Piotr Krzyżanowski

Ćwiczenia zdalne

2020-03-11

Spis treści

1. Polecana literatura

Zajęcia dotyczą rozdziałów 2 i 3 (mniej więcej strony 50. . . 65) skryptu Wykładowcy, P.Kiciaka. Ponadto każdy szanujący się podręcznik z metod numerycznych ma rozdziały o arytmetyce zmiennopozycyjnej (*floating point*), numerycznej poprawności (*backward stability*) algorytmów i o uwarunkowaniu (*conditioning*) zadań obliczeniowych.

Aby się dobrze nastroić przed rozpoczęciem pracy, warto przeczytać bardzo frapujące historyjki z *Lectures 6, 7, 8* w znanej już Państwu książeczce G.W. Stewarta *Afternotes on Numerical Analysis*.

2. Zadania z rozwiązaniami

Namawiam Państwa do próby samodzielnego rozwiązania każdego z zadań, a dopiero potem do przeczytania gotowca.

Będę używał następujących skrótów:

— pisząc o *arytmetyce fl* będę miał na myśli arytmetykę zmiennopozycyjną („*fl*”, bo „*floating point*”)
— precyzję arytmetyki fl będę oznaczać literką ν; jak już wiecie Państwo z wykładu, dla pojedynczej precyzji, $\nu \approx 10^{-7}$; w przypadku podwójnej precyzji, $\nu \approx 10^{-16}$.

— Wykładowca używa oznaczenia $rd(x)$ dla operacji *reprezentacji* liczby x w arytmetyce fl. (Czyli $rd(x)$ jest liczbą maszynową najbliższą x.) Natomiast wynik obliczenia działania $x \diamond y$ w arytmetyce fl oznacza przez $fl(x \diamond y)$. Ja będę tu pisał $fl(x)$ i $fl(x \diamond y)$ w obu przypadkach — nie prowadzi to do nieporozumień.

O ile nie będzie powiedziane inaczej, zawsze będziemy zakładać, że wszystkie *dane*, a także *wyniki* obliczeń (pośrednie i końcowe) dają się reprezentować jako znormalizowane liczby maszynowe (więc na żadnym etapie obliczeń w fl nie wystąpi ani zjawisko nadmiaru, ani niedomiaru, ani stopniowego niedomiaru). Ponadto będziemy też zakładać, że wyniki działań w fl są zaokrąglane do najbliższej liczby maszynowej.

2.1. Przystawki

0. *Najpierw przeczytaj polecaną literaturę!*

1. Wykaż, że każde z podstawowych działań arytmetycznych jest algorytmem numerycznie poprawnym.

Rozw. Niech $x, y \in \mathbb{R}$. Wtedy ich reprezentacja w arytmetyce fl jest postaci

$$\tilde{x} := fl(x) = x \cdot (1+\varepsilon_x), \qquad \tilde{y} := fl(y) = y \cdot (1+\varepsilon_y),$$

gdzie $|\varepsilon_x|, |\varepsilon_y| \leq \nu$.

Nasze „algorytmy" polegają na wykonaniu (aż!) *jednego* działania arytmetycznego. Sprawdzimy numeryczną poprawność „algorytmów" dodawania: $s = x+y$ i mnożenia: $p = x \cdot y$. (Pozostałe działania sprawdza się podobnie.)

Ze względu na błąd reprezentacji, nasze „algorytmy" będą pracowały nie na oryginalnych x, y, tylko na ich reprezentacjach $\tilde{x}, \tilde{y}$. Obliczony wynik też zostanie zaokrąglony do najbliższej liczby maszynowej. Więc jako wyniki dostaniemy odpowiednio

$$\tilde{s} := fl(\tilde{x}+\tilde{y}) = (\tilde{x}+\tilde{y})(1+\varepsilon_s), \qquad \tilde{p} := fl(\tilde{x} \cdot \tilde{y}) = (\tilde{x} \cdot \tilde{y})(1+\varepsilon_p),$$

gdzie $|\varepsilon_x|, |\varepsilon_y| \leq \nu$.

Aby wykazać numeryczną poprawność tych algorytmów musimy pokazać, że wynik ich działania w fl daje się zinterpretować jako *dokładny* wynik na lekko zaburzonych (tzn. zaburzonych na poziomie nieuniknionego błędu reprezentacji) danych. Dla dodawania mamy:

$$\tilde{s} = (\tilde{x}+\tilde{y})(1+\varepsilon_s) = x \cdot \underbrace{(1+\varepsilon_x)(1+\varepsilon_s)}_{=:1+E_x} + y \cdot \underbrace{(1+\varepsilon_y)(1+\varepsilon_s)}_{=:1+E_y} = x \cdot (1+E_x) + y \cdot (1+E_y) = \hat{x}+\hat{y}.$$

Pokazaliśmy więc, że obliczony w fl rezultat to *dokładny* wynik zsumowania dwóch liczb, $\hat{x}$ i $\hat{y}$. Pozostaje pokazać, że $\hat{x}$ i $\hat{y}$ to są *lekko zaburzone* prawdziwe wartości x i y.

Ponieważ $\hat{x} = x \cdot (1+E_x)$, znaczy to, że

$$\frac{\hat{x}-x}{x} = E_x,$$

więc wystarczy oszacować $|E_x|$ przez precyzję arytmetyki ν pomnożoną przez „niewielką" stałą. Mamy, pomijając na koniec człony rzędu ν^2 (zob. też zadanie 9),

$$1+E_x = (1+\varepsilon_x)(1+\varepsilon_s) = 1+\varepsilon_x+\varepsilon_s+\underbrace{\varepsilon_x\varepsilon_s}_{\text{rzędu } \nu^2} \approx 1+\varepsilon_x+\varepsilon_s,$$

skąd $E_x = \varepsilon_x + \varepsilon_s$ (z dokładnością do pominiętych wyrazów rzędu ν^2) i w konsekwencji

$$|E_x| \leq |\varepsilon_x| + |\varepsilon_s| \leq 2\nu.$$

Analogicznie pokazujemy $|E_y| \leq 2\nu$, zatem rzeczywiście pozorne zaburzenia danych: E_x, E_y są na poziomie precyzji arytmetyki. Algorytm jest więc numerycznie poprawny.

Dla mnożenia mamy niemal identycznie

$$\tilde{p} = \hat{x} \cdot \hat{y},$$

gdzie tym razem (na przykład, bo mamy tu pewną dowolność)

$$\hat{x} = x(1+\varepsilon_x)(1+\varepsilon_p) =: x(1+E_x), \qquad \hat{y} = y(1+\varepsilon_y),$$

no i pozorne zaburzenia E_x, ε_y są na poziomie 2ν i ν odpowiednio — skąd znów numeryczna poprawność. □

2. Niech x, y będą liczbami rzeczywistymi. Wykaż, że wyniki obliczenia $x \cdot y$ oraz x/y w arytmetyce fl są obarczone małym błędem względnym. (Porównaj to zadanie z następnym!)

Rozw. Na wejściu algorytmu zamiast x, y dysponujemy ich reprezentacjami w arytmetyce fl:

$$\tilde{x} := fl(x) = x \cdot (1+\varepsilon_x), \qquad \tilde{y} := fl(y) = y \cdot (1+\varepsilon_y),$$

gdzie $|\varepsilon_x|, |\varepsilon_y| \leq \nu$.

Rachunki przeprowadzimy dla dzielenia x/y. (Dla mnożenia $x \cdot y$ jest jeszcze łatwiej.)

Pokażemy, że obliczony w fl wynik dzielenia, $\tilde{q}$, jest dokładnym wynikiem $q := x/y$, zaburzonym na (względnym) poziomie precyzji arytmetyki, tzn. pokażemy, że

$$\tilde{q} = q \cdot (1+\delta),$$

gdzie δ będzie na poziomie ν.

Istotnie, dla pewnego $|\varepsilon_q| \leq \nu$ mamy

$$\tilde{q} = fl(\tilde{x}/\tilde{y}) = \frac{\tilde{x}}{\tilde{y}} \cdot (1+\varepsilon_q),$$

skąd, podstawiając za $\tilde{x}$ i $\tilde{y}$,

$$\tilde{q} = \frac{x}{y} \cdot \underbrace{\frac{(1+\varepsilon_x)}{1+\varepsilon_y}(1+\varepsilon_q)}_{=:1+E_q} = q \cdot (1+E_q).$$

Aby więc oszacować błąd względny

$$\frac{\tilde{q}-q}{q} = E_q,$$

wystarczy oszacować $|E_q|$. Ponieważ — pomijając na koniec składniki wyższego rzędu względem ν (por. zadanie 10) — mamy

$$1+E_q = (1+\varepsilon_x)(1+\varepsilon_q)\frac{1}{1+\varepsilon_y} = (1+\varepsilon_x+\varepsilon_q+\varepsilon_x\varepsilon_q)(1-\varepsilon_y+\varepsilon_y^2-\ldots) \approx 1+\varepsilon_x+\varepsilon_q-\varepsilon_y,$$

to

$$|E_q| \leq |\varepsilon_x| + |\varepsilon_q| + |\varepsilon_y| \leq 3\nu.$$

Zatem faktycznie, błąd względny nigdy nie przekracza 3ν.

Jak zmieni się odpowiedź, gdy dodatkowo wiemy, że x, y są liczbami maszynowymi? □

3. Niech x, y będą liczbami rzeczywistymi. Wykaż, że jeśli $x \approx y$, to wynik obliczenia $x - y$ w fl może być obarczony bardzo dużym błędem względnym. (Porównaj to zadanie z poprzednim! Dodatkowo, przyjrzyj się zadaniom z LABu 2.)

Uwaga. To bardzo ważne zjawisko, noszące potoczną nazwę redukcji cyfr przy odejmowaniu*, pokazuje, że w arytmetyce fl wystarczy* jedno *działanie na specyficznej konfiguracji danych, by całkowicie zdewastować jakość wyniku. Nie trzeba czekać, aż błędy zaokrągleń „skumulują się" po wielkiej liczbie obliczeń.*

Rozw. Bez zmniejszenia, a nawet zwiększając ogólność, rozważmy dowolne $x, y \in \mathbb{R}$ i zadanie obliczenia $s = x + y$. Z zadania 1 mamy, że

$$\tilde{s} = x(1 + E_x) + y(1 + E_y),$$

więc

$$\left|\frac{\tilde{s} - s}{s}\right| = \frac{|\overbrace{x+y}^{=s} + xE_x + yE_y - s|}{|s|} \leq \frac{|x| \cdot |E_x| + |y| \cdot |E_y|}{|x+y|} \leq \frac{|x| + |y|}{|x+y|} \cdot 2\nu,$$

bo w zadaniu 1 pokazano, że $|E_x|, |E_y| \leq 2\nu$.

Z tego oszacowania płyną dwa wnioski: optymistyczny i pesymistyczny.

Optymistyczny: Jeśli x i y są *tego samego* znaku, to

$$\frac{|\tilde{s} - s|}{|s|} \leq 2\nu.$$

Mamy więc wtedy *gwarancję* małego błędu względnego.

Pesymistyczny: (I to jest rozwiązanie postawionego zadania.) Jeśli $x \approx -y$, to

$$\frac{|x| + |y|}{|x+y|} \approx +\infty,$$

więc wtedy (potencjalnie) także błąd względny $\dfrac{|\tilde{s} - s|}{|s|}$ może być bardzo duży. Zgodnie z prawem Murphy'ego zapewne więc *będzie* bardzo duży *naprawdę*. Można to sprawdzić, przeprowadzając eksperyment na komputerze — patrz zestaw ćwiczeń na LAB 2.

□

2.2. Danie główne

4. Niech a i b będą liczbami maszynowymi. Jak obliczać $a^2 - b^2$ w fl?

Rozw. Wygląda na to, że dla $|a| \approx |b|$ grozi nam redukcja cyfr przy odejmowaniu. Rozważmy dwa algorytmy:

```
% Algorytm 1
x = a*a; y = b*b;
w = x-y;
```

oraz

```
% Algorytm 2
s = a+b; r = a−b;
w = s∗r;
```

Oba mają ten sam koszt (3 flopy), a jak będą zachowywać się w fl? Ponieważ (i to jest tutaj kluczowe) założyliśmy, że a i b są liczbami maszynowymi, więc — z definicji — są *dokładnie* reprezentowane w fl. Wynikiem kolejnych kroków działania Algorytmu 1 w fl są więc:

1. $\tilde{x} = fl(a*a) = a^2(1+\varepsilon_1)$, gdzie $|\varepsilon_1| \leq \nu$,
2. $\tilde{y} = fl(b*b) = b^2(1+\varepsilon_2)$, gdzie $|\varepsilon_2| \leq \nu$,
3. $\tilde{w} = fl(\tilde{x}-\tilde{y})$.

Korzystając z rozwiązania zadania 3 wnioskujemy więc, że

$$\frac{|\tilde{w}-w|}{|w|} = fl(\tilde{x}-\tilde{y}) \leq \frac{|a^2|+|b^2|}{|a^2-b^2|}\cdot 2\nu,$$

więc faktycznie dla $|a| \approx |b|$ jest ryzyko utraty cyfr przy odejmowaniu.

Dla Algorytmu 2 zrealizowanego w fl mamy analogicznie

1. $\tilde{s} = fl(a+b) = (a+b)(1+\varepsilon_3)$, gdzie $|\varepsilon_3| \leq \nu$,
2. $\tilde{r} = fl(a-b) = (a-b)(1+\varepsilon_4)$, gdzie $|\varepsilon_4| \leq \nu$,
3. $\tilde{w} = fl(\tilde{s}*\tilde{r}) = \tilde{s}\tilde{r}(1+\varepsilon_5)$, gdzie $|\varepsilon_5| \leq \nu$.

Zatem w tym przypadku

$$\tilde{w} = \tilde{s}\tilde{r}(1+\varepsilon_5) = (a+b)(a-b)(1+\varepsilon_3)(1+\varepsilon_4)(1+\varepsilon_5) = (a^2-b^2)\cdot(1+E),$$

gdzie $|E| \leq 3\nu$ (dlaczego?). A to oznacza, że tym razem błąd względny wyniku nie przekracza 3ν — czyli wynik *zawsze* jest doskonałej jakości!

Zachęcam do zastanowienia się, dlaczego nie stoi to w sprzeczności z wynikiem o redukcji cyfr przy odejmowaniu. □

5. Oblicz wskaźnik uwarunkowania wartości a^2-b^2

— ze względu na zaburzenie a,
— ze względu na zaburzenie b.

Kiedy to zadanie jest źle uwarunkowane?

Rozw. Policzymy uwarunkowanie ze względu na zaburzenie a; dla b postępowanie jest całkowicie analogiczne. Definiując

$$f(a) = a^2-b^2,$$

mamy (por. wykład), że uwarunkowanie obliczania wartości f w punkcie a wynosi

$$\mathrm{cond}(f,a) = \frac{|a\cdot f'(a)|}{|f(a)|} = \frac{2a^2}{|a^2-b^2|} = \frac{2}{|1-(b/a)^2|}.$$

Zadanie jest źle uwarunkowane, gdy $\mathrm{cond}(f,x) \gg 1$. W naszym przypadku tak jest, gdy $|b| \approx |a|$, bo wtedy mianownik ≈ 0. □

6. Niech $x \approx 0$. Przekształć poniższe wyrażenia tak, by uniknąć redukcji cyfr przy odejmowaniu:

1. $\sqrt{x+1}-1$,
2. $(1-\cos x)/\sin x$,

Rozw. Stosujemy uniwersalną zasadę:

Jak zadanie jest za trudne — to zmień zadanie!

1. $\sqrt{x+1}-1 = \dfrac{x}{\sqrt{x+1}+1}$ i kłopotliwe odejmowanie znika.
2. Jak w poprzednim punkcie, sprytnie rozszerzamy ułamek i dostajemy

$$\frac{1-\cos x}{\sin x} = \frac{1-\cos x}{\sin x}\cdot\frac{1+\cos x}{1+\cos x} = \frac{1-\cos^2 x}{\sin x}\cdot\frac{1}{1+\cos x} = \frac{\sin x}{1+\cos x}$$

i dla $x \approx 0$ suma w mianowniku jest niegroźna.

□

7. Wykaż, że standardowy algorytm obliczenia sumy n liczb rzeczywistych,

$$s = x_1 + x_2 + \ldots + x_n$$

```
s = x_1;
for i = 2:n
  s = s + x_i;
return s;
```

jest numerycznie poprawny.

Rozw. Jeśli przeczytałeś historyjki z książeczki Stewarta, to pewnie zorientowałeś się, że wszystkie szczegóły rozwiązania opisane są w *Lecture 7* od początku do *Backward stability* włącznie... *Backward stability* to po polsku właśnie numeryczna poprawność. □

8. Algorytm Hornera obliczania wartości wielomianu w punkcie. Ponieważ

$$p(x) = a_0 + a_1 x + \ldots a_N x^N = a_0 + x\underbrace{\left(a_1 + a_2 x + \ldots a_N x^{N-1}\right)}_{\text{wielomian stopnia } N-1},$$

to wyznaczenie wartości wielomianu stopnia N w zadanym punkcie x daje się sprowadzić do wyznaczenia wartości wielomianu stopnia o jeden niższego... Iterując ten pomysł, otrzymujemy tzw. **algorytm Hornera**:

```
p = a_N;
for k = N − 1 downto : 0
    p = p · x + a_k;
end
return p
```

1. Jaki jest koszt tego algorytmu w zależności od N?
2. Wykaż, że jest to algorytm numerycznie poprawny.
3. Pokaż, że jeśli współczynniki wielomianu oraz x są liczbami maszynowymi, to dla dostatecznie silnej arytmetyki błąd bezwzględny wyniku algorytmu spełnia oszacowanie

$$|fl(p(x)) - p(x)| \leq \left(\sum_{k=0}^{N} (2k+1) |a_k| |x|^k \right) \cdot \nu.$$

Rozw. Rozwiązanie jest nieco przydługawe, bo chciałem, żeby było w miarę kompletne. Proszę się nie zrażać, tylko najpierw spróbować zrobić to zadanie samodzielnie — przez analogię do zadania 7. W końcu i tu, i tam są i sumy, i liczenie w pętli...

Koszt. Skoro w każdym obrocie pętli wykonuje się dwa działania arytmetyczne, a pętla obróci się N razy, to w sumie wykonamy $2N$ flopów (czyli operacji arytmetycznych, ***fl****oating point* ***op****erations*.)

Pozostałe punkty robi się analogicznie, jak zadanie 7, ale na wszelki wypadek dokładnie prześledźmy kolejne kroki.

Numeryczna poprawność. Rachunki przeprowadzimy dla uogólnienia algorytmu Hornera dla przypadku tzw. bazy Newtona — przyda się to nam później, gdy zechcemy wprowadzić efekt działania arytmetyki fl.

Problem w wersji uogólnionej stawiamy następująco:

Dla danych $a_0, \ldots, a_N$ oraz $x_0, \ldots, x_N$, znaleźć wartość wielomianu zadanego w bazie Newtona[a]

$$p(x) = a_0 + a_1(x - x_0) + a_2(x - x_0)(x - x_1) + \ldots + a_N(x - x_0) \cdots (x - x_{N-1})$$

w danym punkcie x. Możemy powyższe skrótowo zapisać w postaci

$$p(x) = \sum_{j=0}^{N} a_j \prod_{s=0}^{j-1} (x - x_s).$$

[a] Baza Newtona dla przestrzeni wielomianów składa się z wielomianów coraz wyższego stopnia postaci: $1, \quad (x-x_0), \quad (x-x_0)(x-x_1), \quad (x-x_0)(x-x_1)(x-x_2), \ldots$ itd., gdzie $x_0, x_1, x_2 \ldots$ są ustalonymi węzłami.

I rzeczywiście, gdy $x_0 = \ldots = x_N = 0$, nasz problem redukuje się do postawionego oryginalnie w zadaniu, $p(x) = \sum_{j=0}^{N} a_j x^j$.

Uogólniony algorytm Hornera dla tego problemu — wyznaczenia wartości wielomianu zadanego w bazie Newtona — miałby następującą postać:

$p = a_N$;
for $k = N-1$ **downto** $: 0$
    $p = p \cdot (x - x_k) + a_k$;
end
return p

Na użytek analizy będzie nam wygodnie oznaczyć przez p_k wartość p w chwili, gdy licznik pętli jest równy k:

$p_N = a_N$;
for $k = N-1$ **downto** $: 0$
    $p_k = p_{k+1} \cdot (x - x_k) + a_k$;
end

return p_0

Najpierw sprawdzimy, że faktycznie ten algorytm wyznacza to, co powinien. Prześledźmy kilka pierwszych kroków:

$$\begin{aligned}
p_N &= a_N && = a_N,\\
p_{N-1} &= p_N\cdot(x-x_{N-1})+a_{N-1} && = a_N\cdot(x-x_{N-1})+a_{N-1},\\
p_{N-2} &= p_{N-1}\cdot(x-x_{N-2})+a_{N-2} && = a_N\cdot(x-x_{N-1})(x-x_{N-2})+a_{N-1}\cdot(x-x_{N-2})+a_{N-2},\\
&\text{...itd...}
\end{aligned}$$

Przyglądając się powyższym zależnościom możemy zauważyć, a następnie przez prostą indukcję udowodnić, że

$$p_k=\sum_{j=k}^{N} a_j\prod_{s=k}^{j-1}(x-x_s). \tag{1}$$

Stąd już wynika, że faktycznie $p_0=p(x)$ tak, jak tego chcieliśmy.

Teraz prześledzimy wpływ realizacji naszego algorytmu w fl na końcowy wynik. Bez zmniejszenia ogólności założymy, że liczby a_i, x_i oraz x są reprezentowane *dokładnie* w fl. (Dlaczego tak można?)

