \vec{b}$ jest jedynym wektorem minimalizującym formę kwadratową*

$$Q(\vec{x}) = \frac{1}{2} \vec{x}^T A \vec{x} - \vec{x}^T \vec{b}.$$

Dowód. Zauważmy, że dla dowolnych \vec{x} i \vec{z} mamy

$$\begin{aligned} Q(\vec{x} + \vec{z}) &= \frac{1}{2} (\vec{x}^T + \vec{z}^T) A (\vec{x} + \vec{z}) - (\vec{x}^T + \vec{z}^T) \vec{b} \\ &= \frac{1}{2} \vec{x}^T A \vec{x} - \vec{x}^T \vec{b} + \frac{1}{2} (\vec{z}^T A \vec{x} + \vec{x}^T A \vec{z}) - \vec{z}^T \vec{b} + \frac{1}{2} \vec{z}^T A \vec{z} \\ &= Q(\vec{x}) - \vec{z}^T (\vec{b} - A\vec{x}) + \frac{1}{2} \vec{z}^T A \vec{z}, \end{aligned}$$

przy czym w ostatniej równości wykorzystaliśmy symetrię macierzy A . Stąd dla $\vec{x} = \vec{x}^*$ i $\vec{z} \neq \vec{0}$ dostajemy

$$Q(\vec{x}^* + \vec{z}) = Q(\vec{x}^*) + \frac{1}{2} \vec{z}^T A \vec{z} > Q(\vec{x}^*),$$

co z kolei wynika z dodatniej określoności A . \square

W metodzie najszybszego spadku konstruuje się kolejne przybliżenie \vec{x}_k przesuwając się od \vec{x}_{k-1} w kierunku przeciwnym do gradientu formy Q tak długo jak długo wartość tej formy maleje. (Przypomnijmy, że minus gradient wyznacza kierunek lokalnie najszybszego spadku.) Ponieważ gradient Q w punkcie \vec{x}_{k-1} wynosi

$$\left(\frac{\partial Q}{\partial x_i}(\vec{x}_{k-1}) \right)_{i=1}^n = A\vec{x}_{k-1} - \vec{b} = -\vec{r}_{k-1},$$

metoda ma postać

$$\vec{x}_k = \vec{x}_{k-1} + \gamma_k \vec{r}_{k-1}, \tag{11.5}$$

gdzie γ_k jest dobrane tak, aby zminimalizować $Q(\vec{x}_k)$. Jak łatwo się przekonać,

$$\begin{aligned} Q(\vec{x}_k) &= Q(\vec{x}_{k-1} + \gamma \vec{r}_{k-1}) \\ &= Q(\vec{x}_{k-1}) + \frac{1}{2} \gamma^2 \vec{r}_{k-1}^T A \vec{r}_{k-1} + \gamma \vec{r}_{k-1}^T A \vec{x}_{k-1} - \gamma \vec{r}_{k-1}^T \vec{b} \\ &= Q(\vec{x}_{k-1}) + \frac{1}{2} \gamma^2 \vec{r}_{k-1}^T A \vec{r}_{k-1} - \gamma \vec{r}_{k-1}^T \vec{r}_{k-1} \end{aligned}$$

jest jako funkcja γ parabolą, która przyjmuje minimum w

$$\gamma_k = \frac{\vec{r}_{k-1}^T \vec{r}_{k-1}}{\vec{r}_{k-1}^T A \vec{r}_{k-1}}. \quad (11.6)$$

Wzory (11.5) i (11.6) definiują już w pełni metodę najszybszego spadku.

Pokażemy jeszcze zbieżność metody. W tym celu zauważmy, że

$$Q(\vec{x}_k) = Q(\vec{x}_{k-1}) - \frac{1}{2} \frac{(\vec{r}_{k-1}^T \vec{r}_{k-1})^2}{\vec{r}_{k-1}^T A \vec{r}_{k-1}} \leq Q(\vec{x}_{k-1}).$$

Ciąg $Q(\vec{x}_k)$ jest nierosnący i ograniczony z dołu przez $Q(\vec{x}^*)$, a więc jest zbieżny. To wymusza

$$\lim_{k \rightarrow \infty} \frac{(\vec{r}_{k-1}^T \vec{r}_{k-1})^2}{\vec{r}_{k-1}^T A \vec{r}_{k-1}} = 0,$$

co z kolei oznacza, że $\vec{r}_k = \vec{b} - A\vec{x}_k$ zbiega do zera i w konsekwencji

$$\lim_{k \rightarrow \infty} \vec{x}_k = \vec{x}^*.$$

Uwagi i uzupełnienia

U. 11.1 (*Dowód Lematu 11.2*) Jeśli $(\vec{\xi}_j)_{j=1}^n$ jest ortonormalną bazą wektorów własnych macierzy A , to dla dowolnego wektora $\vec{x} = \sum_{j=1}^n \alpha_j \vec{\xi}_j$ mamy

$$\begin{aligned} \|A\vec{x}\|_2^2 &= \langle A\vec{x}, A\vec{x} \rangle_2 = \left\langle \sum_{i=1}^n \alpha_i A(\vec{\xi}_i), \sum_{j=1}^n \alpha_j A(\vec{\xi}_j) \right\rangle_2 \\ &= \left\langle \sum_{i=1}^n \alpha_i \lambda_i \vec{\xi}_i, \sum_{j=1}^n \alpha_j \lambda_j \vec{\xi}_j \right\rangle_2 = \sum_{j=1}^n \alpha_j^2 \lambda_j^2 \leq \lambda_1^2 \sum_{j=1}^n \alpha_j^2 = \lambda_1^2 \|\vec{x}\|_2^2, \end{aligned}$$

a stąd $\|A\vec{x}\|_2 \leq |\lambda_1| \|\vec{x}\|_2$. Z drugiej strony,

$$\|A\vec{\xi}_1\|_2 = \|\lambda_1 \vec{\xi}_1\|_2 = |\lambda_1| \|\vec{\xi}_1\|_2 = |\lambda_1|,$$

a więc $\|A\|_2 = |\lambda_1|$.