Zamiast dokładnych wartości p_k, będziemy więc raczej wyznaczać wartości zaburzone $\tilde{p}_k$:

$\tilde{p}_N = a_N$;
for $k = N-1$ **downto** $:0$
$\quad\tilde{p}_k = fl\Big(\tilde{p}_{k+1}\cdot(x-x_k)+a_k\Big)$;
end
return $\tilde{p}_0$

gdzie obliczenie $fl\Big(\tilde{p}_{k+1}\cdot(x-x_k)+a_k\Big)$ składa się z następujących małych kroczków:

1. $\tilde{h}_k = fl(x-x_k) = (x-x_k)\cdot(1+\varepsilon_k)$,
2. $\tilde{q}_k = fl(\tilde{p}_{k+1}\cdot\tilde{h}_k) = \tilde{p}_{k+1}\cdot\tilde{h}_k\cdot(1+\delta_k)$,
3. $\tilde{p}_k = fl(\tilde{q}_k+a_k) = (\tilde{q}_k+a_k)\cdot(1+\eta_k)$,

przy czym $|\varepsilon_k|,|\delta_k|,|\eta_k|\le\nu$. Zatem ostatecznie

$$\tilde{p}_k=\big(\tilde{p}_{k+1}\cdot(1+\varepsilon_k)(1+\delta_k)+a_k\big)\cdot(1+\eta_k)=\tilde{p}_{k+1}\cdot\overbrace{(x-x_k)\cdot(1+\mu_k)}+\underbrace{a_k\cdot(1+\eta_k)},$$

gdzie $1+\mu_k=(1+\varepsilon_k)(1+\delta_k)(1+\eta_k)$. Znaczy to, że realizacja w arytmetyce fl algorytmu Hornera z danymi $(x-x_k)$ oraz a_k jest równoważna wykonaniu algorytmu Hornera w arytmetyce *dokładnej*, w którym zamiast $(x-x_k)$ używamy $(x-x_k)\cdot(1+\mu_k)$, a zamiast a_k bierzemy $a_k\cdot(1+\eta_k)$.

Nas jednak interesuje, czy wynik ostateczny — to znaczy $\tilde{p}_0$ — da się zinterpretować jako dokładna wartość wielomianu na lekko zaburzonych danych. Do tego przyda się nam (1), które przecież zachodzi dla *dowolnych* danych. Dostaniemy zeń, że

$$\tilde{p}_k=\sum_{j=k}^{N}\underbrace{a_j(1+\eta_j)}\prod_{s=k}^{j-1}\overbrace{(x-x_s)(1+\mu_s)},\qquad k=N,\dots,0,$$

więc w szczególności

$$\tilde{p}_0 = \sum_{j=0}^{N} a_j(1+\eta_j)\prod_{s=0}^{j-1}(x-x_s)(1+\mu_s)$$
$$= \sum_{j=0}^{N} a_j \cdot \underbrace{(1+\eta_j)\prod_{s=0}^{j-1}(1+\mu_s)}_{=:1+E_j} \cdot \prod_{s=0}^{j-1}(x-x_s) = \sum_{j=0}^{N} \tilde{a}_j \cdot \prod_{s=0}^{j-1}(x-x_s),$$

gdzie $\tilde{a}_j = a_j(1+E_j)$.

Czyli pokazaliśmy, że faktycznie, wynik obliczony w fl da się zinterpretować jako *dokładny* wynik dla lekko zaburzonych danych $\tilde{a}_j$ i dokładnych x_j i x. (Uważny Czytelnik zwróci zapewne uwagę na to, że pozorne zaburzenia mogliśmy rozdzielić na inne sposoby pomiędzy a_j, x_j oraz x — jest tu pewna dowolność.)

Ale jak duże są pozorne zaburzenia E_j? Ponieważ

$$1+E_j = (1+\eta_j)\prod_{s=0}^{j-1}(1+\varepsilon_s)(1+\delta_s)(1+\eta_s),$$

to (pomijając człony rzędu ν^2 i wyższego — zob. zad. 9)

$$|E_j| \le (3j+1)\nu, \qquad j = 0,\ldots,N.$$

Przyjmując, że N nie jest patologicznie wielkie względem ν możemy uznać, że zaburzenie E_j pozostaje „na poziomie" ν.

Oszacowanie błędu. Po pierwsze zauważmy, że w przypadku zastosowania algorytmu Hornera w przypadku wielomianu zadanego w bazie naturalnej (to niego dotyczy zadanie), zaburzenia pozorne E_j, jakie wyprowadziliśmy powyżej, spełniają ciut lepsze oszacowanie, $|E_j| \le (2j+1)\nu$ (bo odpada odejmowanie). Dalej już łatwo:

$$|\tilde{p}_0 - p(x)| = |\sum_{j=0}^{N}(\tilde{a}_j - a_j)\cdot x^j| \le \sum_{j=0}^{N}|\tilde{a}_j - a_j|\cdot|x|^j = \sum_{j=0}^{N}|a_j E_j|\cdot|x|^j \le \sum_{j=0}^{N}(2j+1)|a_j|\cdot|x|^j\cdot\nu.$$

□

2.3. Deser — nieobowiązkowy!

To są zadania o tematyce nieco pobocznej, dla chętnych i zainteresowanych.

Pierwsze dwa zadania uzasadniają, że dla dostatecznie silnej arytmetyki, czynione przez nas przybliżenia $(1+\nu)^n \approx 1+n\nu$ i $(1+\nu)^{-n} \approx 1+n\nu$ jest rozsądne w sensie takim, że dla wszelkich praktycznie spotykanych n, prowadząc ścisły rachunek musielibyśmy tylko nieco zwiększyć stałą.

9. Wykaż, że jeśli $0 < nu < 1$, to

$$(1+u)^n \le 1+\gamma_n, \qquad \text{gdzie } \gamma_n := \frac{nu}{1-nu/2}.$$

Przykładowo, jeśli $u = 10^{-16}$, to dla $n \leq 10^{15}$ mamy $\gamma_n \leq 1.06 \cdot nu$.

10. Wykaż, że jeśli $0 < nu < 1$, to

$$\frac{1}{(1-u)^n} \leq 1 + \gamma_n, \qquad \text{gdzie } \gamma_n := \frac{nu}{1-nu}.$$

Przykładowo, jeśli $u = 10^{-16}$, to dla $n \leq 10^{15}$ mamy $\gamma_n \leq 1.12 \cdot nu$.

11. (Trudne) Niech x, y będą dodatnimi liczbami maszynowymi. Wykaż, że używając binarnej arytmetyki standardu IEEE-754 mamy gwarancję (o ile w obliczeniach nie wystąpi nadmiar ani niedomiar), iż obliczona w fl wartość $x/\sqrt{x^2+y^2}$ nie przekroczy 1. (To ważna własność, bo dokładny wynik możemy interpretować jako cosinus wiadomego kąta. Byłoby więc wskazane, żeby wynik w fl też dał się zinterpretować jako cosinus *czegoś*.)

3. Konsultacje

Będę online na czacie w najbliższą środę w standardowym terminie zajęć i potem dodatkową godzinę.

Można też zadawać pytania na czacie offline, najlepiej w grupie `mo-cwiczenia-grupa2`, abyśmy mogli uczyć się od siebie nawzajem i uniknęli powtarzających się pytań.

Oczywiście, można też użyć maila.

Matematyka obliczeniowa: ćw. zdalne 2020-03-18
Piotr Krzyżanowski

---

1. Polecana literatura

Zajęcia dotyczą mniej więcej stron 70…80 skryptu Wykładowcy, P. Kiciaka — czyli rozdziału 3 (*Przypomnienia* o normach) i rozdziału 4 (szczegóły wyprowadzenia numerycznej poprawności eliminacji Gaussa można pominąć). Ponadto każdy szanujący się podręcznik z metod numerycznych ma rozdziały o normach, w tym normach macierzowych, rozwiązywaniu układów równań liniowych i ich uwarunkowaniu.

Aby się dobrze nastroić przed rozpoczęciem pracy, warto przeczytać bardzo frapujące historyjki z *Lectures 10, 12* (wstawki o FORTRAN-ie można pominąć) oraz *Lectures 15, 16* w znanej już Państwu książeczce G.W. Stewarta *Afternotes on Numerical Analysis*. Jeśli ktoś potrzebuje, może też przejrzeć *Lecture 9* — takie sympatyczne przypomnienie z GAL-u.

Treści jest dużo i na pewno jeszcze do niej będziemy wracać.

2. Zadania z rozwiązaniami

Namawiam Państwa do próby samodzielnego rozwiązania każdego z zadań, a dopiero potem do przeczytania gotowca.

O ile nie będzie powiedziane inaczej, zawsze będziemy zakładać, że obliczenia są wykonywane w arytmetyce dokładnej.

W przypadku obliczeń w fl, będziemy jak zwykle przyjmować, że wszystkie *dane*, a także *wyniki* obliczeń (pośrednie i końcowe) dają się reprezentować jako znormalizowane liczby maszynowe (więc na żadnym etapie obliczeń w fl nie wystąpi ani zjawisko nadmiaru, ani niedomiaru, ani stopniowego niedomiaru). Ponadto będziemy też zakładać, że wyniki działań w fl są zaokrąglane do najbliższej liczby maszynowej.

2.1. Przystawki

0. *Najpierw przeczytaj polecaną literaturę!*

1. Wykaż, że dla dowolnego $x \in \mathbb{R}^N$,

$$\|x\|_\infty \leq \|x\|_2 \leq \|x\|_1 \leq N\|x\|_\infty.$$

Rozw. To prosta zabawa w szacowania. □

2. Wykaż, że dla dowolnej normy macierzowej indukowanej normą wektorową zachodzi

$$\|I\| = 1.$$

Rozw. Z definicji,

$$\|I\| = \sup_{x\neq 0} \frac{\|Ix\|}{\|x\|} = \sup_{x\neq 0} \frac{\|x\|}{\|x\|} = 1.$$

□

3. Jak zmienią się: wyznacznik i współczynnik uwarunkowania macierzy nieosobliwej A rozmiaru $N \times N$, jeśli pomnożymy ją przez stałą $\alpha > 0$?

Rozw. Oczywiście, $\det(\alpha A) = \alpha^N \det(A)$. Natomiast $\|\alpha A\| = \alpha \|A\|$ i analogicznie $\|(\alpha A)^{-1}\| = \alpha^{-1}\|A^{-1}\|$, skąd $\operatorname{cond}(\alpha A) = \operatorname{cond}(A)$.

Zatem: wyznacznik jest czuły na skalowanie macierzy, a współczynnik uwarunkowania — wcale. □

2.2. Danie główne

2.2.1. Normy macierzowe

Jak Państwo pamiętacie, definiując pojęcie numerycznej poprawności algorytmów odwoływaliśmy się do interpretacji rozwiązania na „lekko zaburzonych danych". Ale jeśli daną jest *macierz*, jak powinniśmy rozumieć małe zaburzenie macierzy? Jednym z pomysłów jest zmierzenie tego zaburzenia w normie — normie macierzowej.

Zapoznajmy się z tym pojęciem bliżej.

4. Podaj definicję, a następnie scharakteryzuj przez jawny wzór, normy macierzowe $\|\cdot\|_p$ dla $p = 1$ i dla $p = \infty$. Dlaczego norma $\|\cdot\|_1$ nazywa się normą *kolumnową*, a norma $\|\cdot\|_\infty$ — normą *wierszową*?

Rozw. Niech A będzie macierzą rzeczywistą $N \times M$. Dla uproszczenia, zamiast $\|\cdot\|_p$ będziemy pisali (dopóki nie będzie to prowadziło do nieporozumień) po prostu $\|\cdot\|$.

Z definicji,

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

(ostatnia równość wynika z tego, że $\dfrac{\|Ax\|}{\|x\|} = \|A \cdot \frac{x}{\|x\|}\|$). Również z definicji wynika wprost, że

$$\|Ax\| \leq \|A\| \cdot \|x\| \qquad \forall x \in \mathbb{R}^M.$$

Zatem: jeśli wskażemy pewną liczbę C taką, że $\|Ax\| \leq C \cdot \|x\|$ dla każdego x i dodatkowo wskażemy x^* takie, że $\|x^*\| = 1$ oraz jednocześnie $\|Ax^*\| = C \cdot \|x^*\|$ (tak, ma zachodzić równość!), to wtedy musi być, że $C = \|A\|$ (no, bo co to jest supremum?).

Zobaczmy, jak to pójdzie dla $\|A\|_1$ (czyli w treści zadania $p = 1$).

Musimy szacować normę $\|y\|_1 = \sum_i |y_i|$, gdzie $y = Ax$. Najpierw przyjrzyjmy się, czym jest y_i: oczywiście,

$$y_i = (Ax)_i = \sum_j A_{ij}x_j, \qquad \text{więc} \qquad |y_i| = |\sum_j A_{ij}x_j| \leq \sum_j |A_{ij}| \cdot |x_j|.$$

Stąd

$$\|Ax\|_1 = \sum_i |y_i| \le \sum_i \sum_j |A_{ij}| \cdot |x_j| = \sum_j \sum_i |A_{ij}| \cdot |x_j| = \sum_j |x_j| \sum_i |A_{ij}| \le \|x\|_1 \cdot \max_j \sum_i |A_{ij}|.$$

Czy więc udało się nam i faktycznie zachodzi $\|A\|_1 = \max_j \sum_i |A_{ij}|$? Aby tak było, musimy dodatkowo wskazać $\|x^*\|_1 = 1$ taki, że $\|Ax^*\|_1 = \max_j \sum_i |A_{ij}|$.

Niech k będzie indeksem tej kolumny, dla którego przyjmowane jest maksimum, tzn.

$$\max_j \sum_i |A_{ij}| = \sum_i |A_{ik}|.$$

Weźmy teraz za x^* taki wersor, który wyłuskuje k-tą kolumnę z A, czyli niech

$$x^* = (0, \ldots, 0, 1, 0, \ldots, 0)^T,$$

gdzie jedynka jest na k-tej pozycji (oczywiście, $\|x^*\|_1 = 1$). Wtedy

$$\|Ax^*\|_1 = \sum_i |\sum_j A_{ij} x_j^*| = \sum_i |A_{ik}|,$$

— co było do okazania. Norma nazywa się *kolumnową*, bo jest równa maksimum z norm $\|\cdot\|_1$ kolumn macierzy A.

Dla normy $\|A\|_\infty$ postępowanie jest podobne:

$$\|Ax\|_\infty = \max_i |\sum_j A_{ij} x_j| \le \max_i \sum_j |A_{ij}| \underbrace{|x_j|}_{\le \|x\|_\infty} \le \|x\|_\infty \cdot \max_i \sum_j |A_{ij}|.$$

Czytelnik zechce dobrać tak znaki w wektorze $x^* = (\pm 1, \ldots, \pm 1)^T$, aby zachodziło $\|Ax^*\|_\infty = \max_i \sum_j |A_{ij}|$. Norma nazywa się *wierszową*, bo jest równa maksimum z norm $\|\cdot\|_1$ wierszy macierzy A. □

5. Wyjaśnij, dlaczego norma macierzowa $\|\cdot\|_2$ nazywa się normą spektralną.

Rozw. Norma nazywa się *spektralną*, bo

$$\|A\|_2 = \max\{\sqrt{\mu} : \mu \text{ jest wartością własną macierzy } A^T A\},$$

czyli jest określona przez wartości własne, czyli widmo (czyli *spektrum*) pewnej macierzy. Za chwilę udowodnimy, że powyższa równość jest prawdziwa.

Podobnie jak w poprzednim zadaniu pokażemy najpierw oszacowanie, a potem, że jest ono przyjmowane.

Zauważmy, że macierz $B = A^T A$ jest symetryczna (bo $B^T = (A^T A)^T = A^T (A^T)^T = A^T A = B$). Zatem wszystkie jej wartości własne μ_i są rzeczywiste[1], oraz istnieje baza ortogonalna, $\{v_i\}$, złożona z wektorów własnych B. Oznaczmy $\mu_{\max} = \max_i \mu_i$. Niech $x \in \mathbb{R}^M$ będzie dowolnym wektorem. Zapiszmy go w bazie wektorów własnych B,

$$x = \sum_i \alpha_i v_i.$$

[1] Dodatkowo, są też nieujemne: dlaczego?

Będzie nam wygodniej rozważać kwadraty norm; na koniec wszystkie nierówności obłożymy pierwiastkiem. Korzystając z ortogonalności bazy, tzn. $v_i^T v_j = \delta_{ij}$ (←delta Kroneckera):

$$\begin{aligned}\|Ax\|_2^2 = (Ax)^T Axx^T A^T Ax = x^T Bx = (\sum_i \alpha_i v_i^T) B (\sum_j \alpha_j v_j) \\ = (\sum_i \alpha_i v_i^T)(\sum_j \alpha_j \underbrace{Bv_j}_{=\mu_j v_j}) = \sum_{i,j} \alpha_i \alpha_j \mu_j \underbrace{v_i^T v_j}_{=\delta_{ij}} \\ = \sum_i \alpha_i^2 \mu_j \le \mu_{\max} \underbrace{\sum_i \alpha_i^2}_{=\|x\|_2^2}.\end{aligned}$$

Równość (bez oszacowań) zachodzi oczywiście na $x^* = v_{\max}$ — wektorze własnym odpowiadającym wartości własnej $\mu_{\max}$. □

6. Wykaż, że dla dowolnej normy macierzowej indukowanej normą wektorową

$$\|AB\| \le \|A\| \cdot \|B\|,$$

gdzie A i B są dającymi się pomnożyć macierzami.

Rozw. Z definicji,

$$\|AB\| = \sup_{x \neq 0} \frac{\|ABx\|}{\|x\|}$$

oraz $\|ABx\| = \|A(Bx)\| \le \|A\|\|Bx\| \le \|A\|\|B\|\|x\|$. □

2.2.2. Rozwiązywanie układów równań — część pierwsza

7. Algorytm rozkładu LU macierzy A symetrycznej i dodatnio określonej jest wykonalny bez wyboru elementu głównego.

Dla takich macierzy na następnym wykładzie zobaczycie Państwo nieco lepszy pomysł na rozkład na iloczyn (innych) dwóch macierzy trójkątnych: to będzie rozkład Cholesky'ego.

Rozw. Na wszelki wypadek przypomnijmy definicję macierzy dodatnio określonej:

$$x^T Ax > 0 \qquad \forall x \neq 0.$$

W macierzy A wyróżnijmy pierwszy wiersz i pierwszą kolumnę:

$$A = \left[\begin{array}{c|c} a_{11} & a_{21}^T \\ \hline a_{21} & A_{22} \end{array}\right]$$

Zwrócmy uwagę, że

— Element a_{11} jest oczywiście liczbą (dodatnią na mocy założenia, że A jest dodatnio określona);
— Pionowy blok a_{21} w ogólności nie jest liczbą, tylko *jest wektorem* kolumnowym długości $N-1$,
— Block A_{22} jest macierzą rozmiaru $(N-1) \times (N-1)$.
— Poziomy górny blok jest *wektorem* wierszowym długości $N-1$, równym a_{21}^T — bo A jest symetryczna.

Ponieważ A jest symetryczna i dodatnio określona, to A_{22} — też (dlaczego?).

Interesuje nas algorytm obliczenia rozkładu LU (bez wyboru elemnetu głównego), czyli wyznaczenia

— macierzy górnej trójkątnej U,
— macierzy dolnej trójkątnej L z jedynkami na diagonali

takich, że

$$A = L \cdot U.$$

Powyższą równość można zapisać, wykorzystując wprowadzony na początku podział macierzy, w postaci

$$\underbrace{\left[\begin{array}{c|c} a_{11} & a_{21}^T \\ \hline a_{21} & A_{22} \end{array}\right]}_{A} = \underbrace{\left[\begin{array}{c|c} 1 & 0^T \\ \hline l_{21} & L_{22} \end{array}\right]}_{L} \cdot \underbrace{\left[\begin{array}{c|c} u_{11} & u_{12}^T \\ \hline 0 & U_{22} \end{array}\right]}_{U},$$

gdzie L_{22} jest dolna trójkątna z jedynkami na diagonali, a U_{22} jest górna trójkątna. Po wymnożeniu[2] L przez U i przyrównaniu do A dostajemy zależności[3]:

$$\begin{aligned}
a_{11} &= 1 \cdot u_{11} + 0^T \cdot 0 = u_{11} && \Longrightarrow\ u_{11} = a_{11}, \\
a_{21} &= l_{21} \cdot u_{11} + L_{22} \cdot 0 = l_{21} \cdot u_{11} && \Longrightarrow\ l_{21} = \frac{1}{u_{11}} a_{21}, \\
a_{21}^T &= 1 \cdot u_{12}^T + 0^T \cdot U_{22} = u_{12}^T && \Longrightarrow\ u_{12} = a_{21}, \\
A_{22} &= l_{21} \cdot u_{12}^T + L_{22} \cdot U_{22} && \Longrightarrow\ L_{22}U_{22} = \underbrace{A_{22} - l_{21} \cdot u_{12}^T}_{=:\tilde{A}_{22}}.
\end{aligned}$$

Widzimy stąd, że fragmenty l_{21}, u_{11}, u_{12} rozkładu jest bardzo łatwo wyznaczyć wprost, a dalej już pozostaje wyznaczyć czynniki L_{22} i U_{22} rozkładu LU macierzy rozmiaru o jeden mniejszego: $\tilde{A}_{22}$.

Jak go wyznaczymy? Oczywiście *tak samo*! To więc prowadzi do rekurencji, która zatrzyma się w momencie, gdy przyjdzie nam wyznaczyć rozkład macierzy 1×1. Nietrudno zauważyć, że w rzeczywistości można algorytm zapisać w prostszej postaci, wyłącznie przy użyciu pętli, jak poniżej[4]:

for $k = 1 : N-1$
  for $i = k+1 : N$
    $a_{ik} = a_{ik}/a_{kk}$ ▷ wyznaczenie k-tej kolumny L
  end
  for $i = k+1 : N$
    for $j = k+1 : N$
      $a_{ij} = a_{ij} - a_{ik}a_{kj}$ ▷ wyznaczenie $\tilde{A}_{22}$
    end
  end
end

[2] Mnożenie macierzy w postaci blokowej wykonuje się tak samo, jak mnożenie macierzy zwykłych, traktując bloki jak zwykłe elementy, pamiętając o jednym odstępstwie: mnożenie bloków nie jest przemienne. Kto nie wierzy, niech to sobie przeliczy.

[3] Pamiętajmy, że a_{21}, l_{21}, u_{12} są wektorami, więc nie zawsze mnożenie przez nie jest przemienne.

[4] Rozkład wykonujemy w miejscu, nadpisując A czynnikami L i U. Jedynek z diagonali L nie zapisujemy, bo i tak wiemy, że tam są...

Przejdźmy wreszcie do zasadniczego pytania postawionego w zadaniu. Czy nie będzie potrzebny wybór elementu głównego, tzn. czy nie zdarzy się, że na którymś kroku podanego powyżej algorytmu będziemy dzielić przez zero?

Jak już powiedzieliśmy, $a_{11} > 0$, więc pierwszy krok algorytmu na pewno jest wykonalny: nie będziemy dzielić przez zero. Pokażemy więc teraz, że macierz $\tilde{A}_{22}$ też jest symetryczna i dodatnio określona! A skoro tak, to wykonując pierwszy krok *jej* rozkładu LU też będziemy dzielić przez liczbę dodatnią, itd. — więc na *każdym* kroku algorytmu będziemy dzielić przez liczbę dodatnią, czyli algorytm nie załamie się — i o to nam właśnie chodziło.

Interesuje nas, czy $x^T \tilde{A}_{22} x > 0$ dla dowolnego $x \neq 0$. Musimy to jakoś wywnioskować z analogicznej własności dla A:

$$\begin{bmatrix} \alpha & x^T \end{bmatrix} \begin{bmatrix} a_{11} & a_{21}^T \\ a_{21} & A_{22} \end{bmatrix} \begin{bmatrix} \alpha \\ x \end{bmatrix} > 0 \qquad \text{dla } \begin{bmatrix} \alpha \\ x \end{bmatrix} \neq 0.$$

Ponieważ

$$\begin{bmatrix} \alpha & x^T \end{bmatrix} \begin{bmatrix} a_{11} & a_{21}^T \\ a_{21} & A_{22} \end{bmatrix} \begin{bmatrix} \alpha \\ x \end{bmatrix} = \alpha^2 a_{11} + 2\alpha a_{21}^T x + x^T A_{22} x,$$

to dla $\alpha = -\dfrac{a_{21}^T x}{a_{11}}$ dostajemy

$$0 < \begin{bmatrix} \alpha & x^T \end{bmatrix} \begin{bmatrix} a_{11} & a_{21}^T \\ a_{21} & A_{22} \end{bmatrix} \begin{bmatrix} \alpha \\ x \end{bmatrix} = x^T \tilde{A}_{22} x,$$

co należało wykazać. □

2.3. Deser — nieobowiązkowy!

To są zadania o tematyce nieco pobocznej, dla chętnych i zainteresowanych.

8. Wykaż, że jeśli A jest odwracalna i $\|A^{-1}\| \cdot \|\Delta\| < 1$, to $(A+\Delta)^{-1}$ istnieje oraz

$$\|(A+\Delta)^{-1}\| \leq \frac{\|A^{-1}\|}{1 - \|A^{-1}\|\|\Delta\|}$$

Z tego faktu korzysta oszacowanie uwarunkowania układu równań liniowych (str. 4.2 skryptu).

Rozw. Rozwiązania poszukaj w *Lecture 15* w książeczce Stewarta. □

9. Rozważmy układ równań $Ax = b$ dla zadanych A i b. Niech $\tilde{x}$ — wynik obliczony w fl *jakimś* algorytmem. Udowodnij, że jeśli

$$\frac{\|b - A\tilde{x}\|_2}{\|A\|_2\|\tilde{x}\|_2 + \|b\|_2} \leq \varepsilon,$$

to istnieją Δ, δ t. że

$$(A+\Delta)\tilde{x} = b + \delta \quad \text{oraz} \quad \frac{\|\Delta\|_2}{\|A\|_2}, \frac{\|\delta\|_2}{\|b\|_2} \leq \varepsilon.$$

Powyższy warunek nazywamy *numerycznym kryterium „numerycznej poprawności”* uzyskanego wyniku.

10. Wyznacz uwarunkowanie (mierzone w normie $\|\cdot\|_1$) macierzy $N \times N$

$$\begin{bmatrix} 1 & \dots & 1 & 1 \\ \varepsilon & & & \\ & \ddots & & \\ & & \varepsilon & \end{bmatrix}$$

w zależności od $\varepsilon \neq 0$ i od N.

W dalszej części zadania przyjmiemy, że $\varepsilon \approx 0$.

— Jak dużym błędem będzie obarczony rozkład LU tej macierzy wyznaczony w arytmetyce fl podwójnej precyzji?

— Czy rozwiązanie układu równań, wyznaczone w arytmetyce fl podwójnej precyzji z wykorzystaniem tego rozkładu LU, może być obarczone dużym błędem względnym?

3. Konsultacje

Będę online na czacie w najbliższą środę w standardowym terminie zajęć i potem dodatkową godzinę.

Można też zadawać pytania na czacie offline, najlepiej w grupie `mo-cwiczenia-grupa2`, abyśmy mogli uczyć się od siebie nawzajem i uniknęli powtarzających się pytań.

Oczywiście, można też użyć maila.

Matematyka obliczeniowa: ćw. zdalne 2020-03-25
Piotr Krzyżanowski

---

1. Polecana literatura

Zajęcia dotyczą mniej więcej stron 80...90 skryptu Wykładowcy, P. Kiciaka — czyli dalszego ciągu rozdziału 4 (na razie bez odbić Householdera). Ponadto każdy szanujący się podręcznik z metod numerycznych ma rozdziały o różnych metodach rozwiązywania specjalnych typów układów równań liniowych.

Dla dobrego humoru i lepszego rozeznania warto przeczytać *Lectures 12, 13, 14* z *Afternotes on Numerical Analysis*. Kto jeszcze nie zrobił tego w poprzednim tygodniu, niech też doczyta pozostałe wykłady o rozwiązywaniu równań liniowych z książeczki G.W.Stewarta.

2. Zadania z rozwiązaniami

Namawiam Państwa do próby samodzielnego rozwiązania każdego z zadań, a dopiero potem do przeczytania gotowca. O ile nie będzie powiedziane inaczej, zawsze będziemy zakładać, że obliczenia są wykonywane w arytmetyce dokładnej.

2.1. Przystawki

0. *Najpierw przeczytaj polecaną literaturę!*

1. Dany jest rozkład $LU = PA$ nieosobliwej macierzy A rozmiaru $N \times N$, pochodzący z eliminacji Gaussa z wyborem elementu głównego w kolumnie. Zidentyfikuj macierze P, L i U, a następnie powiedz,

1. jaki jest koszt wyznaczenia tego rozkładu,
2. jak rozwiązać układ równań $Ax = b$ korzystając z tego rozkładu i ile to kosztuje,
3. jak rozwiązać układ równań $AX = B$, gdzie X, B są macierzami $N \times M$, korzystając z tego rozkładu i ile to kosztuje.

Rozw.

1. Koszt rozkładu LU to oczywiście (jeśli nie wykorzystamy jakichś specjalnych własności macierzy) $\mathcal{O}(N^3)$ flopów.
2. Gdy znamy już rozkład LU, rozwiązanie układu to seria kroków:
   a) Oblicz $g = Pb$ (koszt zero flopów, bo to tylko permutacja elementów b)
   b) Rozwiąż $Ly = g$ (koszt $\mathcal{O}(N^2)$)
   c) Rozwiąż $Ux = y$ (koszt $\mathcal{O}(N^2)$)
   (jeśli nie wykorzystamy jakichś specjalnych własności macierzy L, U).
3. Dla wersji, gdy B jest macierzą sytuacja jest niemal identyczna: Najpierw przestaw kolumny B zgodnie z permutacją indeksów P, a następnie dla $i = 1, \ldots, M$, wykonaj następujące kroki:

a) Rozwiąż $Ly = g_i$ (koszt $\mathcal{O}(N^2)$)
b) Rozwiąż $Ux = y$ (koszt $\mathcal{O}(N^2)$)

Całkowity koszt to więc $\underbrace{\mathcal{O}(N^3)}_{\text{koszt rozkładu}} + \underbrace{\mathcal{O}(MN^2)}_{\text{koszt rozwiązania}}$. To wyraźnie mniej, niż gdybyśmy M razy „od zera" rozwiązali serię układów $Ax_i = b_i$, za każdym razem (niepotrzebie) od nowa wyznaczając rozkład LU macierzy A. Takie podejście kosztowałoby nas $\mathcal{O}(MN^3)$.

□

2. Macierz A rozmiaru $N \times N$ nazywamy diagonalnie dominującą, jeśli zachodzi

$$|a_{ii}| > \sum_{j \neq i} |a_{ij}| \qquad \forall i = 1, \ldots, N.$$

Wykaż, że rozkład LU macierzy diagonalnie dominującej można wykonać bez wyboru elementu głównego.

Rozw. Analogicznie jak w zadaniu z **poprzednich** ćwiczeń korzystamy z postaci blokowej

$$A = \begin{bmatrix} a_{11} & a_{12}^T \\ a_{21} & A_{22} \end{bmatrix}$$

Z warunku diagonalnej dominacji oczywiście $a_{11} > 0$, więc można przezeń dzielić w pierwszym kroku eliminacji Gaussa. Pokażemy, że macierz

$$\tilde{A}_{22} = A_{22} - l_{21} \cdot u_{12}^T = A_{22} - \frac{1}{a_{11}} a_{21} \cdot a_{12}^T$$

też jest diagonalnie dominująca, skąd wykonalność całego algorytmu tak, jak w zadaniu **??**.

Do pokazania jest więc

$$|a_{ii} - \frac{1}{a_{11}} a_{i1} \cdot a_{1i}| > \sum |a_{ij} - \frac{1}{a_{11}} a_{i1} \cdot a_{1j}| \qquad \text{dla } i = 2, \ldots, N,$$

gdzie — uwaga! — sumujemy po $j \geq 2$ takich, że $j \neq i$. Wystarczy pokazać, że dla $i = 2, \ldots, N$

$$\begin{aligned} |a_{ii}| - |\frac{1}{a_{11}} a_{i1} \cdot a_{1i}| &> \sum |a_{ij}| + \sum |\frac{1}{a_{11}} a_{i1} \cdot a_{1j}| \\ |a_{ii}| &> \sum |a_{ij}| + \sum |\frac{1}{a_{11}} a_{i1} \cdot a_{1j}| + |\frac{1}{a_{11}} a_{i1} \cdot a_{1i}| \\ |a_{ii}| &> \sum |a_{ij}| + |a_{i1}| \frac{1}{|a_{11}|} \underbrace{\sum_{j \neq 1} |a_{1j}|}_{< |a_{11}|}, \end{aligned}$$

na mocy założenia od diagonalnej dominacji w pierwszym wierszu. Zatem wystarczy pokazać, że

$$|a_{ii}| > \sum_{\substack{j \geq 2 \\ j \neq i}} |a_{ij}| + |a_{i1}| = \sum_{j \neq i} |a_{ij}|,$$

co jest prawdą na mocy diagonalnej dominacji w i-tym wierszu. □

3. Niech $u, w \in \mathbb{R}^N$ będą dowolnymi wektorami. Wykaż, że $\det(I - wu^T) = 1 - u^T w$.

Rozw. (Szkic.) Znajdujemy — najlepiej wprost z definicji, a nie jako miejsce zerowe wielomianu charakterystycznego – wartości własne μ_i macierzy wu^T: wiele z nich będzie zerowych, wszak to macierz rzędu 1! Wartości własne naszej macierzy są równe $1 - \mu_i$.

Ponieważ na mocy rozkładu Jordana $\det(A)$ jest równy iloczynowi wartości własnych A, to w naszym przypadku

$$\det(I - wu^T) = (1 - u^T w) \cdot 1 \cdot \ldots \cdot 1 = 1 - u^T w.$$

□

2.2. Danie główne

4. Opracuj algorytm wyznaczania rozkładu LU bez wyboru elementu głównego macierzy trójdiagonalnej, tzn. postaci

$$A = \begin{bmatrix} a_1 & b_1 & & \\ c_2 & a_2 & \ddots & \\ & \ddots & \ddots & b_{N-1} \\ & & c_N & a_N \end{bmatrix}$$

(w pozostałych miejscach są zera).

Następnie opracuj algorytm rozwiązywania układu równań $Ax = b$ z wykorzystaniem otrzymanego rozkładu. Oszacuj koszt obliczeniowy i pamięciowy obu algorytmów.

Rozw. Oczywiście, nie dla każdej macierzy nieosobliwej da się wyznaczyć rozkład LU bez wyboru elementu głównego — my założymy, że dla naszej macierzy tak się *da*.

Analogicznie jak w zadaniu **??** korzystamy z postaci blokowej

$$\underbrace{\begin{bmatrix} a_1 & a_{12}^T \\ a_{12} & A_{22} \end{bmatrix}}_{A} = \underbrace{\begin{bmatrix} 1 & 0^T \\ l_{21} & L_{22} \end{bmatrix}}_{L} \underbrace{\begin{bmatrix} u_{11} & u_{12}^T \\ 0 & U_{22} \end{bmatrix}}_{U},$$

gdzie L_{22} jest dolna trójkątna z jedynkami na diagonali, a U_{22} jest górna trójkątna.

Tym razem w wektorach a_{12} i a_{21} są same zera z wyjątkiem pierwszej pozycji:

$$a_{21} = (c_2, 0, \ldots, 0)^T, \qquad a_{12}^T = (b_1, 0, \ldots, 0),$$

więc (por. zadanie z **poprzednich** ćwiczeń) w u_{12} i l_{21} też:

$$l_{21} = (c_2/a_1, 0, \ldots, 0)^T, \qquad u_{12}^T = (b_1, 0, \ldots, 0).$$

Jak więc będzie wyglądać macierz mniejszego rozmiaru $\tilde{A}_{22}$, w której będziemy prowadzić dalszą eliminację (tzn. której rozkład $\tilde{A}_{22} = L_{22}U_{22}$ należy wyznaczyć, by zakończyć algorytm)? Mamy

$$\tilde{A}_{22} = A_{22} - l_{21}u_{12}^T = \begin{bmatrix} a_2 & b_2 & & \\ c_3 & a_3 & \ddots & \\ & \ddots & \ddots & b_{N-1} \\ & & c_N & a_N \end{bmatrix} - \begin{bmatrix} \star & & & \\ & & & \\ & & & \\ & & & \end{bmatrix} = \begin{bmatrix} \star & b_2 & & \\ c_3 & a_3 & \ddots & \\ & \ddots & \ddots & b_{N-1} \\ & & c_N & a_N \end{bmatrix},$$

gdzie symbolu $\star$ użyliśmy, by oznaczyć jakieś, być może różne, ale (potencjalnie) niezerowe elementy w macierzy. A więc $\tilde{A}_{22}$ też jest trójdiagonalna, więc przez prostą indukcję macierze L i U będą miały po dwie niezerowe diagonale.

Koszt pamięciowy będzie więc nie większy niż $3N$ — potrzebujemy miejsca na trzy diagonale A, które następnie nadpiszemy czynnikami rozkładu. Koszt obliczeniowy też będzie liniowy względem N, bo na jednym kroku algorytmu wykonujemy 3 flopy, a kroków do wykonania jest $N-1$.

Napisanie algorytmu pozostawiam Państwu jako ćwiczenie do samodzielnego wykonania. □

5. Opracuj wariant algorytmu rozwiązywania układu równań z macierzą trójdiagonalną z wyborem elementu głównego w kolumnie. Oszacuj jego koszt obliczeniowy i pamięciowy.

Rozw. Pojawi się jedna dodatkowa nad-diagonala w macierzy U. Koszt będzie dalej liniowy. □

6. Niech A będzie ustaloną macierzą nieosobliwą $N \times N$ i przypuśćmy, że dysponujemy tanią metodą o koszcie $\mathcal{O}(N^k)$, gdzie $k < 3$, rozwiązywania układu równań $Ax = b$ dla dowolnego $b \in \mathbb{R}^N$. (Innymi słowy — dysponujemy procedurą $\mathsf{x = solve(b)}$, która dla dowolnego b wyznacza x taki, że $Ax = b$.) Jak rozwiązać układ rozszerzony o jeden dodatkowy wiersz i jedną dodatkową kolumnę:

$$\begin{bmatrix} A & v \\ u^T & \alpha \end{bmatrix} \begin{bmatrix} x \\ \xi \end{bmatrix} = \begin{bmatrix} b \\ \beta \end{bmatrix}.$$

Zapisz w pseudokodzie algorytm. Jaki jest jego koszt w zależności od N?

Rozw. Przede wszystkim, nie wiemy, czy tak stworzona macierz nie będzie czasem osobliwa (a może, np. jeśli $v = 0$ i $\alpha = 0$...) Częścią naszego rozwiązania będzie więc sprawdzenie, czy rozwiązywany układ nie jest osobliwy.

Pokażemy dwa sposoby rozwiązania tego zadania:

Sposób I: Eliminacja Zapiszmy układ w postaci naturalnej:

$$\begin{aligned} Ax + v\xi &= b \\ u^T x + \alpha\xi &= \beta. \end{aligned}$$

Ponieważ A jest nieosobliwa, to możemy — napisać tylko, na razie — że A^{-1} istnieje, więc z pierwszego równania $x = A^{-1}(b - v\xi)$. Podstawiając do drugiego równania, dostajemy równanie skalarne z jedną niewiadomą, ξ:

$$(\alpha - u^T A^{-1} v)\xi = \beta - A^{-1} b,$$

który ma jednoznaczne rozwiązanie wtw gdy $\alpha - u^T A^{-1} v \neq 0$ — i jest to warunek konieczny i dostateczny (dlaczego?) nieosobliwości macierzy rozszerzonej. Stąd algorytm:

c = solve(b); w = solve(v); ▷ Koszt $2\mathcal{O}(N^k)$
$\gamma = \alpha - u^T w$; ▷ Koszt $\mathcal{O}(N)$
if $\gamma == 0$ **then**
  Error ▷ Macierz rozszerzona jest osobliwa!
end if
$\xi = (\beta - c)/\gamma$; ▷ Koszt $\mathcal{O}(1)$
$x = c - \xi w$ ▷ Koszt $\mathcal{O}(N)$

całkowity koszt to $2\mathcal{O}(N^k) + \mathcal{O}(N)$.

Sposób II: Rozkład blokowy Zapiszmy blokowy rozkład „LU" macierzy rozszerzonej:

$$\begin{bmatrix} A & v \\ u^T & \alpha \end{bmatrix} = \underbrace{\begin{bmatrix} A & 0 \\ u^T & 1 \end{bmatrix}}_{\text{nieosobliwa}} \begin{bmatrix} I & w \\ 0 & \gamma \end{bmatrix}$$

Ten rozkład istnieje, o ile spełnione są zależności:

$$\begin{aligned} Aw = v \qquad &\Longrightarrow \qquad w = A^{-1}v \\ \alpha = d^T f + \gamma \qquad &\Longrightarrow \qquad \gamma = \alpha - u^T A^{-1} v. \end{aligned}$$

A więc istnieje on zawsze. Ponadto macierz po lewej stronie jest nieosobliwa wtw gdy obie macierze po prawej stronie sa nieosobliwe, a tak jest wtw gdy $\gamma \neq 0$. Reszta algorytmu przebiega tak samo. Szczegóły do dopracowania samodzielnie. □

2.3. Deser — nieobowiązkowy!

To są zadania o tematyce nieco pobocznej, dla chętnych i zainteresowanych.

7. Podaj algorytm rozwiązywania układu równań $Ax = b$ kosztem $\mathcal{O}(N^2)$, gdy A jest tzw. macierzą cykliczną o stałych diagonalach:

$$A = \begin{pmatrix} c_1 & c_N & \dots & c_2 \\ c_2 & c_1 & \ddots & \vdots \\ \vdots & \ddots & \ddots & c_N \\ c_N & \dots & c_2 & c_1 \end{pmatrix}.$$

8. Wykaż, że dysponując metodą mnożenia dwóch macierzy kwadratowych $n \times n$ o koszcie $\mathcal{O}(n^\omega)$, gdzie $\omega \leq 3$, można wyznaczyć rozkład macierzy symetrycznej i dodatnio określonej rozmiaru N kosztem $\mathcal{O}(N^\omega)$. Przyjmij dla uproszczenia, że N jest potęgą dwójki.

Ten zaskakujący wynik pokazuje, że złożoność rozwiązywania układu równań jest (z dokładnością do stałej) taka sama, jak złożoność mnożenia dwóch macierzy. Algorytm Strassena mnożenia macierzy ma $\omega < 3$, a więc — przynajmniej teoretycznie — *eliminacja Gaussa nie jest optymalna*, jak zatytułował swój słynny artykuł z 1969 roku.

Matematyka obliczeniowa: ćw. zdalne 2020-04-01
Piotr Krzyżanowski

---

1. Polecana literatura

To kolejne zajęcia, które znów dotyczą mniej więcej stron 80…90 skryptu Wykładowcy, P. Kiciaka — czyli dalszego ciągu rozdziału 4 (tym razem z uwzględnieniem odbić Householdera). Ponadto każdy szanujący się podręcznik z metod numerycznych ma rozdziały o różnych metodach rozwiązywania specjalnych typów układów równań liniowych. Metody Householdera i Givensa zazwyczaj są tam omawiane w kontekście zadania najmniejszych kwadratów (*least squares problem*)

2. Zadania z rozwiązaniami

Namawiam Państwa do próby samodzielnego rozwiązania każdego z zadań, a dopiero potem do przeczytania gotowca. O ile nie będzie powiedziane inaczej, zawsze będziemy zakładać, że obliczenia są wykonywane w arytmetyce dokładnej.