Aby pokazać drugą część lematu zauważmy, że macierz A^{-1} jest też symetryczna. Ma ona te same wektory własne co A , a wartości własne wynoszą $1/\lambda_j$, przy czym $|1/\lambda_n| \geq \dots \geq |1/\lambda_1|$. Stąd $\|A^{-1}\|_2 = |1/\lambda_n|$.

U. 11.2 Pokazaliśmy, że warunkiem dostatecznym zbieżności metody iteracji prostej $\vec{x}_k = B\vec{x}_{k-1} + \vec{c}$ jest $\|B\| < 1$. Okazuje się, że można podać warunek konieczny i dostateczny. Metoda iteracji prostej jest zbieżna dla dowolnego przybliżenia początkowego \vec{x}_0 wtedy i tylko wtedy, gdy promień spektralny macierzy B jest nie większy od jedności, tzn.

$$\rho(A) = \max |\lambda| < 1,$$

gdzie maksimum wzięte jest po wszystkich wartościach własnych macierzy A .

U. 11.3 Przedstawiając macierz A w postaci $A = L + U$, gdzie L jest macierzą trójkątną dolną elementów występujących na i pod główną przekątną A , układ równań $A\vec{x} = \vec{b}$ jest równoważny układowi $L\vec{x} = -U\vec{x} + \vec{b}$. Stąd wynika metoda iteracji prostej

$$\vec{x}_k = -L^{-1}U\vec{x}_{k-1} + L^{-1}\vec{b},$$

zwana *metodą Seidla*. Można pokazać, że metoda ta jest zbieżna gdy macierz A jest symetryczna i dodatnio określona.

U. 11.4 Pokazana metoda Czebyszewa dla macierzy $A = A^T > 0$ o wartościach własnych z przedziału $[a, b]$, $0 < a < b < \infty$, jest metodą dwupunktową. Rozpatruje się też jednopunktowe, cykliczne metody Czebyszewa postaci

$$\vec{x}_k = \vec{x}_{k-1} + \alpha_k \vec{r}_{k-1},$$

gdzie parametry α_k są wybierane cyklicznie, $\alpha_{j+m} = \alpha_j$ dla wszystkich j i pewnej ustalonej m , oraz tak, aby zminimalizować normę macierzy przejścia w każdym cyklu. Dokładniej, mamy $\vec{x}_k - \vec{x}^* = (I - \alpha_k A)(\vec{x}_{k-1} - \vec{x}^*)$, a stąd

$$\vec{x}_m - \vec{x}^* = W_m(A)(\vec{x}_0 - \vec{x}^*),$$

gdzie

$$W_m(A) = (I - \alpha_m A)(I - \alpha_{m-1} A) \cdots (I - \alpha_1 A).$$

Wielomian $W_m(t)$ jest więc stopnia m , a jego pierwiastki wynoszą $t_j = 1/\alpha_j$, $1 \leq j \leq m$. Ponadto $W_m(0) = 1$. Jak już wiemy, norma macierzy $\|W_m(A)\|_2$ jest minimalna gdy $W_m(t) = W_m^*(t) = T_m(h(t))/T_m(h(0))$, gdzie $h(t) = (2x - (a+b))/(b-a)$. Stąd parametry α_{j+m} najlepiej jest wybrać tak, aby α_j były odwrotnościami pierwiastków $W_m^*(t)$, czyli

$$\alpha_j = \left(\frac{b+a}{2} - \frac{b-a}{2} \cos\left(\frac{2j-1}{2j}\pi\right) \right)^{-1}, \quad 1 \leq j \leq m.$$

U. 11.5 Metoda najszybszego spadku należy do rodziny *metod gradientowych* postaci

$$\vec{x}_k = \vec{x}_{k-1} + \alpha_k \vec{r}_{k-1},$$

gdzie parametr α_k wybiera się tak, aby zminimalizować normę

$$\|\vec{x}_k - \vec{x}^*\|_B = \sqrt{(\vec{x}_k - \vec{x}^*)^T B (\vec{x}_k - \vec{x}^*)},$$

a B jest pewną macierzą, $B = B^T > 0$. Zwykle przyjmuje się $B = A^p$. Jak łatwo się przekonać, metodę najszybszego spadku otrzymujemy w przypadku $p = 1$, czyli $B = A$.

Można pokazać, że dla metod gradientowych

$$\|\vec{x}_k - \vec{x}^*\|_B \leq \left(\frac{\text{cond}(A) - 1}{\text{cond}(A) + 1} \right)^k \|\vec{x}_0 - \vec{x}^*\|_B.$$

Ćwiczenia

Ćw. 11.1 Niech A będzie macierzą symetryczną o wartościach własnych λ_j , $1 \leq j \leq n$. Pokaż, że dla dowolnego wielomianu $W(t) = \sum_{j=0}^k a_j t^j$, macierz

$$W(A) = \sum_{j=1}^n a_j A^j$$

jest też symetryczna. Ponadto, wektory własne A i $W(A)$ są takie same, a wartości własne $W(A)$ wynoszą $W(\lambda_j)$ dla $1 \leq j \leq n$.

Ćw. 11.2 Niech $A = A^T > 0$. Rozpatrzmy *metodę Richardsona* postaci

$$\vec{x}_k = (I - \alpha A)\vec{x}_{k-1} + \alpha \vec{b}.$$

gdzie α dobieramy tak, aby zminimalizować normę macierzy $\|I - \alpha A\|_2$. Zauważ, że jest to metoda iteracji prostej, ale równocześnie jednopunktowa metoda Czebyszewa z U. 11.4 przy $m = 1$. Pokaż, że optymalne α wyraża się wzorem

$$\alpha_{opt} = \frac{2}{\lambda_1 + \lambda_n},$$

gdzie λ_1 i λ_n są odpowiednio największą i najmniejszą wartością własną A . Ponadto,

$$\|I - \alpha_{opt} A\|_2 = \frac{\lambda_1 - \lambda_n}{\lambda_1 + \lambda_n} = \frac{\text{cond}(A) - 1}{\text{cond}(A) + 1},$$

gdzie $\text{cond}(A) = \|A\|_2 \|A^{-1}\|_2$.