2.1. Przystawki

0. *Najpierw przeczytaj polecaną literaturę!*

1. Dany jest rozkład QR nieosobliwej macierzy A rozmiaru $N \times N$. Zidentyfikuj macierze Q, R, a następnie powiedz,

1. jak rozwiązać układ równań $Ax = b$ korzystając z tego rozkładu i ile to kosztuje,
2. jak rozwiązać układ równań $AX = B$, gdzie X, B są macierzami $N \times M$, korzystając z tego rozkładu i ile to kosztuje.

Porównaj z zadaniem 5.

Rozw. Q jest macierzą ortogonalną, tzn $Q^T Q = I$. Macierz R jest górna trójkątna (tak, jak U w rozkładzie LU).

Aby rozwiązać $AX = B$, oznaczmy przez x_i kolumny X i analogicznie b_i — kolumny B. Wtedy wystarczy:

for $i = 1 : M$
  Oblicz $y = Q^T b_i$ ▷ koszt: $\mathcal{O}(N^2)$
  Rozwiąż $Rx_i = y$ ▷ koszt: $\mathcal{O}(N^2)$
end

Łączny koszt to $\mathcal{O}(N^2 M)$.

Warto zwrócić uwagę na to, że możemy dysponować wyspecjalizowanym solverem, który umie rozwiązywać układy $RX = Y$ z macierzą Y rozmiaru $N \times M$. (Takie solvery istnieją i potrafią — wykorzystując różne tricki — zmniejszyć *czas* potrzebny na rozwiązanie układu.) Wtedy prościej i skuteczniej:

Oblicz $Y = Q^T B$ ▷ koszt: $\mathcal{O}(N^2 M)$
Rozwiąż $RX = Y$ ▷ koszt: $\mathcal{O}(N^2 M)$

(Zysk jest na czasie wykonania, a nie na koszcie mierzonym we flopach.)

□

2. Wykaż, że jeśli Q jest macierzą ortogonalną, to $\|Q\|_2 = 1$.

Rozw. Z definicji, $\|Q\| = \sup_{\|x\|=1} \|Qx\|$. Ponadto Q jest izometrią, bo

$$\|Qx\|_2^2 = x^T Q^T Q x = x^T I x = \|x\|_2^2,$$

stąd teza. □

3. Wykaż, że wartości własne λ rzeczywistej macierzy ortogonalnej Q spełniają $|\lambda| = 1$. Czy musi być $\lambda = \pm 1$?

Rozw. Z definicji i z własności izometrii, $|\lambda| = 1$. Oczywiście λ, bo może być zespolona. □

4. Przekształcenie Householdera. Koszt wyznaczenia H zerującego wszystkie współrzędne zadanego wektora a z wyjatkiem pierwszej. Koszt mnożenia H przez wektor x.

Rozw. Jeśli $\|v\|_2 = 1$, to przekszt. Householdera zadane przez v to $H = I - 2vv^T$. Koszt wyznaczenia v t. że $Ha = \text{Const} \cdot e_1$ jest liniowy wzgl. N (wiedza z wykładu). Ponieważ

$$Hx = x - 2vv^T x = x - \underbrace{2v^T x}_{\text{liczba}} \cdot v,$$

to koszt też jest liniowy. □

5. Realizacja rozkładu QR z wykorzystaniem przekształceń Householdera. Rozwiązanie układu równań z wykorzystaniem tak otrzymanego rozkładu.

Rozw. Było na wykładzie! Por. także zadanie 8. □

2.2. Danie główne

6. Przekształcenie Givensa to

$$G_{ij} = \begin{bmatrix} 1 & & & & & & \\ & \ddots & & & & & \\ & & c & & s & & \\ & & & \ddots & & & \\ & & -s & & c & & \\ & & & & & \ddots & \\ & & & & & & 1 \end{bmatrix}$$

(na diagonali są same jedynki, z wyjątkiem pozycji (i,i) oraz (j,j), gdzie są liczby c; ponadto na pozycji (i,j) jest liczba s, a na (j,i) — liczba $(-s)$; poza tym — same zera). Ponadto zawsze zakładamy, że zachodzi warunek

$$s^2 + c^2 = 1.$$

Sprawdź, że G_{ij} jest macierzą ortogonalną, tzn. $G_{ij}^T G_{ij} = I$. Wyznacz G_{ij} zerujące j-tą współrzędną zadanego wektora a; ile kosztuje obliczenie odpowiednich c i s? Wyznacz koszt mnożenia G_{ij} przez wektor x.

Rozw. Koszt wyznaczenia i pomnożenia przez G_{ij} (reprezentowane przez parę c,s) jest stały i niezależny od rozmiaru G_{ij}, ani od i,j. □

7. Realizacja rozkładu QR macierzy Hessenberga z wykorzystaniem przekształceń Givensa. Koszt.

Rozw. Nasza macierz $A \in R^{N\times N}$ jest postaci Hessenberga, tzn.

$$A = \begin{bmatrix} * & * & \cdots & * \\ * & * & \cdots & * \\ & \ddots & \ddots & \vdots \\ & & * & * \end{bmatrix}.$$

Niech pierwszy krok rozkładu QR polega na wyznaczeniu — kosztem $\mathcal{O}(1)$ — obrotu Givensa G_{12} takiego, że

$$G_{12} \cdot \begin{bmatrix} a_{11} \\ a_{21} \\ 0 \\ \vdots \\ 0 \end{bmatrix} = \begin{bmatrix} r_{11} \\ 0 \\ 0 \\ \vdots \\ 0 \end{bmatrix}$$

dla jakiegoś r_{11}. (Skądinąd, zachodzi $r_{11}^2 = a_{11}^2 + a_{21}^2$ — dlaczego?)

Następnie mnożymy A przez G_{12}. Skoro koszt mnożenia wektora przez obrót Givensa jest stały, równy Const, to koszt mnożenia $N-1$ kolumn macierzy A jest równy $\text{Const}\cdot(N-1)$.

Po przemnożeniu $G_{12}\cdot A$, dostajemy

$$G_{12}\cdot A = \begin{bmatrix} r_{11} & \star & \cdots & \star \\ 0 & \star & \cdots & \star \\ & \ddots & \ddots & \vdots \\ & & * & * \end{bmatrix} \equiv \begin{bmatrix} r_{11} & r_{12} & r_{13} & \cdots & r_{1N} \\ & \tilde{a}_{22} & \star & \cdots & \star \\ & a_{32} & * & \cdots & * \\ & & \ddots & \ddots & \vdots \\ & & & * & * \end{bmatrix} =: A_2.$$

W następnym kroku powtórzymy więc to samo postępowanie dla drugiej kolumny macierzy wynikowej A_2, innymi słowy, użyjemy G_{23} takiego, że

$$G_{23} \cdot \begin{bmatrix} r_{12} \\ \tilde{a}_{22} \\ \tilde{a}_{32} \\ 0 \\ \vdots \\ 0 \end{bmatrix} = \begin{bmatrix} r_{12} \\ r_{22} \\ 0 \\ 0 \\ \vdots \\ 0 \end{bmatrix}$$

dla jakiegoś r_{11}. Jak widać, G_{23} *nie zmieni* pierwszej kolumny A_{23} (bo tam gdzie miesza współrzędne, i tak były dwa zera), więc nie musimy w ogóle mnożyć tej kolumny przez G_{23}. Zatem koszt pomnożenia A_2 przez G_{23} będzie równy $\text{Const}\cdot(N-2)$.

I tak dalej: na i–tym kroku wyznaczymy $G_{i,i+1}$ zerujące element $(i+1,i)$ macierzy A_i, a następnie obliczymy $A_{i+1} = G_{i,i+1}\cdot A_i$, co będzie kosztować $\text{Const}\cdot(N-i)$ flopów (bo we wcześniejszych kolumnach będą zera w i-tym i kolejnym wierszu).

Po $N-1$ iteracjach dostaniemy więc

$$\underbrace{G_{N-1,N}\cdot\ldots\cdot G_{23}\cdot G_{12}}_{=Q^T}\cdot A = R$$

i w sumie będzie to nas kosztować:

— $\mathcal{O}(N)$ flopów — wyznaczenie kolejnych obrotów,
— plus $\text{Const}\big((N-1)+(N-2)+\ldots+1\big) = \mathcal{O}(N^2)$ — kolejne mnożenia $G_{i,i+1}\cdot A_i$.

Zatem łączny koszt to $\mathcal{O}(N^2)$. □

8. Rozwiązywanie układu równań liniowych z macierzą Hessenberga, z wykorzystaniem rozkładu QR metodą Givensa. Koszt.

Rozw. Aby rozwiązać układ równań $Ax = b$ wykorzystamy obliczone czynniki rozkładu. Mamy

$$G_{N-1,N}\cdot\ldots\cdot G_{23}\cdot G_{12}\cdot\underbrace{Ax}_{=b} = Rx,$$

więc następujący algorytm wyznaczy rozwiązanie kosztem $\mathcal{O}(N^2)$:

$y = b$
for $i = 1:N-1$ ▷ Całkowity koszt tej pętli to: $\mathcal{O}(N)$, bo...
  $y = G_{i,i+1}\cdot y$ ▷ Koszt: $\mathcal{O}(1)$
end
Rozwiąż $Rx = y$ ▷ Koszt: $\mathcal{O}(N^2)$

Zauważmy, że *nie wyznaczamy* wcale Q: byłaby to niepotrzebna rozrzutność! □

2.3. Deser — nieobowiązkowy!

To są zadania o tematyce nieco pobocznej, dla chętnych i zainteresowanych.

9. Niech A będzie diagonalnie dominującą macierzą $N\times N$ o dodatniej diagonali, o strukturze

$$A = \begin{bmatrix} \star & \star & \cdots & \star \\ \star & \star & & \\ \vdots & & \ddots & \\ \star & & & \star \end{bmatrix}$$

(jest to tzw. *strzałka Wilkinsona*). Wyznacz kosztem $\mathcal{O}(N)$ rozkład $P_1AP_2 = LU$, gdzie P_1, P_2 — macierze permutacji, a L, U — są czynnikami rozkładu LU.

10. Niech $\omega = e^{-2i\pi/N}$ i niech F będzie (zespoloną) macierzą $N \times N$ postaci:

$$F = \frac{1}{\sqrt{N}} \begin{bmatrix} 1 & 1 & 1 & \cdots & 1 \\ 1 & \omega & \omega^2 & \cdots & \omega^{N-1} \\ 1 & \omega^2 & \omega^4 & \cdots & \omega^{2(N-1)} \\ & & & \cdots & \\ 1 & \omega^{N-1} & \omega^{2(N-1)} & \cdots & \omega^{(N-1)^2} \end{bmatrix}.$$

Wykaż, że F spełnia $F^*F = I$. Znajdź możliwie tani algorytm rozwiązywania układu równań z F w przypadku, gdy N jest całkowitą potęgą liczby 2.

Matematyka obliczeniowa: praca ***samodzielna*** 2020-04-15
Piotr Krzyżanowski

Zapewne już tylko wieloletnia tradycja powoduje, że na różnych kolokwiach i egzaminach z tego przedmiotu pojawiają się zadania *mechaniczne* typu poniżej przedstawionych. Rozwiązanie każdego z nich można sprawdzić w MATLAB-ie, dlatego tym razem ich nie podaję, ani też nie będę weryfikować.

Nie chcę, byśmy na to tracili czas podczas ćwiczeń — ale skoro w świetle powyższego taka mechaniczna sprawność rachunkowa może się Państwu przydać w nader stresujących okolicznościach — zamieszczam poniżej **próbkę** tego typu zadań, w części zaproponowanych przez Wykładowcę, do samodzielnego wykonania.

1. Wyznacz (na kartce, krok po kroku) rozkład $PA = LU$ macierzy

$$A = \begin{bmatrix} 2 & -2 & -2 \\ -2 & 0 & 1 \\ 4 & 1 & -2 \end{bmatrix}$$

2. Wyznacz (na kartce, krok po kroku) rozkład $PA = LU$ macierzy

$$A = \begin{bmatrix} 2 & 4 & 8 & 16 \\ 2 & 8 & 24 & 64 \\ 2 & 8 & 24 & 48 \\ 2 & 8 & 16 & 32 \end{bmatrix}$$

3. Za pomocą algorytmu Householdera znajdź (na kartce, krok po kroku) rozkład QR macierzy

$$A = \begin{bmatrix} 0 & -4 \\ 0 & 0 \\ -5 & -2 \end{bmatrix}$$

Wyznacz też (na kartce, krok po kroku) macierz układu równań normalnych dla LZNK z macierzą A.

4. Wyznacz (na kartce, krok po kroku) rozkład Cholesky'ego macierzy

$$A = \begin{bmatrix} 4 & -2 & 0 & -2 \\ -2 & 2 & 1 & 1 \\ 0 & 1 & 5 & 2 \\ -2 & 1 & 2 & 3 \end{bmatrix}$$

i następnie wykorzystaj go do rozwiązania układu $Ax = \begin{bmatrix} 8 & -3 & 13 & 2 \end{bmatrix}^T$.

5. Niech

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

Znajdź wektory reprezentujące przekształcenia Householdera H_i takie, że $R = H_3H_2H_1A$ jest górna trójkątna. Korzystając z tego rozkładu rozwiąż układ $Ax = b$.

Więcej tego typu zadań znajdziecie Państwo nie tylko w skrypcie, a także w niektórych podręcznikach tego przedmiotu.

Matematyka obliczeniowa: ćw. zdalne 2020-04-15
Piotr Krzyżanowski

---

1. Polecana literatura

Zajęcia dotyczą mniej więcej stron 110...120 skryptu Wykładowcy, P. Kiciaka — czyli liniowego zadania najmniejszych kwadratów i związanych z nim rozkładów macierzy: QR i SVD. Oczywiście, każdy szanujący się podręcznik z metod numerycznych ma rozdziały o (*least squares problem*) i tych rozkładów. O SVD będziemy mówić za tydzień, dziś: LZNK i rozkład QR.

2. Zadania z rozwiązaniami

Namawiam Państwa do próby samodzielnego rozwiązania każdego z zadań, a dopiero potem do przeczytania gotowca. O ile nie będzie powiedziane inaczej, zawsze będziemy zakładać, że

— macierz A jest rozmiaru $M \times N$, przy czym $M \geq N$, oraz A jest pełnego rzędu (równego N);
— obliczenia są wykonywane w arytmetyce dokładnej.

2.1. Przystawki

0. *Najpierw przeczytaj polecaną literaturę!*

1. Jak znaleźć wielomian postaci $w(\xi) = s + a\xi + b\xi^2 + c\xi^3$, spełniający warunek $w(0) = 0$ i najlepiej aproksymujący (w sensie średniokwadratowym) wartości $y_1, \ldots, y_M$ w punktach $x_1, \ldots, x_M$? Podaj algorytm i oszacuj jego koszt w zależności od M.

Rozw. Formułujemy zadanie najmniejszych kwadratów: $\sum_i (w(x_i) - y_i)^2 \to \min!$, skąd w postaci macierzowej $||b - Ax||_2 \to \min!$, gdzie

$$A = \begin{bmatrix} \xi_1 & \xi_1^2 & \xi_1^3 \\ \xi_2 & \xi_2^2 & \xi_2^3 \\ \vdots & \vdots & \vdots \\ \xi_N & \xi_N^2 & \xi_M^3 \end{bmatrix}, \qquad x = \begin{bmatrix} a \\ b \\ c \end{bmatrix}. \qquad b = \begin{bmatrix} y_1 \\ y_2 \\ \vdots \\ y_M \end{bmatrix}.$$

Jeśli węzły $x_1, \ldots, x_M$ są różne, macierz A jest pełnego rzędu. Ponieważ macierz jest stosunkowo mała, można rozwiązać układ równań normalnych $A^T Ax = A^T b$ (o ile węzły są dostatecznie daleko od siebie). Można też zrobić rozkład QR macierzy A na przykład metodą Householdera. Koszt jest rzędu $O(MN^2)$ gdzie $N = 3$, czyli liniowy względem M. □

2. Niech $A \in \mathbb{R}^{M\times N}$. Jaki jest związek LZNK z równaniem z (kwadratową) tzw. macierzą rozszerzoną:

$$\begin{bmatrix} \alpha I & A \\ A^T & 0 \end{bmatrix} \begin{bmatrix} r \\ x \end{bmatrix} = \begin{bmatrix} b \\ 0 \end{bmatrix} ?$$

Przyjmij, że $\alpha \neq 0$.

Rozw. Rozpisując mamy

$$\begin{cases} \alpha r - Ax &= b \\ A^T r &= 0 \end{cases}$$

Wyznaczając z pierwszego równania $\alpha r = b - Ax$ i podstawiając do drugiego mamy (po uproszczeniu przez α) układ równań normalnych dla x. Parametr α może poprawić uwarunkowanie macierzy rozszerzonej. □

3. Wykaż, że jeśli $A \in \mathbb{R}^{M\times N}$, a U, V są (kwadratowymi) macierzami ortogonalnymi odpowiedniego rozmiaru, to $\|UA\|_2 = \|A\|_2$ oraz $\|AV\|_2 = \|A\|_2$.

Rozw. Z definicji i faktu, że U jest izometrią,

$$\|UA\|_2 = \sup_{\|x\|_2=1} \|UAx\|_2 = \sup_{\|x\|_2=1} \|Ax\|_2 = \|A\|_2.$$

Podobnie

$$\|AV\|_2 = \sup_{\|x\|_2=1} \|AVx\|_2 = \sup_{\|Vx\|_2=1} \|A\underbrace{Vx}_{=y}\|_2 = \sup_{\|y\|_2=1} \|Ay\|_2 = \|A\|_2.$$

Ponieważ macierz V jest nieosobliwa, $\{y = Vx : x \in \mathbb{R}^N\} = \mathbb{R}^N$. □

2.2. Danie główne

4. Niech $\alpha > 0$. *Zregularyzowane* zadanie najmniejszych kwadratów:

$$||b - Ax||_2^2 + \alpha||x||_2^2 \to \min!$$

ma jednoznaczne rozwiązanie nawet wtedy, gdy A nie jest pełnego rzędu. Aby to udowodnić, zapisz je jako standardowe zadanie najmniejszych kwadratów dla pewnej macierzy B i prawej strony f i wykaż, że B jest pełnego rzędu.

Dla LZNK z macierzą B sformułuj układ równań normalnych w terminach danych oryginalnego zadania: A, α, b i uzasadnij wprost, że jego macierz jest symetryczna i dodatnio określona.

Rozw.

$$||b - Ax||_2^2 + \alpha||x||_2^2 = ||b - A\cdot x||_2^2 + ||0 - \sqrt{\alpha}I \cdot x||_2^2 = ||\begin{bmatrix} b \\ 0 \end{bmatrix} - \begin{bmatrix} A \\ \sqrt{\alpha}I \end{bmatrix} \cdot x||_2^2 \equiv ||f - Bx||_2^2$$

(ostatnia równość to tw. Pitagorasa wykorzystywane już wcześniej np. w konstrukcji algorytmu rozwiązywania LZNK przez rozkład QR). Macierz B jest pełnego rzędu, gdyż jej dolny blok to macierz diagonalna.

Jest to więc zadanie LZNK $||f - Bx||_2 \to \min!$, a jego układ równań normalnych to $B^T Bx = B^T f$, przy czym

$$B^T B = A^T A + \alpha I, \qquad B^T f = A^T b.$$

□

5. Niech A będzie macierza pełnego rzędu rozmiaru $M \times N$, gdzie $M \geq N$ i przypuśćmy, że znamy jej wąski rozkład QR: $A = Q_1 R_1$, gdzie Q_1 jest prostokątna takiego samego rozmiaru jak A oraz $Q_1^T Q_1 = I$ i ponadto R_1 jest kwadratowa $N \times N$ trójkątna górna.

Często statystyków interesuje tzw. macierz kowariancji, $C = (A^T A)^{-1}$. Pokaż, jak kosztem rzędu $4N^3$ obliczyć jej diagonalę. *Wskazówka. Nie obliczaj macierzy odwrotnej do $A^T A$.*

Rozw. Zauważmy, że $c_{ii} = e_i^T C e_i = e_i^T w_i$, gdzie $A^T A w_i = e_i$, więc należy N razy rozwiązać układ równań z macierza $A^T A$. Ale na mocy rozkładu, $A^T A = R_1^T Q_1^T Q_1 R_1 = R_1^T R_1$, więc od razu dysponujemy rozkładem Cholesky'ego $A^T A$! Koszt rozwiązania układu z macierzą trójkątną kosztuje około $2N^2$ flopów (bo na każdy element macierzy R_1 przypada jedno dodawanie i jedno mnożenie). To oznacza, że oba równania (z R_1^T i R_1) rozwiążemy kosztem $\mathscr{O}(4N^2)$, więc koszt rozwiązania z N prawymi stronami będzie N razy większy. □

6. Dana jest $Q = H_1 \cdots H_N$, gdzie H_i, $i = 1, \ldots, N$, są macierzami Householdera $M \times M$ reprezentowanymi za pomocą wektorów $\tilde{v}_i \in \mathbb{R}^{M-i+1}$, otrzymanych w algorytmie wyznaczania rozkładu QR macierzy $A \in \mathbb{R}^{M \times N}$. Podaj możliwie tani algorytm obliczania Qz, gdzie $z = \begin{bmatrix} z_N \\ 0 \end{bmatrix}$ i $z_N \in \mathbb{R}^N$.

Rozw. Jak pamiętamy, $H_i = I - \gamma_i v_i v_i^T$, gdzie $v_i = \begin{bmatrix} 0 \\ \tilde{v}_i \end{bmatrix}$. Dla *dowolnego* wektora $z \in \mathbb{R}^M$ koszt pomnożenia $H_i z$ jest liniowy w M — dokładniej, $\mathrm{Const} \cdot (M - i + 1)$ (bo musimy działać tylko na ostatnich współrzędnych). Zatem obliczenie $H_1(H_2 \cdots (H_{N-1}(H_N z)) \cdots)$ będzie kosztować $\mathscr{O}(MN)$, a dokładniej

$$\sum_{i=1}^{N} \mathrm{Const} \cdot (M - i + 1) = \mathrm{Const}\, M + (M-1) + \ldots + (M - N + 1) = \mathrm{Const} \cdot (MN - N(N-1)/2).$$

□

7. Wykaż, że wąski rozkład QR macierzy A pełnego rzędu rozmiaru $M \times N$, gdzie $M \geq N$:

$$A = Q_1 R_1,$$

gdzie Q_1 jest prostokątna takiego samego rozmiaru jak A oraz $Q_1^T Q_1 = I$ i ponadto R_1 jest kwadratowa trójkątna górna i na diagonali ma liczby dodatnie — jest jednoznaczny.

Rozw. Załóżmy przeciwnie, że $A = Q_1 R_1 = Q_2 R_2$, skąd $Q_2^T Q_1 = R_2 R_1^{-1}$. Macierz po prawej, $R_2 R_1^{-1}$, jest górna trójkątna (jako iloczyn górnych trójkątnych) i ortogonalna (bo równa $Q_2^T Q_1$), skąd natychmiast (przez indukcję po kolejnych kolumnach) wynika, że musi być diagonalna. Macierz ortogonalna i diagonalna musi mieć na diagonali ± 1, oznaczmy ją $\mathscr{I}$. Stąd $R_2 = \mathscr{I} R_1$ więc na diagonali zachodzi $r_{ii}^{(1)} = \pm r_{ii}^{(2)}$. Z założenia stałości znaku wynika więc, że „$\pm$" = „$+$" i w konsekwencji $\mathscr{I} = I$, więc $R_1 = R_2$. Na koniec dostajemy, że $Q_2^T Q_1 = I$, czyli $Q_1 = Q_2$. □

8. Niech $A^T \in \mathbb{R}^{N\times M}$, przy czym $M \geq N$ (czyli A^T ma więcej kolumn niż wierszy) i przypuśćmy, że A jest pełnego rzędu. Wyznacz rozwiązanie $x \in \mathbb{R}^M$ układu równań

$$A^T x = b$$

o najmniejszej normie $\|\cdot\|_2$. Jaki jest koszt w zależności od N, M?

Rozw. Rząd macierzy A jest równy M. Istnieje więc rozkład QR macierzy A

$$A = QR, \qquad R = \begin{bmatrix} R_1 \\ 0 \end{bmatrix},$$

gdzie $Q \in \mathbb{R}^{M\times M}$ jest ortogonalna, $Q^T Q = I$, natomiast R_1 jest górna trójkątna $N \times N$ i nieosobliwa (bo A^T jest pełnego rzędu). Zatem ma być $R^T Q^T x = b$ i oznaczając $\begin{bmatrix} z_1 \\ z_2 \end{bmatrix} = Q^T x$ gdzie $z_1 \in \mathbb{R}^N$ mamy

$$\begin{bmatrix} R_1^T & 0^T \end{bmatrix} \begin{bmatrix} z_1 \\ z_2 \end{bmatrix} = b \in \mathbb{R}^N,$$

skąd $R_1^T z_1 = b$ (które wystarczy rozwiązać kosztem $\mathcal{O}(N^2)$, przy czym z_2 jest dowolne. Ponieważ Q jest ortogonalna, to $\|x\|_2^2 = \|z\|_2^2 = \|z_1\|_2^2 + \|z_2\|_2^2 = \|R_1^{-T} b\|_2^2 + \|z_2\|_2^2$. Minimum tej wartości jest dla $z_2 = 0$. A więc koszt obliczenia x to standardowy koszt wyznaczenia rozkładu QR macierzy A, równy $\mathcal{O}(MN^2)$ ze stałą zależną od wyboru konkretnego algorytmu, plus koszt rozwiązania $R_1^T z_1 = b$ równy $\mathcal{O}(N^2)$, plus koszt obliczenia $x = Q \begin{bmatrix} z_1 \\ 0 \end{bmatrix} = \begin{bmatrix} Q_1 & Q_2 \end{bmatrix} \begin{bmatrix} z_1 \\ 0 \end{bmatrix} = Q_1 z_1$. Gdyby dysponować Q_1, to ten ostatni krok kosztuje $\mathcal{O}(NM)$. □

2.3. Deser — nieobowiązkowy!

To są zadania o tematyce nieco pobocznej, dla chętnych i zainteresowanych.

9. Rozkładając macierz A na dwa mniej więcej równe bloki, można zapisać jej rozkład QR blokowo

$$\begin{bmatrix} A_1 & A_2 \end{bmatrix} = \begin{bmatrix} Q_1 & Q_2 \end{bmatrix} \begin{bmatrix} R_{11} & R_{12} \\ & R_{22} \end{bmatrix},$$

gdzie $Q_1 R_{11}$ to rozkład QR macierzy A_1. Wykorzystując tę obserwację, sformułuj *rekurencyjny* algorytm wyznaczenia rozkładu QR macierzy A i oszacuj jego koszt. Dla uproszczenia przyjmij, że A ma liczbę kolumn, która jest potęgą dwójki.

10. Wyznacz wskaźnik uwarunkowania w normie $\|\cdot\|_2$ macierzy rozszerzonej z zadania 2 w zależności od $\alpha > 0$ i wartości szczególnych A. Przyjmij, że A jest pełnego rzędu.

Matematyka obliczeniowa: ćw. zdalne 2020-04-22
Piotr Krzyżanowski

---

1. Polecana literatura

Zajęcia dotyczą mniej więcej stron 110... 120 skryptu Wykładowcy, P. Kiciaka — tym razem skupimy się na własnościach i zastosowaniach SVD. Oczywiście, każdy szanujący się podręcznik z metod numerycznych (lub uczenia maszynowego) ma rozdziały o *singular value decomposition*.

2. Zadania z rozwiązaniami

Namawiam Państwa do próby samodzielnego rozwiązania każdego z zadań, a dopiero potem do przeczytania gotowca. O ile nie będzie powiedziane inaczej, zawsze będziemy zakładać, że obliczenia są wykonywane w arytmetyce dokładnej.

2.1. Przystawki

0. *Najpierw przeczytaj polecaną literaturę!*

1. Niech $U\Sigma V^T$ będzie rozkładem SVD macierzy prostokątnej A rozmiaru $M \times N$, gdzie $M \geq N$. Wykaż, że rząd A jest równy k: liczbie niezerowych wartości szczególnych A, tzn. $\sigma_1 \geq \ldots \geq \sigma_k > \sigma_{k+1} = \ldots = \sigma_N = 0$. Dokładniej, $\ker(A) = \mathrm{lin}\{v_{k+1}, \ldots, v_N\}$ oraz $\mathrm{range}(A) = \mathrm{lin}\{u_1, \ldots, u_k\}$.

Rozw. Rzeczywiście, ponieważ $A = \sum_{i=1}^{k} \sigma_k u_i v_i^T$, to dla $j > k$ mamy $Av_j = \sum_{i=1}^{k} \sigma_k u_i \underbrace{(v_i^T v_j)}_{=0} = 0$, a dla $j \leq k$ mamy $Av_j = \sum_{i=1}^{k} \sigma_k u_i \underbrace{(v_i^T v_j)}_{=\delta_{ij}} = \sigma_j u_j$. Stąd teza. □

2. Wykaż, że jeśli $A = A^T$ oraz $A = U\Lambda U^T$ jest rozkładem własnym macierzy A przy czym $U^T U = I$, to SVD dla A jest postaci $A = U\Sigma V^T$, gdzie $\sigma_i = |\lambda_i|$ oraz $v_i = \mathrm{sign}(\lambda_i) u_i$.

Rozw. Dowód wynika wprost z definicji SVD. □

3. Niech $U\Sigma V^T$ będzie rozkładem SVD macierzy prostokątnej A rozmiaru $M \times N$, gdzie $M \geq N$. Wartości własne $A^T A$ są równe σ_i^2, a wektory własne $A^T A$ to v_i.

Rozw. $A^T A = V\Sigma^T \Sigma U^T U \Sigma V^T = V \begin{bmatrix} \Sigma_1 & 0^T \end{bmatrix} \begin{bmatrix} \Sigma_1 \\ 0 \end{bmatrix} V^T = V\Sigma_1^2 V^T$ i to jest rozkład własny macierzy $A^T A$. □

4. Niech $U\Sigma V^T$ będzie rozkładem SVD macierzy prostokątnej A rozmiaru $M \times N$, gdzie $M \geq N$. Wartości własne AA^T to σ_i^2 oraz $M-N$-krotne zero, a wektory własne AA^T odpowiadające σ_i^2 to u_i.

Rozw.

$$AA^T = U\Sigma V^T V\Sigma^T U^T = U\Sigma\Sigma^T U^T = \begin{bmatrix} U_1 & U_2 \end{bmatrix} \begin{bmatrix} \Sigma_1^2 & 0 \\ 0 & 0 \end{bmatrix} \begin{bmatrix} U_1 & U_2 \end{bmatrix}^T$$

— i to jest rozkład własny macierzy AA^T. □

5. Niech $U_1\Sigma_1 V^T$ będzie wąskim rozkładem SVD macierzy pełnego rzędu A rozmiaru $M \times N$, gdzie $M \geq N$. Rozwiązanie LZNK $\|b - Ax\|_2 \to \min!$ jest dane wzorem $x = V\Sigma_1^{-1} U_1^T b$.

Rozw. Ponieważ $A^T Ax = A^T b$, to podstawiając SVD (zob. zadanie 3) mamy

$$V\Sigma_1^2 V^T x = V \underbrace{\Sigma^T U^T}_{=(U\Sigma)^T=(U_1\Sigma_1)^T=\Sigma_1^T U_1^T} b.$$

Mnożąc stronami kolejno przez V^T (ortogonalną) a potem przez Σ_1^{-1} (nieosobliwą, bo A pełnego rzędu), dostajemy tezę. □

2.2. Danie główne

6. Wykaż, że jeśli $A \in \mathbb{R}^{M\times N}$, to $\|A\|_2 = \sigma_1$ oraz $\|A\|_F = \sqrt{\sigma_1^2 + \ldots \sigma_N^2}$, gdzie $\sigma_1 \geq \ldots \sigma_N \geq 0$ są wartościami szczególnymi A.

Rozw. Niech $A = U\Sigma V^T$ będzie rozkładem SVD macierzy A. Ponieważ mnożenie przez macierz ortogonalną nie zmienia ani normy spektralnej macierzy, ani Frobeniusa, to $\|A\|_2 = \|U\Sigma V^T\|_2 = \|\Sigma\|_2 = \max_i \sigma_i = \sigma_1$ i podobnie $\|A\|_F^2 = \|U\Sigma V^T\|_F^2 = \|\Sigma\|_F^2 = \sum_i \sigma_i^2$. □

7. Niech A będzie macierzą $M \times N$, gdzie $M \geq N$ i niech $1 \leq k \leq N$. Wykaż, że najlepsza aproksymacja A w normie $\|\cdot\|_2$ macierzą rzędu co najwyżej k jest postaci

$$A_k = \sum_{i=1}^{k} \sigma_i u_i v_i^T,$$

gdzie $A = U\Sigma V^T = \sum_{i=1}^{N} \sigma_i u_i v_i^T$ jest rozkładem SVD macierzy A oraz $\sigma_1 \geq \sigma_2 \geq \ldots \sigma_N \geq 0$.

Rozw. Z definicji, A_k jest rzędu co najwyżej k, bo ma co najwyżej k niezerowych wartości szczególnych. Ponadto

$$\|A - A_k\|_2 = \|\sum_{i=k+1}^{N} \sigma_i u_i v_i^T\|_2 = \|U \begin{bmatrix} 0 & & & & \\ & \ddots & & & \\ & & \sigma_{k+1} & & \\ & & & \ddots & \\ & & & & \sigma_N \end{bmatrix} V^T\|_2 = \|\begin{bmatrix} 0 & & & & \\ & \ddots & & & \\ & & \sigma_{k+1} & & \\ & & & \ddots & \\ & & & & \sigma_N \end{bmatrix}\|_2 = \sigma_{k+1}.$$

Wystarczy pokazać, że nie da się lepiej. Niech B będzie dowolną macierzą rzędu co najwyżej k, zatem jej jądro jest przestrzenią liniową wymiaru co najmniej $N-k$. Ponieważ przestrzeń rozpięta przez

$v_1, \ldots, v_{k+1}$ jest wymiaru $k+1$, to obie te przestrzenie muszą mieć nietrywialną część wspólną. Niech $h \neq 0$ będzie takim wektorem i załóżmy od razu, że jest unormowany, $\|h\|_2 = 1$.

Wtedy

$$\|A-B\|_2 = \sup_{\|x\|_2=1} \|(A-B)x\|_2 \geq \|(A-B)h\|_2 = \|Ah\|_2 = \|U\Sigma V^T h\|_2$$
$$= \|\Sigma V^T h\|_2 \geq \sigma_{k+1}\|V^T h\|_2 = \sigma_{k+1}\|h\|_2 = \sigma_{k+1}.$$

□

8. Jak zachowują się rozwiązania zregularyzowanego zadania najmniejszych kwadratów

$$\|b-Ax\|_2^2 + \lambda\|x\|_2^2 \to \min!$$

gdy $\lambda \to 0$? A gdy $\lambda \to \infty$?

Rozw. Załóżmy, że rząd macierzy A jest równy $k \leq N$. Skorzystajmy z rozkładu SVD macierzy $A = U\Sigma V^T$ i podstawmy go do układu równań normalnych dla zadania zregularyzowanego, $(A^TA + \lambda I)x = A^Tb$:

$$(\Sigma^T\Sigma + \lambda I)\underbrace{V^Tx}_{=y} = \Sigma^T U^T b$$

skąd

$$\begin{bmatrix} \sigma_1^2+\lambda & & & \\ & \ddots & & \\ & & \sigma_N^2+\lambda \end{bmatrix} y = \begin{bmatrix} \sigma_1 & & & \\ & \ddots & & \\ & & \sigma_k & \\ & & & 0 \end{bmatrix} \begin{bmatrix} u_1^Tb \\ \vdots \\ u_k^Tb \\ u_N^Tb \end{bmatrix},$$

czyli

$$x = x(\lambda) = \sum_{i=1}^{k} \left(\frac{\sigma_i}{\sigma_i^2+\lambda} u_i^T b \right) v_i.$$

W szczególności, gdy $\lambda \to 0$, to $x(\lambda)$ dąży do $\sum_{i=1}^{k} \left(\sigma_i u_i^T b\right) v_i$ — rozwiązania nieregularnego zadania najmniejszych kwadratów o minimalnej normie (por. Wykład). A gdy $\lambda \to \infty$, to (jak można było się od razu spodziewać) wtedy $x(\lambda) \to 0$. □

2.3. Deser — nieobowiązkowy!

To są zadania o tematyce nieco pobocznej, dla chętnych i zainteresowanych.

9. Niech A będzie macierzą $M \times N$, gdzie $M \geq N$ i niech $1 \leq k \leq N$. Wykaż, że najlepsza aproksymacja A, w normie $\|\cdot\|_F$, macierzą rzędu co najwyżej k jest postaci

$$A_k = \sum_{i=1}^{k} \sigma_i u_i v_i^T,$$

gdzie $A = U\Sigma V^T = \sum_{i=1}^{N} \sigma_i u_i v_i^T$ jest rozkładem SVD macierzy A oraz $\sigma_1 \geq \sigma_2 \geq \ldots \sigma_N \geq 0$.

Matematyka obliczeniowa: ćw. zdalne 2020-04-29
Piotr Krzyżanowski

Zajęcia dotyczą stron 130... 145 skryptu Wykładowcy, P. Kiciaka — czyli zagadnienia własnego. Ponadto każdy szanujący się podręcznik z metod numerycznych ma rozdziały o *eigenvalue problem*.

1. Zadania z rozwiązaniami

Namawiam Państwa do próby samodzielnego rozwiązania każdego z zadań, a dopiero potem do przeczytania gotowca. O ile nie będzie powiedziane inaczej, zawsze będziemy zakładać, że obliczenia są wykonywane w arytmetyce dokładnej.

1.1. Przystawki

1. Wykaż, że wszystkie wartości własne macierzy A leżą w kole

$$K = \{z \in \mathbb{C} : |z| \leq \|A\|\},$$

gdzie $\|\cdot\|$ jest dowolną normą indukowaną.

Rozw. Trywialne. Z definicji $Ax = \lambda x$ i (po unormowaniu) $\|x\| = 1$, skąd $\|Ax\| = |\lambda| \cdot \|x\| = |\lambda|$, a z drugiej strony $\|Ax\| \leq \|A\| \cdot \|x\| = \|A\|$. □

2. (Twierdzenie Gerszgorina). Wartości własne macierzy A leżą w $\bigcup_{i=1}^{N} K_i$, gdzie

$$K_i = \{z \in \mathbb{C} : |z - a_{ii}| \leq \sum_{j \neq i} |a_{ij}|\}.$$

Rozw. Niech $\lambda \in \mathbb{C}$ będzie wartością własną i niech x taki, że $\|x\|_\infty = 1$, będzie odpowiadającym jej wektorem własnym. Z definicji normy maksimum istnieje współrzędna, x_i, wektora x taka, że $\|x\|_\infty = \max_j |x_j| = |x_i|$. Mamy $Ax = \lambda x$, więc na i–tej współrzędnej zachodzi

$$\sum_j a_{ij} x_j = \lambda x_i \qquad \Longleftrightarrow \quad (a_{ii} - \lambda) x_i = \sum_{j \neq i} a_{ij} x_j.$$

Obkładając modułami, $|a_{ii} - \lambda| \underbrace{|x_i|}_{=1} \leq \sum_{j \neq i} |a_{ij}| \underbrace{|x_j|}_{\leq 1}$, czyli $\lambda \in K_i$. □

3. Wartości własne A i XAX^{-1} są takie same. W szczególności dla macierzy ortogonalnej Q wartości własne A i QAQ^T są takie same.

Rozw. Niech (λ, x) będzie parą własną A, tzn. $Ax = \lambda x$. Jest tak wtedy i tylko wtedy, gdy $AX^{-1}Xx = \lambda X^{-1}Xx$, a to wtw, gdy $XAX^{-1}y = \lambda y$ dla $y = X^{-1}x$. Zatem wartości własne są takie same, a odpowiadające im wektory własne spełniają zależność $y = X^{-1}x$. □

4. Wyznacz wartości własne i wartość wyznacznika przekształcenia Householdera $I - 2vv^T/v^Tv$.

Rozw. Skoro jest to przekształcenie ortogonalne, to wszystkie wartości własne mają moduły równe 1, a skoro jest symetryczne rzeczywiste, to są też rzeczywiste. A tak w ogóle, to to zadanie wcześniej już robiliśmy — w ogólniejszej postaci. □

1.2. Danie główne

5. Sprowadź serią przekształceń ortogonalnych $Q_2, Q_3, \ldots$ macierz kwadratową A rozmiaru $N \times N$ do postaci Hessenberga:

$$A \to Q_2 A Q_2^T =: A_2 \to Q_3 A_2 Q_3^T =: A_3 \to \cdots \to \textbf{Hessenberga}.$$

Oszacuj koszt w zależności od N. Taka macierz będzie miała to samo spektrum, co A. A jakie wektory własne?

Rozw. Niech Q_2 będzie oparta na przekształceniu Householdera i taka, że

$$Q_2 a_1 = Q_2 \begin{bmatrix} a_{11} \\ a_{21} \\ a_{31} \\ \vdots \\ a_{N1} \end{bmatrix} = \begin{bmatrix} a_{11} \\ \star \\ 0 \\ \vdots \\ 0 \end{bmatrix}$$

Innymi słowy, jeśli weźmiemy „standardowe” przekształcenie Householdera H_1 rozmiaru $(N-1) \times (N-1)$ takie, że

$$H_1 \begin{bmatrix} a_{21} \\ a_{31} \\ \vdots \\ a_{N1} \end{bmatrix} = \begin{bmatrix} \star \\ 0 \\ \vdots \\ 0 \end{bmatrix} \qquad \text{to wtedy} \quad Q_2 = \begin{bmatrix} 1 & \\ & H_1 \end{bmatrix}.$$

Oczywiście,

$$Q_2 A = \begin{bmatrix} a_{11} & \star & \cdots & \star \\ \star & \star & \cdots & \star \\ 0 & \star & \cdots & \star \\ \vdots & \star & \cdots & \star \end{bmatrix}, \quad \text{skąd} \quad A_2 = Q_2 A Q_2^T = Q_2 A Q_2 = \begin{bmatrix} a_{11} & \tilde{\star} & \cdots & \tilde{\star} \\ \star & \tilde{\star} & \cdots & \tilde{\star} \\ 0 & \tilde{\star} & \cdots & \tilde{\star} \\ \vdots & \tilde{\star} & \cdots & \tilde{\star} \end{bmatrix},$$

czyli mnożenie przez Q_2^T z *prawej* strony nie rusza pierwszej kolumny. Dalej powtarzamy ten sam schemat dla podmacierzy $A_2(2:N, 2:N)$, tzn. konstruujemy $H_3 \in \mathbb{R}^{(N-2)\times(N-2)}$ takie, że

$$H_3 \begin{bmatrix} a_{32}^{(2)} \\ a_{43}^{(2)} \\ \vdots \\ a_{N3}^{(2)} \end{bmatrix} = \begin{bmatrix} \star \\ 0 \\ \vdots \\ 0 \end{bmatrix} \qquad \text{i bierzemy} \qquad Q_3 = \begin{bmatrix} I_2 & \\ & H_3 \end{bmatrix},$$

itd.

Jeśli chodzi o koszt, to konstrukcja Q_i oraz ich pomnożenie przez A z lewej strony kosztuje w zasadzie tyle samo, co w algorytmie rozkładu QR (nie przykładamy tylko pierwszego przekszt. Householdera i ostatniego) — czyli $\mathcal{O}(\frac{4}{3}N^3)$ flopów. Dodatkowo, musimy jeszcze pomnożyć macierz A z prawej strony, co mnoży główną część kosztu przez dwa. □

6. Sprowadź macierz A symetryczną przekształceniami jak w zadaniu 5 do postaci trójdiagonalnej. Jaki będzie koszt tego algorytmu?