Ćw. 11.3 Pokaż, że warunkiem koniecznym i dostatecznym zbieżności metody Richardsona z Ćw. 11.2, dla dowolnego przybliżenia początkowego \vec{x}_0 , jest

$$0 < \alpha < \frac{2}{\|A\|_2}. \quad (11.7)$$

Ponadto, dla α spełniającego (11.7), iloraz zbieżności wynosi

$$\|I - \alpha A\|_2 = \begin{cases} \alpha \lambda_1 - 1, & \alpha_{opt} < 2/\lambda_1, \\ 1 - \alpha \lambda_n, & 0 < \alpha \leq \alpha_{opt}, \end{cases}$$

gdzie $\alpha_{opt} = \frac{2}{\lambda_1 + \lambda_n}$.

Ćw. 11.4 Wykaż, że m -cykliczna jednopunktowa metoda Czebyszewa z U. 11.4 jest zbieżna liniowo z ilorazem równym $\frac{b-a}{b+a}$ dla $m = 1$ i dążącym do $\frac{\sqrt{b}-\sqrt{a}}{\sqrt{b}+\sqrt{a}}$, gdy $m \rightarrow \infty$.

Ćw. 11.5 Niech $A = A^T > 0$ i niech

$$Q(\vec{x}) = \frac{1}{2} \vec{x}^T A \vec{x} + \vec{x}^T \vec{b}.$$

Pokaż, że zbiór

$$\{\vec{x} \in \mathbb{R}^n : Q(\vec{x}) = \text{const.}\}$$

tworzy hiperelipsoidę w \mathbb{R}^n o środku $A^{-1}\vec{b}$ i osiach w kierunkach wektorów własnych macierzy A o długościach proporcjonalnych do $\lambda_j^{-1/2}$, gdzie λ_j są wartościami własnymi A .

Wskazówka. Wykorzystaj Lemat 11.1.

Rozdział 12

Iteracje dla równań nieliniowych

W Rozdziale 11 analizowaliśmy metody iteracyjne dla rozwiązywania równań liniowych $A\vec{x} = \vec{b}$. Teraz zajmiemy się równaniami *nieliniowymi*, w których zamiast operatora liniowego (macierzy) A mamy odwzorowanie nieliniowe $f : D \rightarrow \mathbb{R}^n$, gdzie D jest podzbiorem w \mathbb{R}^n . Dla uproszczenia, będziemy zakładać, że $\vec{b} = \vec{0}$, gdyż w przeciwnym przypadku możemy zastąpić f funkcją $f_1(\vec{x}) = f(\vec{x}) - \vec{b}$. Będziemy również zakładać, że f należy do pewnej znanej klasy F funkcji przekształcających D w \mathbb{R}^n i mających co najmniej jedno zero w D . Zadanie polega więc na znalezieniu rozwiązania $\vec{x}^* = \vec{x}^*(f)$ równania nieliniowego

$$f(\vec{x}) = \vec{0}, \quad \text{dla} \quad f \in F. \quad (12.1)$$

Jak zwykle w przypadku, gdy danymi są funkcje zakładamy, że jedyną informacją “a priori” o f jest, że $f \in F$. Dodatkową informację możemy zdobyć przez obliczanie wartości, a niekiedy i pochodnych f (o ile istnieją) w wybranych punktach. Zauważmy, że metody iteracyjne Czebyszewa i najszybszego spadku z Rozdziału 11 dla równań liniowych korzystały właśnie z takiej (częściowej) informacji o macierzy A , chociaż formalnie zakładaliśmy, że dostępna jest pełna informacja o jej współczynnikach. Dla równań nieliniowych, możliwość obliczenia jedynie informacji częściowej jest założeniem modelowym, a to wymusza już fakt, że każda metoda rozwiązywania równania (12.1) może dawać co najwyżej wynik przybliżony.

12.1 Bisekcja

Metoda bisekcji, czyli *połowienia*, jest dość naturalną metodą obliczania zer skalarnych funkcji ciągłych określonych na danym przedziale $[a, b]$ i zmieniających znak. Dokładniej, rozpatrzmy klasę funkcji

$$F = \{ f \in C([a, b]) : f(a) < 0 < f(b) \}. \quad (12.2)$$

Oczywiście, każda funkcja $f \in F$ ma co najmniej jedno zero w $[a, b]$. Startując z przedziału $[a, b]$, w kolejnych krokach metody bisekcji obliczamy informację o wartości f w środku przedziału, co pozwala nam, w zależności od znaku obliczonej wartości, zmniejszyć o połowę przedział, w którym na pewno znajduje się zero funkcji. Bisekcję realizuje następujący ciąg poleceń, po wykonaniu którego x jest przybliżeniem zera funkcji f z zadaną dokładnością ε .

```
xl := a;   xr := b;
x := (a + b)/2;   e := (b - a)/2;
while (e > ε) do
```



```

begin
  if (f(x) < 0) then xr := x else xl := x;
  x = (xl + xr)/2; e := e/2
end.

```

Z konstrukcji metody łatwo wynika, że po wykonaniu k iteracji (albo po obliczeniu k wartości funkcji) otrzymujemy $x = x_k$, które odległe jest od pewnego rozwiązania x^* o co najwyżej

$$|x_k - x^*| \leq \left(\frac{1}{2}\right)^k \left(\frac{b-a}{2}\right). \quad (12.3)$$

Metoda bisekcji jest więc zbieżna *liniowo* z ilorazem $1/2$. Choć ta zbieżność nie jest imponująca, bisekcja ma kilka istotnych zalet. Oprócz jej prostoty, należy podkreślić fakt, że bisekcja jest w pewnym sensie uniwersalna. Dla jej zbieżności wystarcza bowiem jedynie ciągłość funkcji. Poza tym możemy łatwo kontrolować jej błąd. Konsekwencją (12.3) jest bowiem następujący wniosek.

Wniosek 12.1 Dla znalezienia zera z dokładnością $\varepsilon > 0$, wystarczy obliczyć w metodzie bisekcji

$$k = k(\varepsilon) = \left\lceil \log_2 \frac{(b-a)}{\varepsilon} \right\rceil - 1$$

wartości funkcji.

Dodajmy jeszcze, że bisekcja minimalizuje błąd najgorszy w klasie F zdefiniowanej przez (12.2), wśród wszystkich algorytmów korzystających z określonej liczby obliczeń wartości funkcji, cf. U. 12.1.

12.2 Iteracje proste

Przedstawimy teraz metodę *iteracji prostych* dla rozwiązywania równań nieliniowych, którą można traktować jako uogólnienie iteracji prostych dla równań liniowych. Najpierw równanie (12.1) przekształcamy do równania równoważnego (tzn. mającego te same rozwiązania)

$$\vec{x} = \phi(\vec{x}). \quad (12.4)$$

Następnie, startując z pewnego przybliżenia początkowego \vec{x}_0 , konstruujemy ciąg kolejnych przybliżeń \vec{x}_k według wzoru

$$\vec{x}_k = \phi(\vec{x}_{k-1}), \quad k \geq 1.$$

Twierdzenie 12.1 Niech D_0 będzie domkniętym podzbiorem dziedziny D ,

$$\overline{D_0} = D_0 \subset D,$$

w którym ϕ jest odwzorowaniem zwężającym. To znaczy, $\phi(D_0) \subset D_0$, oraz istnieje stała $0 \leq L < 1$ taka, że

$$\|\phi(\vec{x}) - \phi(\vec{y})\| \leq L \|\vec{x} - \vec{y}\| \quad \text{dla } \vec{x}, \vec{y} \in D_0.$$

Wtedy równanie (12.4) ma dokładnie jedno rozwiązanie \vec{x}^* , oraz

$$\vec{x}^* = \lim_{k \rightarrow \infty} \vec{x}_k,$$

dla dowolnego przybliżenia początkowego $\vec{x}_0 \in D_0$.

Dowód. Wobec

$$\|\vec{x}_k - \vec{x}_{k-1}\| = \|\phi(\vec{x}_{k-1}) - \phi(\vec{x}_{k-2})\| \leq L \|\vec{x}_{k-1} - \vec{x}_{k-2}\| \leq \dots \leq L^{k-1} \|\vec{x}_1 - \vec{x}_0\|,$$

dla $k \geq s$ mamy

$$\begin{aligned} \|\vec{x}_k - \vec{x}_s\| &\leq \sum_{j=s+1}^k \|\vec{x}_j - \vec{x}_{j-1}\| \leq \sum_{j=s+1}^k L^{j-1} \|\vec{x}_1 - \vec{x}_0\| \\ &= L^s (1 + L + \dots + L^{k-s-1}) \|\vec{x}_1 - \vec{x}_0\| \leq \frac{L^s}{1-L} \|\vec{x}_1 - \vec{x}_0\|. \end{aligned}$$

Ciąg $\{\vec{x}_k\}_k$ jest więc ciągiem Cauchy'ego. Stąd istnieje granica $\vec{\alpha} = \lim_{k \rightarrow \infty} \vec{x}_k$, która należy do D_0 , wobec domkniętości tego zbioru. Ponieważ Lipschitzowskość ϕ implikuje jej ciągłość, mamy też

$$\phi(\vec{\alpha}) = \phi\left(\lim_{k \rightarrow \infty} \vec{x}_k\right) = \lim_{k \rightarrow \infty} \phi(\vec{x}_k) = \lim_{k \rightarrow \infty} \vec{x}_k = \vec{\alpha},$$

tzn. $\vec{\alpha}$ jest punktem stałym odwzorowania ϕ . Dla jednoznaczności zauważmy, że jeśliby istniał drugi, różny od $\vec{\alpha}$, punkt stały $\vec{\beta}$, to mielibyśmy

$$\|\vec{\alpha} - \vec{\beta}\| = \|\phi(\vec{\alpha}) - \phi(\vec{\beta})\| \leq L \|\vec{\alpha} - \vec{\beta}\|.$$

Stąd $1 \leq L$, co byłoby sprzeczne z założeniem, że ϕ jest zwężająca. \square

Z powyższych rozważań otrzymujemy natychmiastowy wniosek dotyczący zbieżności iteracji prostych.

Wniosek 12.2 *Przy założeniach Twierdzenia 12.1, metoda iteracji prostych jest zbieżna co najmniej liniowo z ilorazem L , tzn.*

$$\|\vec{x}_k - \vec{x}^*\| \leq L^k \|\vec{x}_0 - \vec{x}^*\|.$$

Dla ilustracji, rozpatrzmy natępujące proste równanie skalarne:

$$x = \cos(x) \quad \text{dla} \quad x \in D = \mathbb{R}. \quad (12.5)$$

W tym przypadku $\phi(x) = \cos(x)$. Zauważamy, że w przedziale $[0, 1]$ funkcja ϕ jest zwężająca ze stałą

$$L = \max_{0 \leq x \leq 1} |\cos'(x)| = \sin(1) < 1.$$

Stąd istnieje dokładnie jedno rozwiązanie równania (12.5) w przedziale $[0, 1]$. Rozwiązanie to może być aproksymowane z dowolnie małym błędem przy pomocy iteracji prostych, startując z dowolnego przybliżenia początkowego $\vec{x}_0 \in [0, 1]$.

Zaletą iteracji prostych jest fakt, że zbieżność nie zależy od wymiaru n zadania, ale tylko od stałej Lipschitza L . Ma ona szczególne zastosowanie w przypadku, gdy funkcja ϕ jest zwężająca na całym zbiorze D , tzn. $D_0 = D$. Jeśli ponadto D ma skończoną średnicę $\text{diam}(D)$, to dla osiągnięcia ε -sproksymacji zera funkcji f wystarczy wykonać

$$k = k(\varepsilon) = \left\lceil \frac{\log(\|\vec{x}_0 - \vec{x}^*\|/\varepsilon)}{\log(1/L)} \right\rceil \leq \left\lceil \frac{\log(\text{diam}(D)/\varepsilon)}{\log(1/L)} \right\rceil$$

iteracji, niezależnie od x_0 . Metody zbieżne dla dowolnego przybliżenia początkowego, nazywamy *zbieżnymi globalnie*.