Rozw. Było na wykładzie! Zastosujmy algorytm z poprzedniego zadania, który sprowadza dowolną macierz do postaci Hessenberga. Ale zauważmy, że jeśli $A = A^T$, to także QAQ^T jest symetryczna. W konsekwencji wynik działania algorytmu: macierz Hessenberga, też jest symetryczna — jak to zrobić dobrze, proszę zobaczyć w skrypcie. Koszt będzie z grubsza o połowę niższy niż w zadaniu 5, bo wystarczy prowadzić działania w jednym (np. górnym) trójkącie macierzy A (bo z symetrii $a_{ij} = a_{ji}$) — czyli $\mathcal{O}(\frac{4}{3}N^3)$ flopów. □

7. Macierz stowarzyszona z wielomianem $p(x) = a_N x^N + \ldots a_1 x + a_0$ to macierz

$$C = \begin{bmatrix} -\frac{a_{N-1}}{a_N} & -\frac{a_{N-2}}{a_N} & \cdots & -\frac{a_0}{a_N} \\ 1 & & & \\ & \ddots & & \\ & & 1 & \end{bmatrix}$$

Udowodnij, że $\{\lambda \in \mathbb{C} : \lambda \text{ jest w. wł } C\} = \{x \in \mathbb{C} : p(x) = 0\}$. Jak ten fakt wykorzystać do wyznaczenia wszystkich pierwiastków p?

Rozw. Załóżmy, że wielomian już podzieliliśmy przez $a_N \neq 0$, więc jest on już postaci $p(x) = x^N + a_{N-1}x^{N-1} + \ldots a_1 x + a_0$, więc macierz stowarzyszona upraszcza się do

$$C = \begin{bmatrix} -a_{N-1} & -a_{N-2} & \cdots & -a_0 \\ 1 & & & \\ & \ddots & & \\ & & 1 & \end{bmatrix}.$$

Liczba $x \in \mathbb{C}$ jest wartością własną C wtedy i tylko wtedy, gdy $\det(C - xI) = 0$, więc istnieje niezerowy wektor v taki, że $(C - xI)v = 0$:

$$\begin{bmatrix} -a_{N-1} - x & -a_{N-2} & \cdots & -a_0 \\ 1 & -x & & \\ & \ddots & \ddots & \\ & & 1 & -x \end{bmatrix} \begin{bmatrix} v_{N-1} \\ \vdots \\ v_1 \\ v_0 \end{bmatrix} = \begin{bmatrix} 0 \\ \vdots \\ 0 \\ 0 \end{bmatrix}.$$

Zauważmy, że w szczególności musi zachodzić

$$v_{i+1} = x v_i, \qquad \text{dla } i = 0, \ldots, N-1,$$

skąd $v_0 \neq 0$. Unormujmy więc v w taki sposób, by $v_0 = 1$. Wtedy $v_i = x^i$ i pierwsze równanie daje nam zależność

$$(-a_{N-1} - x)x^{N-1} - a_{N-2}x^{N-2} - \ldots - a_0 = 0 \qquad \Longleftrightarrow \qquad -p(x) = 0.$$

Zatem x jest wartością własną $C \iff$ gdy jest miejscem zerowym p.

Oczywiście, pomysł na wyznaczenie pierwiastków p polega na wyznaczeniu wszystkich wartości własnych C, np. jakąś mocną wersją metody QR. Skądinąd MATLAB właśnie tak implementuje polecenie **roots**. □

8. Wykaż, że jeżeli u, v są niezerowymi wektorami, to metoda potęgowa dla $A = uv^T$ zbiegnie (pod pewnym drobnym warunkiem) w jednej iteracji (do czego?).

Rozw. Jak wiadomo, macierz uv^T jest rzędu co najwyżej 1 i pokazaliśmy wcześniej, że jej wartościami własnymi są: zero, krotności geometrycznej $(N-1)$ (przestrzeń własna jej odpowiadająca, nazwijmy ją Z, jest rozpięta przez wektory prostopadłe do v), oraz $\lambda_1 = v^T u$ (z wektorem własnym u). Jeśli $v^T u \neq 0$, to λ_1 jest krotności 1. Zakładając więc, że

— $\lambda_1 = v^T u \neq 0$ (i to jest ów „drobny warunek" z treści zadania),
— x_0 ma niezerową składową w kierunku u,

dostajemy, że baza Z i u rozpinają całą przestrzeń, więc $x_0 = \alpha z + \beta u$ i $\beta \neq 0$, skąd

$$Ax_0 = \alpha \underbrace{Az}_{=0} + \beta \underbrace{Au}_{=\lambda_1 u} = \mathrm{Const}\, u.$$

Czyli zbieżność w jednej iteracji do (kierunku) wektora własnego u, odpowiadającego (dominującej, jako jedynej niezerowej!) wartości własnej $\lambda_1 = v^T u \neq 0$. □

9. Co zrobić, jeśli przesunięcie μ w metodzie Wielandta jest *dokładną* wartością własną, a tym samym macierz $A - \mu I$ nie jest odwracalna?

Rozw. Wystarczy ją ciutkę zaburzyć, $\tilde{\mu} = \mu(1+\varepsilon)$. Wtedy dalej jesteśmy najbliżej wartości własnej (no, chyba, że jest inna, patologicznie bliska μ), więc iteracja szybko zbiegnie. □

1.3. Deser — nieobowiązkowy!

To są zadania o tematyce nieco pobocznej, dla chętnych i zainteresowanych.

10. „Numeryczne kryterium numerycznej poprawności". Wykaż, że dla każdej pary λ, v („kandydatów na parę własną macierzy A") (przy czym zakładamy $\|v\|_2 = 1$) istnieje macierz E taka, że

$$(A+E)v = \lambda v \qquad \text{oraz} \qquad \|E\|_2 \leq \|r\|_2,$$

gdzie $r = Av - \lambda x$.

11. Niech będą dane współczynniki $b_0, \ldots, b_N \in \mathbb{R}$ wielomianu p w bazie Newtona związanej z (parami różnymi) węzłami $x_0, \ldots, x_N \in \mathbb{R}$. Wskaż macierz, której współczynniki zależą wprost od $\{b_i\}$ i której spektrum jest tożsame ze zbiorem wszystkich pierwiastków (być może zespolonych) wielomianu p.

Matematyka obliczeniowa: ćw. zdalne 2020-05-06
Piotr Krzyżanowski

---

1. Polecana literatura

Zajęcia dotyczą stron 145...160 skryptu Wykładowcy, P. Kiciaka — czyli interpolacji wielomianowej. Ten temat jest też omówiony w znanej już Państwu książeczce G.W. Stewarta *Afternotes on Numerical Analysis* (której tak bardzo nam brakowało przez ostatnie tygodnie...) Ponadto każdy szanujący się podręcznik z metod numerycznych ma rozdziały o *polynomial interpolation*.

2. Zadania z rozwiązaniami

Namawiam Państwa do próby samodzielnego rozwiązania każdego z zadań, a dopiero potem do przeczytania gotowca. O ile nie będzie powiedziane inaczej, zawsze będziemy zakładać, że obliczenia są wykonywane w arytmetyce dokładnej.

2.1. Przystawki

0. *Najpierw przeczytaj polecaną literaturę!*

1. Ile kosztuje algorytm Hornera wyznaczania wartości wielomianu w punkcie x, gdy wielomian jest zadany w bazie Newtona?

Rozw. Koszt jest liniowy. Wyprowadzenie jest identyczne jak w przypadku standardowego Hornera: $p(x) = b_0 + (x - x_0)$(wielomian stopnia $n-1$) — skąd rekurencja, którą możemy rozwinąć . □

2. Na ilu węzłach należy oprzeć wielomian interpolacyjny Lagrange'a stopnia co najwyżej n?

Rozw. Oczywiście $n+1$, żeby zadanie miało jednoznaczne rozwiązanie. □

3. Ile kosztuje wyznaczenie wielomianu interpolacyjnego w bazie Lagrange'a?

Rozw. Koszt jest zerowy. □

4. Wyznacz wielomian, który interpoluje tabelkę:

x_i	-1	1	0	2
f_i	0	4	1	0

Rozw. Sformułowanie zadania to oczywiście slang. Chodzi o wyznaczenie wielomianu interpolacyjnego Lagrange'a, interpolującego wartości f_i w węzłach x_i. Wyznaczymy go w bazie Newtona. Różnice dzielone:

x_i	f_i			
-1	0			
1	4	$\frac{-4}{-2}=2$		
0	1	3	1	
2	0	$-1/2$	$\frac{-1/2-3}{2-1}=-7/2$	$\frac{-7/2-1}{2-(-1)}=-3/2$

Stąd $p(x)=0\cdot 1+2\cdot(x+1)+1\cdot(x+1)(x-1)-\frac{3}{2}\cdot(x+1)(x-1)x$. Kto chce, może rozpisać go w bazie naturalnej. □

5. Zapisz pseudokod, wyznaczający wielomian interpolacyjny Lagrange'a w bazie Newtona.

Rozw. Na wejściu mamy tablicę węzłów $x_0,\ldots,x_N$ i wartości $y_0=f(x_0),\ldots,y_N=f(x_N)$. Na pozycji ij, gdzie $j=0,\ldots,N$ oraz $i=j,\ldots,N$, w tabelce jest $T_{ij}=f[x_{i-j},\ldots,x_i]$ i z definicji

$$T_{ij}=\frac{f[x_{i-j},\ldots,x_{i-1}]-f[x_{i-j+1},\ldots,x_i]}{x_{i-j}-x_i}=\frac{\overbrace{f[x_{(i-1)-(j-1)},\ldots,x_{(i-1)}]}^{T_{i-1,j-1}}-\overbrace{f[x_{i-(j-1)},\ldots,x_i]}^{T_{i,j-1}}}{x_{i-j}-x_i},$$

skąd algorytm:

for $j=1:N$ ▷ wyznaczamy kolejne kolumny tabelki
  for $i=N:-1:j$ ▷ elementy kolumny od dołu o góry
    $T_{ij}=(T_{i-1,j-1}-T_{i,j-1})/(x_{i-j}-x_i)$ ▷ patrz opis powyżej
  end
end

Jeśli będziemy wyznaczać tabelkę T różnic dzielonych kolumna po kolumnie jak powyżej, to wystarczy, że będziemy pamiętać jedynie ostatnio wyznaczoną. Jeśli w elementy w ustalonej kolumnie będziemy obliczać od ostatniego — jak już zrobiliśmy przewidująco powyżej — to będziemy mogli go zapisywać w tamtej komórce — jej stara wartość nie przyda się już do obliczania wyrazów w tej kolumnie.

for $j=1:N$ ▷ wyznaczamy kolejne kolumny tabelki
  for $i=N:-1:j$ ▷ elementy kolumny od dołu o góry
    $y_i=(y_{i-1}-y_i)/(x_{i-j}-x_i)$ ▷ w chwili j po wykonaniu instrukcji jest $y_i=T_{ij}$
  end
end

I już! Na wyjściu y_i to współczynniki wielomianu w bazie Newtona, czyli $p(x)=\sum_{i=0}^{N} y_i\phi_i(x)$. □

6. Wyznacz wielomian interpolacyjny Hermite'a, który interpoluje tabelkę:

x_i	$f(x_i)$	$f'(x_i)$	$f''(x_i)$	$f'''(x_i)$
0	0	1	2	
1	0	2	0	6

Rozw.

Pamiętając, że $f(\underbrace{x_0,\ldots,x_0}_{k+1\text{ razy}})=f^{(k)}(x_0)/k!$, wypełniamy danymi tabelkę różnic dzielonych:

x_i	$f(x_i)$			
0	0			
0	0	$1/1!$		
0	0	$1/1!$	$2/2!$	
1	0			
1	0	$2/1!$		
1	0	$2/1!$	$0/2!$	
1	0	$2/1!$	$0/2!$	$6/3!$

a następnie uzupełniamy brakujące pola stosując standardowy algorytm różnic dzielonych:

x_i	$f(x_i)$						
0	0						
0	0	$1/1!$					
0	0	$1/1!$	$2/2!$				
1	0	0	-1	-2			
1	0	$2/1!$	2	3	5		
1	0	$2/1!$	$0/2!$	-2	-5	10	
1	0	$2/1!$	$0/2!$	$6/3!$	3	8	-2

Zatem wielomian jest postaci $w(x) = 0 + x + x^2 - 2x^3 + 5x^3(x-1) + 10x^3(x-1)^2 - 2x^3(x-1)^3$. □

2.2. Danie główne

7. Wyprowadź wzór na błąd aproksymacji funkcji $f \in C^{N+1}[a,b]$ wielomianem interpolacyjnym opartym na $N+1$ *równoodległych* węzłach w $[a,b]$:

$$\|f-p\|_\infty \leq \frac{h^{N+1}}{4(N+1)}\|f^{N+1}\|_\infty,$$

gdzie h jest odległością między sąsiednimi węzłami.

Rozw. Ponumerujmy węzły tak, że $a = x_0 < x_1 < \ldots < x_N = b$. Ze wzoru na błąd interpolacji, dla dowolnego $x \in [a,b]$

$$|f(x)-p(x)| \leq \frac{\|f^{N+1}\|_\infty}{(N+1)!} \sup_{x\in[a,b]} |\phi_{N+1}(x)|,$$

gdzie $\phi_{N+1}(x) = (x-x_0)\cdots(x-x_N)$ jest kolejnym wielomianem bazy Newtona. Pokażemy więc, że $|\phi_{N+1}(x)| \leq \frac{h^{N+1}}{4}N!$, skąd teza. W tym celu użyjemy indukcji po N. Gdy $N=1$ to prawda, bo

$$\max_{x\in[a,b]} |x-x_0||x-x_1| = \frac{|x_1-x_0|^2}{2} = \frac{h^2}{2}.$$

Zróbmy krok indukcyjny z $N-1$ do N. Nasz x leży albo w $[x_0, x_{N-1}]$, albo w $[x_1, x_N]$. Rozważmy ten pierwszy przypadek (drugi jest identyczny); z założenia indukcyjnego,

$$|\phi_{N+1}(x)| = \overbrace{|x-x_0|\cdots|x-x_{N-1}|}^{\leq h^N(N-1)!/4} \cdot \underbrace{|x-x_N|}_{\leq N\cdot h} \leq \frac{h^{N+1}}{4}N!,$$

co kończy dowód. □

8. Ile *równoodległych* węzłów wystarczy, by wielomian interpolacyjny Lagrange'a (na nich oparty) aproksymował funkcję $f(x) = \sin(2x)$ na odcinku $[-1,1]$ z błędem bezwzględnym nie przekraczającym ε?

Rozw. Zauważmy, że $f^{(k)}(x) = 2^k G(x)$, gdzie $G(x)$ to albo $\pm\sin(x)$, albo $\pm\cos(x)$ (czyli zawsze $\|G\| \leq 1$). Dla węzłów równoodległych

$$||f-w|| \leq \frac{\|f^{(n+1)}\|}{4(n+1)}\left(\frac{2}{n}\right)^{n+1} = \frac{4^n}{(n+1)n^{n+1}}.$$

Wystarczy więc, że

$$\frac{4^n}{(n+1)n^{n+1}} \leq \varepsilon.$$

□

9. Podaj oszacowanie błędu $\|f-p\|$ interpolacji funkcji $f \in C^{n+1}(a,b)$ wielomianem interpolacyjnym Lagrange'a p opartym na węzłach $a = x_0 < \cdots < x_n = b$, jeśli wartości funkcji wykorzystywane do wyznaczenia p są znane z błędem bezwzględnym nie przekraczającym ε.

Rozw. Zatem p jest interpolantem nie dla f, tylko $\tilde{f} = f + h$, oraz $|h(x_i)| \leq \varepsilon$. Niech p^* będzie interpolantem dla dokładnej f.

$$\|f-p\| = \|f-p^*+p^*-p\| \leq \|f-p^*\| + \|p^*-p\|.$$

Zauważmy, że

$$\|p^*-p\| = \|\sum_i f(x_i)l_i(x) - \sum_i (f+h)(x_i)l_i(x)\| = \|\sum_i h(x_i)l_i(x)\| = \sup_{x\in[a,b]} |\sum_i h(x_i)l_i(x)| \leq \varepsilon \cdot \sup_{x\in[a,b]} \sum_i |l_i(x)| = \varepsilon\cdot\Lambda_n,$$

gdzie Λ_n — stała Lebesgue'a. Stąd

$$\|f-p\| \leq \|f-p^*\| + \varepsilon\cdot\Lambda_n.$$

□

10. Dla jakich $x > 0$ sensowne jest korzystanie z interpolacji liniowej by wyznaczyć $f(x) = \ln(x)$, mając do dyspozycji stablicowane wartości $y_i \approx \ln(x_i)$ z dokładnością do 10^{-5} dla $x_i = ih$, $i = 1,2,\ldots$, gdzie $h = 10^{-3}$?

Rozw. Co to znaczy: „sensowne"? Oznaczmy $\varepsilon = 10^{-5}$, więc wiemy, że $|y_i - f(x_i)| \leq \varepsilon$. Chciałoby się, by tak (z grubsza) samo było w dowolnym x, powiedzmy, $|p_i(x) - f(x)| \leq 5\varepsilon = 5\cdot 10^{-5}$.

Dla $x \in [x_i, x_{i+1}]$, ze wzoru na interpolację dla równoodległych, dla p_i^* interpolującego prawdziwe wartości f,

$$|f(x) - p_i^*(x)| \leq \|f''\|_{\infty,[x_i,x_{i+1}]} h^2/8.$$

Łatwo policzyć, że $f''(x) = (1/x)' = -1/x^2$, skąd $\|f''\|_{\infty,[x_i,x_{i+1}]} = 1/x_i^2$. Z zadania 9 wiemy, że wtedy $\|f-p_i\| \leq \|f-p_i^*\| + \varepsilon\Lambda_1$. Reszta to rachunki. Ponieważ funkcje bazy Lagrange'a stopnia 1 są bardzo proste, $l_i(x) = \dfrac{x-x_i}{x_1-x_0}$, to

$$\Lambda_1 = \max_{x\in[[x_i,x_{i+1}]} |l_0(x)| + |l_1(x)| = 1.$$

Zatem warunek wystarczający na x_i jest taki, że

$$\frac{h^2}{8x_i^2} + \varepsilon \cdot 1 \leq 5\varepsilon.$$

Jak widać, x_i nie może być zbyt mały — co, mam nadzieję, wiedzieliśmy intuicyjnie od początku... □

11. Niech będą dane (dowolnie wybrane) $x_0, \ldots, x_N$ i $y_0, \ldots, y_N$. Wykaż, że istnieje co najmniej jeden wielomian p stopnia co najwyżej N taki, że

$$p(x_0) = y_0, \quad p'(x_1) = y_1, \quad \ldots, \quad p^{(N)}(x_N) = y_N.$$

Rozw. Zapiszmy to jako układ równań liniowych na współczynniki i sprawdźmy, że ta macierz jest nieosobliwa. Stąd wynika, że jest dokładnie jeden taki wielomian. □

2.3. Deser — nieobowiązkowy!

Dziś deseru nie ma: trzeba uczyć się do kolokwium!

Matematyka obliczeniowa: ćw. zdalne 2020-05-20
Splajny. Piotr Krzyżanowski

---

1. Zadania z rozwiązaniami

Namawiam Państwa do próby samodzielnego rozwiązania każdego z zadań, a dopiero potem do przeczytania gotowca. O ile nie będzie powiedziane inaczej, zawsze będziemy zakładać, że obliczenia są wykonywane w arytmetyce dokładnej.

1.1. Przystawki

1. Jak reprezentować splajny liniowe? Kubiczne?

Rozw. Zasadniczo dwie opcje reprezentacji splajnu stopnia k:

1. PP (*piecewise polynomial*), czyli splajn na każdym odcinku $[x_i, x_{i+1})$ zadajemy przez współczynniki $a_{i0}, \ldots, a_{ik}$ rozwinięcia w (jakiejś ustalonej) bazie wielomianowej. Wygodnie może być w bazie Newtona z wielokrotnym węźle x_i, czyli $1, (x-x_i), \ldots, (x-x_i)^k$, ale są też inne opcje, opisane na wykładzie.
2. w B-bazie.

Zauważmy, że pierwszy sposób nadaje się do reprezentowania dowolnej funkcji kawałkami wielomianowej (bez żadnych wymagań regularności). Ale ponieważ splajny mają dodatkowy warunek bycia funkcją klasy C^{k-1}, reprezentacja PP jest nieco nadmiarowa. Reprezentacja w B-bazie (skoro jest to baza!) jest zaś minimalna. □

2. Wyznacz układ równań jaki ma spełniać interpolacyjny splajn liniowy w B-bazie splajnowej.

Rozw. Ponieważ $s(x) = \sum_{i=-1}^{n-1} c_i B_i^1(x)$, gdzie $c_i = f(x_i)$ (dlaczego?), to układ jest trywialny (z macierzą jednostkową) i naprawdę nic nie trzeba rozwiązywać! □

3. Wykaż, że splajn liniowy s_1 interpolujący $f \in C^2[a,b]$ w $n+1$ węzłach równoodległych $x_i = a + i \cdot h$, $i = 0, \ldots, n$ przy czym $h = (b-a)/n$, spełnia oszacowanie błędu aproksymacji:

$$||f - s_1||_\infty \leq \frac{||f''||_\infty}{8} h^2.$$

Wykaż ponadto, że $\|s_1\|_\infty \leq \|f\|_\infty$.

Rozw. Wystarczy na każdym odcinku $[x_i, x_i + h]$ skorzystać z oszacowania błędu interpolacji wielomianowej na węzłach „równoodległych". Oszacowanie normy wynika stąd, że dla splajnu liniowego interpolacyjnego $\|s\|_\infty = \max |s(x_i)| = \max |f(x_i)| \leq \max_{x \in [a,b]} |f(x)| = \|f\|_\infty$. □

4. Jeśli splajn kubiczny s klasy C^2 i oparty na węzłach $x_0 < x_1 < x_2 < x_3$ jest taki, że $s = 0$ w $[x_0, x_1] \cup [x_2, x_3]$, to $s \equiv 0$.