12.3 Metody interpolacyjne

Metody interpolacyjne dla równań nieliniowych powstają podobnie jak kwadratury interpolacyjne dla całkowania. Najpierw daną funkcję interpolujemy wielomianem Lagrange'a albo Hermite'a ustalonego stopnia, a następnie jako przybliżenie zera bierzemy zero tego wielomianu interpolacyjnego. Proces taki można iterować (powtarzać) wymieniając za każdym razem jeden z punktów interpolacyjnych na punkt nowo obliczony. Zauważmy, że w ogólności takie metody muszą korzystać w kolejnych iteracjach nie tylko z ostatniego przybliżenia x_{k-1} , ale również z kilku poprzednich, tzn.

$$x_k = \phi(x_{k-1}, \dots, x_{k-s}).$$

Wymagają one również podania nie jednego, ale s przybliżeń początkowych x_0, \dots, x_{s-1} .

W praktyce stosuje się jednak tylko interpolację wielomianami niskich stopni, np. wielomianami liniowymi lub najwyżej kwadratowymi, których zera można łatwo obliczyć korzystając z bezpośrednich formuł.

Jak się przekonamy, metody interpolacyjne należą do grupy metod *zbieżnych lokalnie*. Znaczy to, że zbieżność ciągu $\{x_k\}_k$ do zera danej funkcji f jest zapewniona jedynie wtedy, gdy przybliżenia początkowe zostały wybrane dostatecznie blisko x^* .

W dalszych rozważaniach będziemy zakładać dla uproszczenia, że dziedzina $D = \mathbb{R}$.

W przypadku, gdy wielomianem interpolacyjnym jest wielomian liniowy Hermite'a, mówimy o *metodzie Newtona*, albo *stycznych*. Startując z pewnego przybliżenia początkowego x_0 , w kolejnych krokach metody, k -te przybliżenie x_k jest punktem przecięcia stycznej do wykresu f w punkcie x_{k-1} . Ponieważ równanie stycznej wynosi $y(x) = f(x_{k-1}) + f'(x_{k-1})(x - x_{k-1})$, otrzymujemy wzór

$$x_k = x_{k-1} - \frac{f(x_{k-1})}{f'(x_{k-1})}.$$

Oczywiście, aby metoda Newtona była dobrze zdefiniowana, musimy założyć, że $f'(x_{k-1})$ istnieje i nie jest zerem.

Zauważmy, że metodę Newtona można traktować jako szczególny przypadek iteracji prostych, gdzie

$$\phi(x) = x - \frac{f(x)}{f'(x)}.$$

Widać też, że nie jest ona zbieżna globalnie. Nawet jeśli pochodna w x_{k-1} się nie zeruje, ciąg $\{x_k\}_k$ może nie zbiegać do zera funkcji f , cf. Ćw. 12.2. Okazuje się jednak, że jeśli wystartujemy dostatecznie blisko rozwiązania x^* , to metoda Newtona jest zbieżna. Dokładniej, załóżmy najpierw, że $f(x^*) = 0$ oraz

$$f'(x^*) \neq 0.$$

Ponadto załóżmy, że f jest dwukrotnie różniczkowalna w sposób ciągły, $f \in C^2(D)$. Rozwijając ϕ w szereg Taylora w punkcie x^* otrzymujemy

$$x_k - x^* = \phi(x_{k-1}) - \phi(x^*) = (x_{k-1} - x^*)\phi'(x^*) + \frac{1}{2}(x_{k-1} - x^*)^2\phi''(\xi_k),$$

gdzie $\min(x^*, x_{k-1}) \leq \xi_k \leq \max(x^*, x_{k-1})$. Wobec tego, że

$$\phi'(x^*) = \frac{f(x)f''(x)}{(f'(x))^2} = 0 \quad \text{i} \quad \phi''(\xi_k) = \frac{f''(\xi_k)}{f'(\xi_k)},$$

mamy

$$x_k - x^* = (x_{k-1} - x^*)^2 \frac{f''(\xi_k)}{2f'(\xi_k)}. \quad (12.6)$$

Zdefiniujmy liczbę

$$R_f = \sup \left\{ r \geq 0 : \sup_{\{x: |x-x^*| \leq r\}} \left| \frac{2(x-x^*)f''(x)}{f'(x)} \right| < 1 \right\}.$$

Oczywiście $R_f > 0$. Dla x_0 spełniającego $|x_0 - x^*| \leq R < R_f$ mamy zbieżność ciągu $\{x_k\}_{k \geq 0}$ do x^* , bowiem z równości (12.6) wynika, iż wtedy

$$|x_k - x^*| \leq q^k |x_0 - x^*|,$$

gdzie $q < 1$ i q zależy od R .

Niech teraz x^* będzie zerem m -krotnym,

$$f(x^*) = f'(x^*) = \dots = f^{(m-1)}(x^*) = 0 \neq f^{(m)}(x^*),$$

gdzie $m \geq 2$, oraz niech f będzie m -krotnie różniczkowalna w sposób ciągły. Wtedy

$$\begin{aligned} x_k - x^* &= (x_{k-1} - x^*) - \frac{(x_{k-1} - x^*)^m f^{(m)}(\eta_k^{(1)})/m!}{(x_{k-1} - x^*)^{m-1} f^{(m-1)}(\eta_k^{(2)})/(m-1)!} \\ &= (x_{k-1} - x^*) \left(1 - \frac{1}{m} \frac{f^{(m)}(\eta_k^{(1)})}{f^{(m)}(\eta_k^{(2)})} \right) \\ &\approx (x_{k-1} - x^*) \left(1 - \frac{1}{m} \right), \end{aligned} \quad (12.7)$$

o ile x_{k-1} jest “blisko” x^* .

Metoda Newtona jest więc zbieżna lokalnie. Z (12.6) i (12.7) można też wywnioskować, jaki jest charakter zbieżności metody Newtona. Dla zera jednokrotnego x^* oraz $f''(x^*) \neq 0$ mamy bowiem

$$(x_k - x^*) \approx (x - x_{k-1})^2 \frac{f''(x^*)}{2f'(x^*)}.$$

Mówimy, że zbieżność jest *kwadratowa*. Jeśli zaś $f''(x^*) = 0$ to zbieżność jest nawet szybsza. Z kolei dla zera m -krotnego zbieżność jest liniowa z ilorazem $(1 - 1/m)$.

Metoda Newtona jest pierwszą poznaną tutaj metodą iteracyjną, która jest (dla zer jednokrotnych) zbieżna szybciej niż liniowo. Dla takich metod wprowadza się pojęcie *wykładnika zbieżności*, który jest zdefiniowany następująco.

Powiemy, że metoda iteracyjna ϕ jest w klasie funkcji F rzędu co najmniej $p \geq 1$, gdy spełniony jest następujący warunek. Niech $f \in F$ i $f(x^*) = 0$. Wtedy istnieje $C < +\infty$ taka, że dla dowolnych przybliżeń początkowych x_0, \dots, x_{s-1} dostatecznie bliskich x^* , kolejne przybliżenia $x_k = \phi(x_{k-1}, \dots, x_{k-s})$ generowane tą metodą spełniają

$$|x_k - x^*| \leq C |x_{k-1} - x^*|^p.$$

Ponadto, jeśli $p = 1$ to dodatkowo żąda się, aby $C < 1$.

Definicja 12.1 Wykładnikiem zbieżności metody iteracyjnej ϕ w klasie F nazywamy liczbę p^* zdefiniowaną równością

$$p^* = \sup \{ p \geq 1 : \phi \text{ jest rzędu co najmniej } p \}.$$

Możemy teraz sformułować następujące twierdzenie, które natychmiast wynika z poprzednich rozważań.

Twierdzenie 12.2 Wykładnik zbieżności metody Newtona (stycznych) wynosi $p^* = 2$ w klasie funkcji o zerach jednokrotnych, oraz $p^* = 1$ w klasie funkcji o zerach wielokrotnych.

Inną znaną i często używaną metodą interpolacyjną jest *metoda siecznych*, w której stosuje się liniowy wielomian interpolacyjny Lagrange'a. Metoda ta wykorzystuje więc do konstrukcji x_k przybliżenia x_{k-1} i x_{k-2} . Musimy również wybrać dwa różne punkty startowe x_0 i x_1 . Ponieważ prosta interpolująca f w x_{k-1} i x_{k-2} ma wzór

$$y(x) = \frac{x - x_{k-2}}{x_{k-1} - x_{k-2}} f(x_{k-1}) + \frac{x - x_{k-1}}{x_{k-2} - x_{k-1}} f(x_{k-2}),$$

otrzymujemy

$$x_k = x_{k-1} - \frac{x_{k-1} - x_{k-2}}{f(x_{k-1}) - f(x_{k-2})} f(x_{k-1}).$$

Zauważmy, że jeśli x_{k-1} i x_{k-2} są blisko siebie, to x_k jest podobny do tego z metody Newtona, bowiem wtedy iloraz różnicowy $(f(x_{k-1}) - f(x_{k-2})) / (x_{k-1} - x_{k-2})$ dobrze przybliża pochodną $f'(x_{k-1})$. Ale to nie wystarcza, aby osiągnąć zbieżność z wykładnikiem 2. Dokładniej, można pokazać, że wykładnik zbieżności metody siecznych dla zer jednokrotnych wynosi $p^* = \frac{1}{2}(1 + \sqrt{5}) = 1.618\dots$, cf. U. 12.3.

Niewątpliwą zaletą metody siecznych jest jednak to, że nie wymaga ona obliczania pochodnej funkcji (co w praktyce jest często bardzo trudne, a niekiedy nawet niemożliwe), a tylko jej wartości.

12.4 Obliczanie zer wielomianów

Obliczanie zer w przypadku, gdy funkcja f jest wielomianem jest właściwie osobnym zadaniem. My tylko dotknijemy tego tematu.

Założmy, że wielomian $w : \mathbb{R} \rightarrow \mathbb{R}$ jest dany poprzez jego współczynniki a_j w bazie potęgowej,

$$w(x) = \sum_{j=0}^n a_j x^j$$

($a_n \neq 0$). Zastosujmy metodę Newtona dla znalezienia największego zera tego wielomianu. Jedna iteracja metody Newtona jest wtedy stosunkowo prosta, gdyż obliczenie wartości i pochodnej wielomianu w kolejnym punkcie $\xi = x_{k-1}$ polega po prostu na dwukrotnym zastosowaniu schematu Hornera. Rzeczywiście, przypomnijmy, że schemat Hornera oblicza nie tylko wartość $w(\xi)$, ale również współczynniki w bazie potęgowej ilorazu $w_1(x)$ przy dzieleniu $w(x)$ przez jednomian $(x - \xi)$, cf. Ćw. 6.2. Zauważmy dalej, że równość

$$w(x) = (x - \xi)w_1(x) + w(\xi)$$

implikuje $w'(\xi) = w_1(\xi)$. Stąd schemat obliczania jednocześnie $w(\xi)$ i $w'(\xi)$ wygląda następująco.

```

w0 := w1 := a_n;
for j := n - 1 downto 1 do
begin
  w0 := w0 * ξ + a_j;
  w1 := w1 * ξ + w0
end;
w0 := w0 + a_0.

```

Po wykonaniu tych poleceń mamy $w_0 = w(\xi)$ i $w_1 = w'(\xi)$.

Twierdzenie 12.3 *Załóżmy, że wielomian $w(x) = \sum_{j=0}^n a_j x^j$ ma zera rzeczywiste*

$$\xi_1 \geq \xi_2 \geq \dots \geq \xi_n.$$

Jeśli tylko przybliżenie początkowe $x_0 > \xi_1$, to ciąg x_k generowany metodą Newtona jest ściśle malejący i zbieżny do ξ_1 .

Dowód. Wielomian w można przedstawić w postaci

$$w(x) = (x - \xi_1)w_1(x),$$

gdzie $w_1(x) = a_n(x - \xi_2) \cdots (x - \xi_n)$. Wtedy

$$w'(x) = w_1(x) + (x - \xi_1)w_1'(x),$$

a stąd

$$\begin{aligned} x_k - \xi_1 &= (x_{k-1} - \xi_1) - \frac{w(x_{k-1})}{w'(x_{k-1})} \\ &= (x_{k-1} - \xi_1) \left(1 - \frac{w_1(x_{k-1})}{w_1(x_{k-1}) + (x_{k-1} - \xi_1)w_1'(x_{k-1})} \right). \end{aligned} \quad (12.8)$$

Zauważmy teraz, że wielomiany $w_1(x)$ i $w_1'(x)$ mają stały znak dla $x > \xi_1$, który równy jest znakowi a_n . Rzeczywiście, ponieważ $w_1(x)$ ma wszystkie $n - 1$ pierwiastków w przedziale $(-\infty, \xi_1]$, to $w_1'(x)$ ma również wszystkie $n - 2$ pierwiastki w tym samym przedziale. Stąd oba wielomiany mają ten sam znak na $(\xi_1, +\infty)$ równy znakowi współczynnika przy najwyższych potęgach tych wielomianów. Współczynnik ten jest w obu przypadkach równy a_n .