Rozw. Ponieważ s ma być klasy C^2, wielomian p, który reprezentuje s na odcinku $[x_1, x_2]$ musi spełniać następujące warunki zgodności:

$$\begin{aligned} p(x_1) &= 0 = p(x_2) \\ p'(x_1) &= 0 = p'(x_2) \\ p''(x_1) &= 0 = p''(x_2) \end{aligned}$$

Jak wiadomo istnieje dokładnie jeden wielomian interpolacji Hermite'a (stopnia ≤ 5) oparty na dwóch węzłach o krotności 3 każdy. Z drugiej strony $p \equiv 0$ spełnia wszystkie te warunki — zatem to musi być on i tylko on.

Można też sprokurować tabliczkę różnic dzielonych (z samymi zerami) i stwierdzić, że wypełni się też zerami. □

1.2. Danie główne

5. Jeśli węzły, na ktorych oparta jest przestrzeń splajnów stopnia k, są równoodległe (tzn. $x_{i+1} - x_i = h$ dla każdego i), to funkcje B-bazy są identyczne z dokładnością do przesunięcia argumentu: $B_i^k(x) = B_0^k(x - ih)$.

Rozw. Rekurencyjna definicja B-bazy przekłada się w nauralny sposób na dowód indukcyjny. Dla funkcji $B_i^0(x)$ teza w oczywisty sposób zachodzi. Następnie korzystamy z definicji B-bazy:

$$B_i^k(x) = V_i^k(x) \cdot B_i^{k-1}(x) + (1 - V_{i+1}^k(x)) \cdot B_{i+1}^{k-1}(x)$$

gdzie

$$V_i^k(x) = \frac{x - x_i}{x_{i+k} - x_i}, \qquad B_i^0(x) = \begin{cases} 1 & \text{dla } x \in [x_i, x_{i+1}) \\ 0 & \text{w p.p.} \end{cases}$$

Zakładając, że dla $k-1$ teza jest prawdziwa mamy:

$$B_i^{k-1}(x) = B_0^{k-1}(x - ih) \qquad \text{oraz} \qquad B_{i+1}^{k-1}(x) = B_0^{k-1}(x - (i+1)h) = B_0^{k-1}(x - ih - h) = B_1^{k-1}(x - ih).$$

Ponadto

$$V_i^k(x) = \frac{x - x_i}{x_{i+k} - x_i} = \frac{x - x_0 - ih}{x_0 + ih + kh - x_0 - ih} = \frac{(x - ih) - x_0}{x_0 + kh - x_0} = V_0^k(x - ih)$$

i w konsekwencji także

$$V_{i+1}^k(x) = V_0^k(x - (i+1)h) = V_0^k(x - ih - h) = V_1^k(x - ih),$$

skąd teza. □

1.3. Deser — nieobowiązkowy!

To są zadania o tematyce nieco pobocznej, dla chętnych i zainteresowanych.

6. Wykaż, że każdy splajn $s \in S_k$ oparty na węzłach $x_0 < \ldots < x_n$ można jednoznacznie reprezentować na $[x_0, x_n]$ w postaci

$$s(x) = \sum_{i=0}^{k} a_i x^i + \sum_{j=1}^{n-1} b_j (x - x_i)_+^j, \qquad x \in [x_0, x_n],$$

gdzie

$$(x - x_i)_+^j = \begin{cases} (x - x_i)^j & \text{dla } x \geq x_i, \\ 0 & \text{dla } x < x_i. \end{cases}$$

Matematyka obliczeniowa: ćw. zdalne 2020-05-27
Splajny — ciąg dalszy; FFT. Piotr Krzyżanowski

---

1. Zadania z rozwiązaniami

Namawiam Państwa do próby samodzielnego rozwiązania każdego z zadań, a dopiero potem do przeczytania gotowca. O ile nie będzie powiedziane inaczej, zawsze będziemy zakładać, że obliczenia są wykonywane w arytmetyce dokładnej.

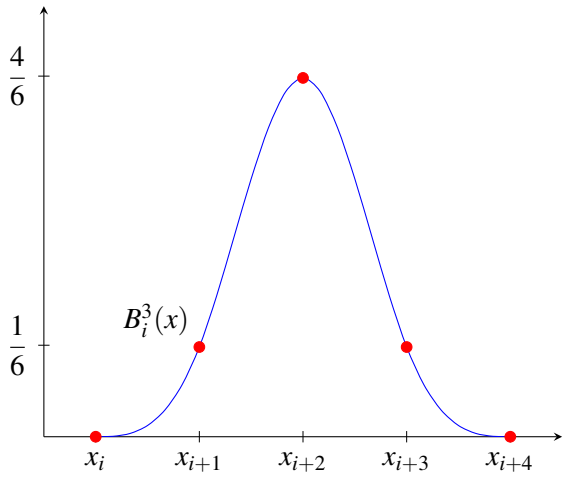
1.1. Przystawki

Zostały z zeszłego tygodnia, więc — szanowni Państwo wybaczą — nie podajemy...

1.2. Danie główne

1. Wyznacz układ równań, jaki ma spełniać interpolacyjny splajn kubiczny w B-bazie splajnowej.

Rozw. Dla S_3 opartych na węzłach $x_0 < \ldots < x_n$, B-baza to $B^3_{-3}, \ldots, B^3_{n-1}$. Tych funkcji jest $n+3$.



Szukany splajn to

$$s(x) = \sum_{i=-3}^{n-1} c_i B_i^3(x).$$

Ma być z warunków interpolacji

$$s(x_j) = \sum_{i=-3}^{n-1} c_i B_i^3(x_j) = f(x_j), \qquad j = 0, \ldots, n.$$

Można liczyć $B_i^3(x_j)$ (i ewentualnie pochodne do warunków brzegowych) z wzoru de Boora, ale prościej kazać to zrobić komputerowi i w efekcie dostać tabelę 1.

x	x_i	x_{i+1}	x_{i+3}	x_{i+3}	x_{i+4}
$B_i^3(x)$	0	$1/6$	$4/6$	$1/6$	0
$(B_i^3(x))'$	0	$1/2h$	0	$-1/2h$	0
$(B_i^3(x))''$	0	$1/h^2$	$-2/h^2$	$1/h^2$	0

Tabela 1. Tabela wartości splajnu bazowego B_i^3 (w węzłach równoodległych o h)

Ponieważ nośnik B_i^3 jest zawarty w (x_i, x_{i+4}), to suma się skraca do trzech składników

$$s(x_j) = \sum_{i=j-3}^{j-1} c_i B_i^3(x_j) = f(x_j), \qquad j = 0, \ldots, n.$$

Skąd wyznaczamy warunki interpolacji w każdym z węzłów x_j:

$$\frac{1}{6}(c_{j-3} + 4c_{j-2} + c_{j-1}) = f(x_j), \qquad j = 0, \ldots, n.$$

Odpowiada mu macierz „trójdiagonalna" o $n+1$ wierszach i $n+3$ kolumnach:

$$\begin{bmatrix} 1 & 4 & 1 & & \\ & \ddots & \ddots & \ddots & \\ & & 1 & 4 & 1 \end{bmatrix} \cdot \begin{bmatrix} c_{-3} \\ c_{-2} \\ \vdots \\ c_{n-2} \\ c_{n-1} \end{bmatrix} = 6 \begin{bmatrix} f(x_0) \\ \vdots \\ f(x_n) \end{bmatrix}$$

dlatego m.in. potrzebujemy jeszcze dwóch dodatkowych warunków, np. brzegowych. □

2. Wyznacz układ równań, jaki ma spełniać interpolacyjny splajn $s \in S_k$, $k \geq 1$, reprezentowany w B-bazie splajnowej i określony na $[x_0, x_n]$.

Rozw. To uogólnienie zadania 1. Mamy $s = \sum_{j=-k}^{n-1} c_j B_j^k$ i warunki interpolacji

$$s(x_i) = \sum_{j=-k}^{n-1} c_j B_j^k(x_i) = f(x_i), \qquad i = 0, \ldots, n.$$

Ponieważ nośnik B_j^k jest zawarty w (x_j, x_{j+k+1}) — przedział otwarty! — więc w punkcie x_i nie zerują się jedynie $B_{i-k}, B_{i-k+1}, \ldots, B_{i-2}, B_{i-1}$. Zatem warunki interpolacji upraszczają się do

$$c_{i-k} B_{i-k}^k(x_i) + \ldots + c_{i-1} B_{i-1}^k(x_i) = f(x_i), \qquad i = 0, \ldots, n.$$

Odpowiada mu macierz pasmowa o szerokości pasma k, $n+1$ wierszach i $n+k$ kolumnach:

$$\begin{bmatrix} \star & \cdots & \star & & \\ & \ddots & \cdots & \ddots & \\ & & \star & \cdots & \star \end{bmatrix} \cdot \begin{bmatrix} c_{-k} \\ c_{-k+1} \\ \vdots \\ c_{n-2} \\ c_{n-1} \end{bmatrix} = \begin{bmatrix} f(x_0) \\ \vdots \\ f(x_n) \end{bmatrix}$$

□

3. Algorytm de Boora wyznaczania wartości splajnu kubicznego zadaniego w B-bazie. Splajn kubiczny oparty na węzłach $x_0 < \ldots < x_N$ jest zadany przez współczynniki rozwinięcia w B-bazie, $s(x) = \sum_{i=-3}^{N-1} c_i B_i^3(x)$. Sformułuj algorytm obliczania wartości $s(x)$ w zadanym $x \in [x_0, x_N]$. *Wskazówka. Skorzystaj z rekurencyjnej definicji B-bazy.*

Rozw. Punkt x należy do pewnego przedziału $[x_j, x_{j+1}]$ (który to, możemy rozstrzygnąć metodą wyszukiwania binarnego w czasie $\mathcal{O}(\log_2 N)$). Skoro funkcje bazowe mają ograniczony nośnik (patrz zadanie 1), to suma w punkcie x się skraca do *czterech* składników

$$s(x) = \sum_{i=j-3}^{j} c_i^3 B_i^3(x).$$

Dalszy ciąg rozwiązania przedstawimy w ogólniejszym przypadku — splajnów k-tego stopnia. Wtedy dla wartości w x mamy co najwyżej $k+1$ niezerowych składników

$$s(x) = \sum_{i=j-k}^{j} c_i^k B_i^k(x).$$

Następnie korzystamy z definicji B-bazy:

$$B_i^k(x) = V_i^k(x) \cdot B_i^{k-1}(x) + (1 - V_{i+1}^k(x)) \cdot B_{i+1}^{k-1}(x)$$

gdzie

$$V_i^k(x) = \frac{x - x_i}{x_{i+k} - x_i}, \qquad B_i^0(x) = \begin{cases} 1 & \text{dla } x \in [x_i, x_{i+1}) \\ 0 & \text{w p.p.} \end{cases}$$

i zapisujemy nasz splajn w terminach bazy niższego stopnia (pomijając argument x):

$$s(x) = s = \sum_{i=j-k}^{j} c_i^k B_i^k = \sum_{i=j-k}^{j} c_i^k \left(V_i^k \cdot B_i^{k-1} + (1 - V_{i+1}^k) \cdot B_{i+1}^{k-1} \right) = \sum_{i=j-k}^{j} c_i^k V_i^k \cdot B_i^{k-1} + \sum_{i=j-k}^{j} c_i^k (1 - V_{i+1}^k) \cdot B_{i+1}^{k-1}.$$

Zmieniając indeksowanie w ostatniej sumie tak, że nowe i jest równe staremu $i+1$ mamy

$$\sum_{i=j-k}^{j} c_i^k (1 - V_{i+1}^k) \cdot B_{i+1}^{k-1} = \sum_{i=j-k+1}^{j+1} c_{i-1}^k (1 - V_i^k) \cdot B_i^{k-1}$$

zatem

$$\begin{aligned} s &= \sum_{i=j-k}^{j} c_i^k V_i^k \cdot B_i^{k-1} + \sum_{i=j-k+1}^{j+1} c_{i-1}^k (1 - V_i^k) \cdot B_i^{k-1} \\ &= \underbrace{c_{j-k}^k V_{j-k}^k \cdot B_{j-k}^{k-1}}_{=0} + \sum_{i=j-(k-1)}^{j} \underbrace{\left(c_i^k V_i + c_{i-1}^k (1 - V_{i-1}^k) \right)}_{=: c_i^{k-1}} \cdot B_i^{k-1} + \underbrace{c_j^k (1 - V_{j+1}^k) \cdot B_{j+1}^{k-1}}_{=0}, \end{aligned}$$

gdyż wśród funkcji bazowych stopnia $k-1$, jedyne niezerowe nośniki w $[x_j, x_{j+1}]$ mają $B_j^{k-1}, B_{j-1}^{k-1}, \ldots, B_{j-(k-1)}^{k-1}$. Dostajemy więc zależność rekurencyjną

$$s = \sum_{i=j-k}^{j} c_i^k B_i^k = \sum_{i=j-k+1}^{j} c_i^{k-1} B_i^{k-1},$$

gdzie

$$c_i^{k-1} = c_i^k V_i + c_{i-1}^k (1 - V_{i-1}^k), \qquad i = j-k+1, \ldots, j.$$

Iterując ten pomysł, należy wyznaczyć kolejne kolumny tabelki

$$\begin{array}{c|cccc} c_j^k & c_j^{k-1} & \cdots & \cdots & c_j^0 \\ c_{j-1}^k & c_{j-1}^{k-1} & & \iddots & \\ \vdots & \vdots & c_{j-k+2}^{k-2} & & \\ \vdots & c_{j-k+1}^{k-1} & & & \\ c_{j-k}^k & & & & \end{array}$$

Wartość splajnu to $s(x) = c_j^0$. □

4. Jakim algorytmem wyznaczać naturalny splajn kubiczny interpolacyjny? Jakie są własności macierzy układu równań (symetria, dodatnia określoność, struktura, uwarunkowanie)? A jeśli szukamy okresowego splajnu? A Hermitowskiego?

Rozw. To ciąg dalszy zadania 1. Dokładamy naturalne warunki brzegowe,

$$s''(x_0) = s''(x_n) = 0,$$

które po zapisaniu s w B-bazie i zróżniczkowaniu redukują się do

$$c_{-3}\underbrace{B''_{-3}(x_0)}_{=1/h^2} + c_{-2}\underbrace{B''_{-2}(x_0)}_{=-2/h^2} + c_{-1}\underbrace{B''_{-1}(x_0)}_{=1/h^2} = 0$$

i analogicznie w x_n. Dokładając te warunki (pomnożone uprzednio przez h^2) do układu równań jako pierwsze i ostatnie równanie mamy

$$\begin{bmatrix} 1 & -2 & 1 & & \\ 1 & 4 & 1 & & \\ & \ddots & \ddots & \ddots & \\ & & 1 & 4 & 1 \\ & & 1 & -2 & 1 \end{bmatrix} \cdot \begin{bmatrix} c_{-3} \\ c_{-2} \\ \vdots \\ c_{n-2} \\ c_{n-1} \end{bmatrix} = 6 \begin{bmatrix} 0 \\ f(x_0) \\ \vdots \\ f(x_n) \\ 0 \end{bmatrix}.$$

Dopieśćmy ten układ tak, by wystarczyło rozwiązać układ równań z macierzą trójdiagonalną. Odejmując pierwszy wiersz od drugiego (i analogicznie ostatni od przedostatniego) dostajemy układ równoważny

$$\begin{bmatrix} 1 & -2 & 1 & & & & \\ 0 & 6 & 0 & & & & \\ & 1 & 4 & 1 & & & \\ & & \ddots & \ddots & \ddots & & \\ & & & 1 & 4 & 1 & \\ & & & & 0 & 6 & 0 \\ & & & & 1 & -2 & 1 \end{bmatrix} \cdot \begin{bmatrix} c_{-3} \\ c_{-2} \\ c_{-1} \\ \vdots \\ c_{n-3}s \\ c_{n-2} \\ c_{n-1} \end{bmatrix} = 6 \begin{bmatrix} 0 \\ f(x_0) \\ f(x_1) \\ \vdots \\ f(x_{n-1}) \\ f(x_n) \\ 0 \end{bmatrix},$$

więc jak widać wystarczy najpierw rozwiązać układ równań rozmiaru $n+1$ z macierzą trójdiagonalną:

$$\begin{bmatrix} 6 & 0 & & & \\ 1 & 4 & 1 & & \\ & \ddots & \ddots & \ddots & \\ & & 1 & 4 & 1 \\ & & & 0 & 6 \end{bmatrix} \cdot \begin{bmatrix} c_{-2} \\ c_{-1} \\ \vdots \\ c_{n-3} \\ c_{n-2} \end{bmatrix} = 6 \begin{bmatrix} f(x_0) \\ f(x_1) \\ \vdots \\ f(x_{n-1}) \\ f(x_n) \end{bmatrix},$$

a na koniec wyznaczyć z warunków brzegowych c_{-3} i c_{n-1}. Ale ten układ dalej się upraszcza, bo $6c_{-2} = 6f(x_0)$ i podobnie c_{n-2}, więc ostatecznie wystarczy rozwiązać

$$\begin{bmatrix} 4 & 1 & & \\ \ddots & \ddots & \ddots & \\ & 1 & 4 & \end{bmatrix} \cdot \begin{bmatrix} c_{-1} \\ \vdots \\ c_{n-3} \end{bmatrix} = \begin{bmatrix} 6f(x_1) - f(x_0) \\ \vdots \\ 6f(x_{n-1}) - f(x_n) \end{bmatrix}.$$

Koszt całkowity oczywiście jest liniowy. Macierz jest faktycznie symetryczna. Jej uwarunkowanie w normie spektralnej łatwo oszacować z twierdzenia Gerszgorina. Rzeczywiście, wartości własne są rzeczywiste, bo macierz jest symetryczna i dodatkow0 siedzą w kołach

$$\{z : |z-4| \le 2\} \cup \{z : |z-4| \le 1\}$$

skąd $\operatorname{cond}_2(A) = \dfrac{\max_i |\lambda_i|}{\min_i |\lambda_i|} \le 6/2 = 3.$ □

5. Wyznacz naturalny splajn kubiczny s oparty na węzłach: $0, \pm h, \pm 2h$ taki, że $s(0) = 1$, $s(\pm 2h) = 0$, $s(\pm h) = 1/4$

(a) w bazie kawalkami wielomianowej
(b) w B-bazie

rozwiązując odpowiednie układy równań.

Rozw. Zróbmy to w B-bazie. Na podstawie zadania 4 mamy do rozwiązania układ równań

$$\begin{bmatrix} 1 & -2 & 1 & & & & \\ 1 & 4 & 1 & & & & \\ & 1 & 4 & 1 & & & \\ & & 1 & 4 & 1 & & \\ & & & 1 & 4 & 1 & \\ & & & & 1 & 4 & 1 \\ & & & & 1 & -2 & 1 \end{bmatrix} \begin{bmatrix} c_{-3} \\ c_{-2} \\ c_{-1} \\ c_{-0} \\ c_1 \\ c_2 \\ c_3 \end{bmatrix} = 6 \cdot \frac{1}{4} \cdot \begin{bmatrix} 0 \\ 0 \\ 1 \\ 4 \\ 1 \\ 0 \\ 0 \end{bmatrix}.$$

Ponieważ prawa strona to pomnożona przez $3/2$ czwarta kolumna macierzy układu, to rozwiązaniem jest

$$\begin{bmatrix} c_{-3} \\ c_{-2} \\ c_{-1} \\ c_{-0} \\ c_1 \\ c_2 \\ c_3 \end{bmatrix} = \frac{3}{2} \cdot \begin{bmatrix} 0 \\ 0 \\ 0 \\ 1 \\ 0 \\ 0 \\ 0 \end{bmatrix},$$

czyli szukany splajn to po prostu bazowa funkcja B_2^3 pomnożona przez $3/2$. □

6. Podaj algorytm obliczania $A \cdot f$, gdzie A jest tzw. macierzą cykliczną o stałych diagonalach:

$$A = \begin{bmatrix} c_0 & c_{N-1} & \dots & c_1 \\ c_1 & c_0 & \ddots & \vdots \\ \vdots & \ddots & \ddots & c_{N-1} \\ c_{N-1} & \dots & c_1 & c_0 \end{bmatrix}.$$

Rozw.

Pokażemy, że

$$A = F_N \Lambda F_N^{-1}, \tag{1}$$

gdzie F_N jest macierzą FFT, natomiast Λ jest macierzą diagonalną:

$$\Lambda = \operatorname{diag}(F_N c),$$

gdzie $c = [c_0, \dots, c_{N-1}]^T$. Wtedy obliczenie Ax sprowadza się do obliczenia $x = F_N\left(\Lambda\left(F_N^{-1} b\right)\right)$.

Niech N będzie potęgą dwójki. Mnożenie przez F_N i odwrotną do niej kosztuje $\mathcal{O}(N \log_2 N)$, a mnożenie przez macierz diagonalną kosztuje $\mathcal{O}(N)$. Wcześniej trzeba jeszcze obliczyć $F_N c$, co kosztuje kolejne $\mathcal{O}(N \log_2 N)$. Ostateczny koszt jest więc $\mathcal{O}(N \log_2 N)$.

Gdy zaś N jest liczbą pierwszą, to FFT nie można użyć i wtedy mnożenie przez F_N kosztuje $\mathcal{O}(N^2)$, więc ostatecznie tyle też kosztuje wyznaczenie x.