Z rozważań tych wynika, że mnożnik przy $(x_{k-1} - \xi_1)$ w prawej stronie równania (12.8) jest dla $x_{k-1} > \xi_1$ dodatni i mniejszy od 1, a stąd, że ciąg x_k jest malejący i ograniczony z dołu przez ξ_1 . Niech $g = \lim_{k \rightarrow \infty} x_k \geq \xi_1$. Biorąc granicę obu stron (12.8) przy $k \rightarrow \infty$ otrzymujemy

$$g - \xi_1 = (g - \xi_1) \left(1 - \frac{w_1(g)}{w_1(g) + (g - \xi_1)w_1'(g)} \right),$$

co wymusza $g = \xi_1$, kończąc dowód. \square

Dodajmy, że jeśli zero ξ_1 jest jednokrotne, $\xi_1 > \xi_2$, to mamy zbieżność kwadratową. Należy jednak podkreślić, że zbieżność kwadratową uzyskujemy dopiero pod koniec procesu iteracyjnego. Mamy bowiem

$$\lim_{x \rightarrow +\infty} \frac{w_1(x)}{w_1(x) + (x - \xi_1)w_1'(x)} = \lim_{x \rightarrow +\infty} \left(1 + \frac{xw_1'(x)}{w_1(x)} \right)^{-1} = \frac{1}{n}.$$

Jeśli więc wystartujemy z $x_0 \gg \xi_1$ to początkowo zaobserwujemy zbieżność tylko liniową z ilorazem $1 - 1/n$. W miarę zbliżania się do ξ_1 zbieżność będzie rosła, aż osiągnie w końcu poziom zbieżności kwadratowej,

$$x_k - \xi_1 \approx (x_{k-1} - \xi_1)^2 \frac{w'_1(\xi_1)}{w_1(\xi_1)},$$

Dla zera wielokrotnego, $\xi_1 = \dots = \xi_m > \xi_{m+1}$, gdzie $m < n$, zbieżność pozostanie jednak cały czas na poziomie liniowym, przy czym iloraz zbieżności wzrośnie do $1 - 1/m$. W końcu, jeśli ξ_1 jest zerem o maksymalnej krotności n , to łatwo zauważyć, że

$$x_k - \xi_1 = \left(1 - \frac{1}{n}\right)(x_{k-1} - \xi_1).$$

Pozostaje jeszcze problem znalezienia przybliżenia $x_0 > \xi_1$. Można to zrobić na kilka sposobów. Na przykład, zauważmy, że dla

$$x > M = \max\left(1, \sum_{j=0}^{n-1} \frac{|a_j|}{|a_n|}\right)$$

mamy

$$\frac{w(x)}{a_n} \geq x^n \left(1 - \sum_{j=0}^{n-1} \frac{|a_j|}{|a_n x^{n-j}|}\right) \geq x^n \left(1 - \frac{1}{M} \sum_{j=0}^{n-1} \frac{|a_j|}{|a_n|}\right) > 0.$$

Stąd $w(x)$ ma stały znak na $(M, +\infty)$, a to oznacza, że $\xi_1 \leq M$. Jako x_0 można więc wziąć dowolną liczbę większą od M .

Uwagi i uzupełnienia

U. 12.1 Metoda bisekcji jest optymalna w następującym sensie. Niech $A : F \rightarrow \mathbb{R}$ będzie dowolną metodą (algorytmem) aproksymującą zero $x^*(f)$ funkcji f z klasy F zdefiniowanej w (12.2), korzystającą jedynie z obliczeń (informacji o) f w k punktach. Wtedy dla dowolnego $\gamma > 0$ istnieje funkcja $f_\gamma \in F$ mająca tylko jedno zero $x^*(f_\gamma)$ w $[a, b]$ i taka, że

$$|A(f_\gamma) - x^*(f_\gamma)| \geq \left(\frac{1}{2}\right)^k \left(\frac{b-a}{2}\right) + \gamma.$$

Co więcej, można pokazać, że fakt ten zachodzi też w węższej klasie F_1 funkcji $f \in F$, które są dowolnie wiele razy różniczkowalne. Oczywiście, nie wyklucza to istnienia metod iteracyjnych (takich jak metoda Newtona), które dla $f \in F_1$ są zbieżne szybciej niż liniowo.

U. 12.2 Wszystkie metody iteracyjne opisane w tym rozdziale należą do grupy *metod adaptacyjnych*. Przypomnijmy, że są to metody, które obliczają kolejną wartość (ew. pochodną) funkcji f w punkcie x_k , którego wybór istotnie zależy od poprzednio obliczonych wartości f , tzn. $x_k = x_k(f(x_0), \dots, f(x_{k-1}))$. (Wcześniej, w U. 9.3 poznaliśmy adaptacyjne metody dla zadania całkowania.) Okazuje się, że metody nieadaptacyjne są dużo gorsze przy rozwiązywaniu równań nieliniowych. Na przykład, najlepsza metoda korzystająca z k wartości funkcji wybranych nieadaptacyjnie ma w klasie (12.2) błąd najgorszy równy $(b-a)/(2k+2)$.

U. 12.3 Uzasadnimy teraz, że dla zer jednokrotnych wykładnik zbieżności metody siecznych wynosi

$$p^* = \frac{1 + \sqrt{5}}{2}.$$

Niech $f \in C^2(D)$. Wykorzystując ilorazy różnicowe otrzymujemy

$$\begin{aligned} x_k - x^* &= (x_{k-1} - x^*) - \frac{f(x_{k-1})}{f[x_{k-2}, x_{k-1}]} \\ &= (x_{k-1} - x^*) \left(1 - \frac{f[x_{k-1}, x^*]}{f[x_{k-2}, x_{k-1}]} \right) \\ &= (x_{k-1} - x^*)(x_{k-2} - x^*) \frac{f[x_{k-2}, x_{k-1}, x^*]}{f[x_{k-2}, x_{k-1}]} \\ &= (x_{k-1} - x^*)(x_{k-2} - x^*) \frac{f''(\xi_k^{(1)})}{2f'(\xi_k^{(2)})}. \end{aligned}$$

Stąd widać, że dla x_{k-2} i x_{k-1} dostatecznie bliskich x^* , ciąg x_k zbiega do x^* . Aby znaleźć wykładnik zbieżności, założmy bez zmniejszenia ogólności, że $f''(x^*) \neq 0$ (w przeciwnym przypadku zbieżność jest jeszcze szybsza). Wtedy

$$x_k - x^* \approx C(x_{k-1} - x^*)(x_{k-2} - x^*), \quad (12.9)$$

gdzie $C = f''(x^*)/(2f'(x^*))$. Stąd, po podstawieniu $z_j = \ln(|C(x_j - x^*)|)$, dostajemy

$$z_k \approx z_{k-1} + z_{k-2}.$$

Ostatnie równanie jest *równaniem różnicowym*, którego rozwiązaniem jest $z_k \approx q^k$, gdzie $q = \frac{1}{2}(1 + \sqrt{5})$ jest dominującym pierwiastkiem równania $z^2 = z + 1$. Ostatecznie więc $|x_k - x^*| \approx |C^{-1}e^{q^k}|$, czyli

$$|x_k - x^*| \approx |C|^{q-1} \left(|C^{-1}e^{q^{k-1}} \right)^q \approx |C|^{q-1} |x_{k-1} - x^*|^q.$$

U. 12.4 Dla znalezienia zera funkcji $f: \mathbb{R}^n \rightarrow \mathbb{R}^n$ można zastosować wielowymiarową metodę Newtona. Jest ona zdefiniowana następującym wzorem rekurencyjnym:

$$\vec{x}_k = \vec{x}_{k-1} - (f'(\vec{x}_{k-1}))^{-1} f(\vec{x}_{k-1}),$$

gdzie $f'(\vec{\xi})$ jest macierzą-Jacobianem funkcji f wziętym w punkcie $\vec{\xi}$, tzn.

$$f'(\vec{\xi}) = \begin{pmatrix} \frac{\partial f_1}{\partial x_1}(\vec{\xi}) & \cdots & \frac{\partial f_1}{\partial x_n}(\vec{\xi}) \\ \vdots & & \vdots \\ \frac{\partial f_n}{\partial x_1}(\vec{\xi}) & \cdots & \frac{\partial f_n}{\partial x_n}(\vec{\xi}) \end{pmatrix}, \quad f(\vec{\xi}) = \begin{pmatrix} f_1(\vec{\xi}) \\ \vdots \\ f_n(\vec{\xi}) \end{pmatrix}, \quad \vec{\xi} = \begin{pmatrix} \xi_1 \\ \vdots \\ \xi_n \end{pmatrix}.$$

W praktyce metodę tą realizuje się rozwiązując w każdym kroku iteracyjnym układ równań liniowych $n \times n$. Dokładniej,

$$\vec{x}_k = \vec{x}_{k-1} - \vec{y}_{k-1},$$

gdzie \vec{y}_{k-1} jest rozwiązaniem układu

$$f'(\vec{x}_{k-1})\vec{y}_{k-1} = f(\vec{x}_{k-1}).$$

Tak jak w przypadku jednowymiarowym, wielowymiarowa metoda Newtona jest zbieżna lokalnie do zera \vec{x}^* z wykładnikiem 2, o ile Jacobian w punkcie \vec{x}^* jest nieosobliwy, tzn. $\det(f'(\vec{x}^*)) \neq 0$.

Ćwiczenia

Ćw. 12.1 Uzasadnij, że metoda iteracji prostych $x_k = \cos(x_{k-1})$ jest zbieżna do jedyne go rozwiązania równania $x = \cos(x)$, dla dowolnego przybliżenia początkowego $x_0 \in \mathbb{R}$.

Ćw. 12.2 Podaj przykład funkcji f i przybliżenia początkowego x_0 takich, że ciąg generowany metodą Newtona oscyluje, tzn. $x_{2s} = x_0$ i $x_{2s+1} = x_1$, dla wszystkich s .

Ćw. 12.3 Jeśli w danym modelu obliczeniowym liczenie pierwiastka kwadratowego \sqrt{a} dla $a > 0$ nie jest dopuszczalne, to możemy go przybliżyć stosując wzór rekurencyjny

$$x_k = \frac{1}{2} \left(x_{k-1} + \frac{a}{x_{k-1}} \right),$$

który powstaje przez zastosowanie metody Newtona do równania $f(x) = x^2 - a = 0$. Pokaż, że ciąg x_k jest zbieżny do \sqrt{a} dla dowolnego przybliżenia początkowego $x_0 > 0$. Scharakteryzuj szybkość zbieżności.

Ćw. 12.4 Zaproponuj metodę iteracyjną obliczania $1/a$ dla $a > 0$ nie używającą dzielenia. Jak wybrać przybliżenie początkowe aby metoda była zbieżna? Jaki jest wykładnik zbieżności?

Wskazówka. Zastosuj metodę Newtona do funkcji $f(x) = 1/x - a$.

Ćw. 12.5 Rozpatrzmy metodę iteracyjną

$$x_k = x_{k-1} - \frac{x_{k-1} - a}{f(x_{k-1}) - f(a)} f(x_{k-1}),$$

gdzie a jest pewnym parametrem. Pokaż, że metoda ta jest zbieżna liniowo, o ile a jest dostatecznie blisko zera x^* .

Ćw. 12.6 Pokaż, że metoda Steffensena,

$$x_k = x_{k-1} - \frac{f^2(x_{k-1})}{f(x_{k-1} + f(x_{k-1})) - f(x_{k-1})},$$

jest zbieżna lokalnie z wykładnikiem 2.

Ćw. 12.7 Rozpatrzmy metodę iteracyjną daną wzorem

$$x_k = x_{k-1} - m \frac{f(x_{k-1})}{f'(x_{k-1})}.$$

Pokaż, że jeśli f ma m -krotne zero x^* to metoda ta jest lokalnie zbieżna do x^* z wykładnikiem 2.