Aby pokazać (1), musimy pokazać, że

$$AF_N = F_N \Lambda,$$

innymi słowy — że wektory własne A to kolumny macierzy F_N, a wartości własne to λ_k. A to jest tylko rachunek! Rzeczywiście, k–ta kolumna F_N jest postaci

$$v_k = \begin{bmatrix} 1 \\ (\omega_N^k)^1 \\ \vdots \\ (\omega_N^k)^{N-1} \end{bmatrix}$$

oraz

$$\lambda_k = (F_N c)_k = \sum_{j=0}^{N-1} \omega_N^{kj} \cdot c_j,$$

więc m–ta współrzędna iloczynu $\lambda_k v_k$ jest równa

$$(\lambda_k v_k)_m = \underbrace{\sum_{j=0}^{N-1} \omega_N^{kj} \cdot c_j}_{\lambda_k} \cdot \omega_N^{km} = \sum_{j=0}^{N-1} (\omega_N^k)^{j+m} \cdot c_j$$

natomiast m–ta współrzędna iloczynu Av_k to (sumę warto rozbić na „do diagonali A" i „od diagonali A")

$$(Av_k)_m = \sum_{j=0}^{m-1} c_{N-m+j} \omega_N^{kj} + \sum_{j=m}^{N-1} c_{j-m} \omega_N^{kj}.$$

Zmieniając indeksy tak, że w pierwszej sumie nowy indeks $s = N - m + j$, a w drugiej sumie nowy indeks $s = j - m$ dostajemy

$$(Av_k)_m = \sum_{s=N-m}^{N-1} c_s (\omega_N^k)^{s+m-N} + \sum_{s=0}^{N-m-1} c_s (\omega_N^k)^{s+m} = \sum_{s=0}^{N-1} (\omega_N^k)^{s+m} \cdot c_s,$$

bo $(\omega_N^k)^{s+m-N} = (\omega_N^k)^{-N} \cdot (\omega_N^k)^{s+m}$ i $(\omega_N^k)^{-N} = \omega_N^N)^{-k} = 1^{-k} = 1$. □

1.3. Deser — nieobowiązkowy!

To są zadania o tematyce nieco pobocznej, dla chętnych i zainteresowanych.

7. Wyprowadź algorytmy obliczania w punkcie $x \in [x_0, x_n]$ pierwszej i drugiej pochodnej splajnu kubicznego zadanego w B-bazie i oszacuj ich koszt w zależności od n.

8. Niech N będzie potęgą dwójki. Podaj algorytm rozwiązywania układu równań $Ax = b$ kosztem $\mathcal{O}(N \log N)$, gdy A jest tzw. cykliczną macierzą Toeplitza:

$$A = \begin{bmatrix} c_1 & c_N & \dots & c_2 \\ c_2 & c_1 & \ddots & \vdots \\ \vdots & \ddots & \ddots & c_N \\ c_N & \dots & c_2 & c_1 \end{bmatrix}.$$

Uwaga. Na ćwiczeniach dwa miesiące temu pojawiło się identyczne zadanie, ale wtedy trzeba było rozwiązać je kosztem $\mathcal{O}(N^2)$. Dzisiaj podnoszę poprzeczkę, chociaż... naprawdę jest chyba łatwiej?

Matematyka obliczeniowa: ćw. zdalne 2020-06-03
Aproksymacja jednostajna i średniokwadratowa. Piotr Krzyżanowski

---

1. Zadania z rozwiązaniami

Namawiam Państwa do próby samodzielnego rozwiązania każdego z zadań, a dopiero potem do przeczytania gotowca. O ile nie będzie powiedziane inaczej, zawsze będziemy zakładać, że obliczenia są wykonywane w arytmetyce dokładnej.

1.1. Przystawki

1. Wykaż, że wielomianem stopnia co najwyżej N, najlepiej aproksymującym w sensie jednostajnym na $[-1,1]$ wielomian $p(x) = x^{N+1}$ jest $p^*(x) = x^{N+1} - 2^{-N}T_{N+1}(x)$, gdzie $T_n(x) = \cos(n\arccos(x))$. Uzasadnij, że p^* to na pewno wielomian, w dodatku stopnia N.

Rozw. Musimy sprawdzić, czy dla p i p^* jest alternans. Po pierwsze zauważmy, że p^* jest faktycznie wielomianem stopnia N, a nie $N+1$, bo wyrazy z x^{N+1} się zniosą. Ponadto, $p - p^* = 2^{-N}T_{N+1}$ ma dokładnie $N+2$ równe sobie ekstrema na przemian: $2^{-N}T_{N+1}(y_i) = (-1)^i$, gdzie $y_i = \cos(i\pi/(N+1))$ — ekstrema wielomianu Czebyszewa. Zatem mamy alternans. □

2. Wykaż, że wielomianem p^* stopnia co najwyżej N, który najlepiej aproksymuje funkcję $f(x) = x^N$ w sensie średniokwadratowym na $[-1,1]$ (tzn. w przestrzeni $L^2(-1,1)$ z normą $\|f\|_2 = (\int_{-1}^{1} f^2(x)\,dx)^{1/2}$) jest

$$p^*(x) = x^{N+1} - \alpha \cdot L_{N+1},$$

gdzie L_{n+1} to wielomian Legendre'a stopnia $N+1$, natomiast stała α jest tak dobrana, by współczynnik przy najwyższej potędze wielomianu $\alpha \cdot L_{N+1}$ był równy 1.

Rozw. Oczywiście tak zdefiniowany p^* jest stopnia nie wyższego niż N (najwyższe potęgi się uproszczą). Jak wiadomo, musi zachodzić $(f - p^*, L_j) = 0$ dla $j = 0,\ldots,N$. No i mamy $f - p^* = x^{N+1} - x^{N+1} + aL_{N+1} = aL_{N+1}$, więc

$$(f - p^*, L_j) = a(L_{N+1}, L_j) = 0$$

dla $j = 0,\ldots,N$, z ortogonalności. □

3. Wykaż, że wielomian w stopnia N, najlepiej aproksymujący (w sensie jednostajnym) na przedziale $[a,b]$ zadaną funkcję ciągłą f, musi być dla niej wielomianem interpolacyjnym Lagrange'a opartym na *pewnych* (niekoniecznie równoodległych) $N+1$ punktach. *Wskazówka. Dla f i w musi istnieć alternans; skorzystaj z tego faktu.*

Rozw. Z twierdzenia o alternansie, w $[a,b]$ różnica $f - w$ musi mieć na przemian ekstrema w $N+2$ punktach alternansu. Z ciągłości musi się więc zerować gdzieś pomiędzy punktami alternansu. □

1.2. Danie główne

4. Zygmunt chwali się: „Jeśli podacie mi wartości funkcji sinus w 10 wybranych przeze mnie punktach, będę mógł tanio obliczyć jej wartość dla dowolnego $x \in \mathbb{R}$ z błędem nieprzekraczającym 10^{-8}”. Czy ma rację i dlaczego? I co to znaczy: „tanio”? (Przyjmij, że jedyne dopuszczalne działania to podstawowe cztery działania arytmetyczne, a arytmetyka jest dokładna.)

Rozw. Po pierwsze, wystarczy, że Zygmunt będzie znał te wartości $f(x) = \sin(x)$ w punktach przedziału $[0, \pi/2]$, bo każdą inną wyznaczy sobie korzystając albo z symetrii, albo okresowości. Na $[0, \pi/2]$ oprzyjmy wielomian interpolacyjny p na $n+1 = 10$ węzłach Czebyszewa. Zgodnie z teorią, błąd interpolacji w $n+1$ węzłach Czebyszewa na $[a,b]$ szacuje się następująco:

$$\|f - p\|_\infty \leq \frac{\|f^{(n+1)}\|_\infty}{(n+1)!} \|\phi_{n+1}\|_\infty \leq \frac{\|f^{(n+1)}\|_\infty}{2^n (n+1)!} \left(\frac{b-a}{2}\right)^{n+1},$$

skąd w naszym przypadku

$$\|f - p\|_\infty \leq \frac{1}{2^n (n+1)!} \left(\frac{\pi}{4}\right)^{n+1} = \frac{1}{2^9 10!} \left(\frac{\pi}{4}\right)^{10} \approx 4.8069 \cdot 10^{-11}.$$

Już dla $n+1 = 9$ błąd jest grubo poniżej 10^{-8}. Notabene, w naszym szczególnym przypadku, dla węzłów równoodległych jest tylko ciutkę gorzej...

Na koniec zastanówmy się, jaki to koszt? Trzeba najpierw zredukować x do przedziału $[0, \pi/2]$, co kosztuje kilka flopów i prawdopodobnie parę instrukcji **if**, a następnie obliczyć jedną wartość wielomianu stopnia 10 (co kosztuje około $3 \cdot 10$ flopów). W sumie — to nie jest jakoś *bardzo* tanio... Koszt obliczenia sinusa w *podwójnej* precyzji w standardowej bibliotece matematycznej jest rzędu kilkunastu flopów. □

5. Znajdź funkcję liniową najlepiej aproksymującą na $[x_0, x_2]$ w sensie jednostajnym wypukłą funkcję różniczkowalną f.

Rozw. Szukany wielomian to $w(x) = Ax + B$. Gdyby to był interpolant przez x_0, x_2, to wtedy $A = (f(x_2) - f(x_0))/(x_2 - x_0)$, ale jednak wtedy wszystkie ekstrema błędu będą tego samego znaku: cięciwa funkcji wypukłej nie może leżeć pod wykresem samej funkcji.

Ekstremum błędu x_1 wypadające wewnątrz (x_0, x_2) musi spełniać $f'(x_1) - w'(x_1) = 0$. Weźmy więc x_1 taki, że $f'(x_1) = A$. (Oczywiście istnieje na mocy twierdzenia Cauchy'ego. Możemy go znaleźć, rozwiązując równanie nieliniowe $f'(x_1) = A$.)

Aby był alternans, w trzech punktach musi być ekstremum, ze znakami na przemian. Załóżmy, że dwa z nich to krańce odcinka, a trzeci to x_1. Z warunku alternansu musi być

$$f(x_0) - Ax_0 - B = -f(x_1) + Ax_1 + B,$$

skąd możemy już wyznaczyć B:

$$B = \frac{f(x_0) + f(x_1)}{2} - A\frac{x_0 + x_1}{2}.$$

Jest alternans, więc tak wyznaczony $w = Ax + B$ musi być optymalny. □

6. Pokaż, że wielomiany *ortogonalne* w $L^2_\rho(a,b)$ postaci $p_n(x) = x^n + \ldots$ niższe potęgi$\ldots$ spełniają

$$p_{n+1}(x) = (x-\beta_n)p_n(x) - \gamma_n p_{n-1}(x),$$

przy czym $\gamma_n > 0$.

Rozw. Formuła trójczłonowa — oczywiste (wykład!). Mnożąc skalarnie przez p_{n-1} mamy

$$\underbrace{(p_{n+1}, p_{n-1})}_{=0} = (x\cdot p_n, p_{n-1}) - \beta_n \underbrace{(p_n, p_{n-1})}_{=0} - \gamma_n (p_{n-1}, p_{n-1}),$$

skąd (wykład!)

$$\gamma_n = \frac{(x\cdot p_n, p_{n-1})}{(p_{n-1}, p_{n-1})}$$

Ale z tej samej formuły 3-członowej — wszak prawdziwej dla dowolnego k — dostajemy, że $(p_{k+1}, p_{k+1}) = (x\cdot p_k, p_{k+1}) = (x\cdot p_{k+1}, p_k)$, skąd (dla $k = n-1$)

$$\gamma_n = \frac{(x\cdot p_n, p_{n-1})}{(p_{n-1}, p_{n-1})} = \frac{(p_n, p_n)}{(p_{n-1}, p_{n-1})} > 0.$$

□

7. Niech $\{p_n\}$ będą rodziną wielomianów ortogonalnych w $L^2_\rho(-a,a)$, gdzie waga ρ jest funkcją parzystą, $\rho(-x) = \rho(x)$. Wykaż, że dla n parzystych p_n są funkcjami parzystymi, a dla n nieparzystych p_n są funkcjami nieparzystymi.

Rozw. Skorzystajmy z powyższej formuły 3-członowej i działajmy przez indukcję. Zauważmy, że $p_0 = 1$ jest oczywiście funkcją parzystą, natomiast $p_1(x) = \mathrm{Const}\cdot x$ (aby $(p_1, p_0) = 0$), więc jest nieparzystą.

Załóżmy więc, że dla stopni mniejszych od n jest to prawda. Pokażemy, że w formule 3-członowej musi wtedy być zawsze $\beta_n = 0$: mnożąc skalarnie formułę przez p_n mamy

$$\underbrace{(p_{n+1}, p_n)}_{=0} = (x\cdot p_n, p_n) - \beta_n (p_n, p_n) - \gamma_n \underbrace{(p_{n-1}, p_n)}_{=0},$$

skąd

$$\beta_n = \frac{(x\cdot p_n, p_n)}{(p_n, p_n)} = 0,$$

bo $(x\cdot p_n, p_n) = \int_{-a}^{a} x \underbrace{p_n^2(x)\rho(x)}_{\text{parz.}}\, dx$ no i funkcja podcałkowa jest nieparzysta.

Jeśli więc teraz n jest parzyste, to $n+1$ jest nieparzyste i z założenia indukcyjnego

$$p_{n+1}(x) = \underbrace{x p_n(x)}_{\text{nieparz.}} - \beta_n p_n(x) - \underbrace{\gamma_n p_{n-1}(x)}_{\text{nieparz.}}.$$

Identyczne rozumowanie przeprowadzamy dla nieparzystego n. □

8. Niech będą dane współczynniki α_n, β_n **formuły trójczłonowej**, definiującej ciąg wielomianów ortogonalnych:

$$P_{n+1}(x) = (x-\alpha_n)P_n(x) + \beta_n P_{n-1}(x).$$

Podaj algorytm obliczania wartości $w(x) = c_0P_0(x) + \ldots + c_NP_N(x)$ w zadanym punkcie x kosztem $\mathscr{O}(N)$. *Wskazówka. Zastosuj algorytm Clenshaw'a.*

Rozw. „Naturalny" pomysł to iterowanie od najniższego stopnia do najwyższego, zaczynając od $P_{-1}(x) = 0$ i $P_0(x) = 1$:

$w = c_0$; $P_{-1} = 0$; $P_0 = 1$
for $n = 0 : N-1$
  $P_{n+1} = (x - \alpha_n)P_n + \beta_n P_{n-1}$ ▷ wartość kolejnego wielomianu bazowego w x
  $w = w + c_{n+1}P_{n+1}$
end
return w

Ten algorytm kosztuje $\mathscr{O}(3N)$ mnożeń i $\mathscr{O}(3N)$ dodawań — ale można taniej (i to właśnie będzie robić algorytm Clenshaw'a).

Idąc od najwyższych potęg x, daje się zmniejszyć liczbę mnożeń. Ponieważ (dla uproszczenia zapisu pomijamy oznaczenie, że rzecz dzieje się w punkcie x i wprowadzamy oznaczenie $a_n = x - \alpha_n$):

$$\begin{aligned} w &= c_N P_N + \ldots + c_0 P_0 \\ &= c_N \left(a_{N-1}P_{N-1} + \beta_{N-1}P_{N-2}\right) \\ &\quad + c_{N-1}P_{N-1} \\ &\quad + c_{N-2}P_{N-2} + \ldots + c_0 P_0, \end{aligned}$$

to, porządkując, dostajemy

$$w = \underbrace{(c_N a_{N-1} + c_{N-1})}_{A_{N-1}} P_{N-1} + \underbrace{(c_{N-2} + \beta_{N-2})}_{B_{N-2}} P_{N-2} + c_{N-3}P_{N-3} + \ldots + c_0 P_0,$$

czyli w jest wartością wielomianu stopnia $N-1$ o zmienionych dwóch współczynnikach. Z tej rekurencji wynika prosta iteracja:

Algorytm Clenshaw'a (wersja beta)
for $n = N-1 : -1 : 2$
  $\begin{bmatrix} c_n \\ c_{n-1} \end{bmatrix} = \begin{bmatrix} c_{n+1}(x - \alpha_n) + c_n \\ c_{n-1} + \beta_{n-1} \end{bmatrix}$ ▷ modyfikujemy dwa współczynniki jednocześnie
end
return $w = c_0 + (x - \alpha_0)c_1$ ▷ $P_1 = (x - \alpha_0)$

Koszt całości to tym razem $\mathscr{O}(3 \cdot N)$ dodawań i tylko $\mathscr{O}(1 \cdot N)$ mnożeń!. Na deser możemy użyć zmiennych pomocniczych, by nie zamazywać wektora c:

Algorytm Clenshaw'a (wersja finalna)
$A = c_N$; $B = c_{N-1}$
for $n = N-1 : -1 : 2$
  $\begin{bmatrix} A \\ B \end{bmatrix} = \begin{bmatrix} A(x - \alpha_n) + B \\ c_{n-1} + \beta_{n-1} \end{bmatrix}$
end
return $w = B + A \cdot (x - \alpha_0)$ ▷ $P_1 = (x - \alpha_0)$

□

9. Wyznacz wielomian p^* stopnia $n \leq 2$ najlepiej aproksymujący funkcję $f(x) = |x|$ na przedziale $[-2, 2]$ w sensie aproksymacji jednostajnej. Ile wynosi błąd tej aproksymacji? *Wska-*

zówka. Punktów alternansu będzie więcej, niż minimalna potrzebna liczba. Szukaj wielomianu, który (tak jak f) jest funkcją parzystą.

Rozw. Potrzebujemy co najmniej $n+2=4$ węzły alternansu. Z symetrii zadania, postulujemy $-x_1, -x_0, x_0, x_1$, w których różnica $e(x) = p(x) - f(x)$ przyjmuje wartość ekstremalną. Ponadto postulujemy (też z symetrii), że $p(x) = ax^2 + c$. Te ekstrema mogą znaleźć się w punktach $-2, 0, 2$ (krańce oraz punkt nieróżniczkowalności) oraz w takich $x \in (0,2)$, że $e'(x) = 0$:

$$e'(x) = (ax^2 + c - x)' = 2ax - 1 = 0,$$

stąd $x_0 = 1/2a$ i musi być $x_0 \in [0,2]$. Zatem drugim punktem musi być koniec przedziału, $x_1 = 2$. Stąd $e(2) = -e(x_0)$, czyli $4a + c - 2 = -(a/4a^2 + c - 1/2a)$, skąd

$$4a + 2c - 2 = 1/4a.$$

Jak widać, zostaje nam jeszcze jeden stopień swobody, a tymczasem wiemy, że element najlepszej aproksymacji jest jedyny. Stąd wniosek, że musi być jeszcze jeden punkt alternansu, w zerze:

$$e(2) = -e(x_0) = e(0).$$

Czyli $e(2) = e(0)$, czyli $4a + c - 2 = c - 0$, skąd $a = 1/2$. Podstawiając dalej dostajemy $x_0 = 1$ i w konsekwencji $b = 1/4$. Poszukiwaną funkcją jest $p(x) = \frac{1}{2}x^2 + \frac{1}{4}$, a alternansem dla f i p są punkty $-2, -1, 0, 1, 2$. □

10. Niech $f \in C[a,b]$ oraz

— $p^* \in P_N$ będzie wielomianem najlepszej aproksymacji jednostajnej dla f na $[a,b]$,
— $p \in P_N$ będzie wielomianem interpolacyjnym dla f, opartym na węzłach $a \le x_0 < \cdots < x_N \le b$.

Wtedy

$$\|f - p^*\|_\infty \le \|f - p\|_\infty \le (1 + \Lambda_N) \cdot \|f - p^*\|_\infty,$$

gdzie $\Lambda_N = \|\lambda_N\|_\infty$ (stała Lebesgue'a), gdzie

$$\lambda_N(x) := \sum_{i=0}^{N} |l_i(x)| \qquad \leftarrow l_i : \text{funkcje bazy Lagrange'a.}$$

Rozw. Lewa nierówność jest oczywista (wszak p^* jest optymalny). Prawa nierówność:

$$\|f - p\|_\infty \le \|f - p^* + p^* - p\|_\infty \le \|f - p^*\|_\infty + \|p^* - p\|_\infty,$$

więc wystarczy udowodnić, że $\|p^* - p\|_\infty \le \Lambda_N \cdot \|f - p^*\|_\infty$. Ponieważ interpolant stopnia N wielomianu p^* to on sam, to

$$\begin{aligned} |p^*(x) - p(x)| = |\sum_i p^*(x_i) l_i(x) - \sum_i f(x_i) l_i(x)| &= |\sum_i (p^*(x_i) - f(x_i)) \cdot l_i(x)| \\ &\le \sum_i \underbrace{|(p^*(x_i) - f(x_i)|}_{\le \|f - p^*\|_\infty} \cdot |l_i(x)| \le \|f - p^*\|_\infty \cdot \lambda_N(x). \end{aligned}$$

□

1.3. Deser — nieobowiązkowy!

To są zadania o tematyce nieco pobocznej, dla chętnych i zainteresowanych.

11. Wanda upiera się, że wielomianem interpolacyjnym Lagrange'a opartym na 65 równoodległych węzłach w przedziale $[-\pi/2, \pi/2]$ można aproksymować funkcję sinus (w tym przedziale) z błędem *względnym* 10^{-64}. Klaus twierdzi, że to nie ma sensu. Rozstrzygnij ich spór wiedząc, że implementacja kodu Wandy odbędzie się w MATLAB-ie w wersji takiej, jaką mamy w labie. Kod będzie uruchamiany na biznesowym ultrabooku Klausa z procesorem Core i5, wyświetlaczem 13.3 cala w białej obudowie, pamięcią RAM 8GB i dyskiem SSD o pojemności 256GB, pracującym pod kontrolą Linuxa z najnowszej dystrybucji Ubuntu.

12. Wykaż, że jeśli s_1^* jest splajnem liniowym optymalnie aproksymującym w sensie jednostajnym funkcję ciągłą f opartym na węzłach $a = x_0 < x_1 < \ldots < x_n = b$, a s_1 — splajnem liniowym interpolującym f w tychże węzłach, to wtedy

$$||f - s_1^*||_\infty \leq ||f - s_1||_\infty \leq 2||f - s_1^*||_\infty.$$

13. Czy zadanie wyznaczenia elementu najlepszej aproksymacji w podprzestrzeni sk. wymiaru V przestrzeni unitarnej X jest dobrze uwarunkowane w normie tej przestrzeni? *Wskazówka. Chodzi o współczynnik uwarunkowania bezwzględnego. Zadaniem obliczeniowym jest wynik odwzorowania* $X \ni f \mapsto w^* \in V \subset X$.