Krzysztof Adamczyk 327264 IAD 23/24 gr. Wtorek 12

Problem: Rozwiązywanie układu równań liniowych $Ax = b, gdzie\ A \in R^{n\,x\,n}$ – trójdiagonalna macierz symetryczna, dodatnio określona, $b \in R^n$, zmodyfikowaną metodą Cholesky'ego-Banachiewicza. Wyznacz również $det(A)$ – wprost z rozkładu.

Krótki wstęp matematyczny:

Rozkład Cholesky'ego-Banachiewicza:

Procedura rozkładu symetrycznej, dodatnio określonej macierzy A, na iloczyn macierzy postaci:

$$A = L * L^T,$$

gdzie L – macierz dolnotrójkątna, a L^T – jej transpozycja. Jest to szczególny przypadek rozkładu LU.

Macierz trójdiagonalna:

Macierz, w której wszystkie elementy poza diagonalą, a także przekątnymi nad oraz pod nią, są zerowe. Również nazywana macierzą pasmową lub wstęgową. Przykładowa macierz trójdiagonalna:

$$\begin{bmatrix} B_{11} & B_{12} & 0 & \cdots & \cdots & 0 \\ B_{21} & B_{22} & B_{23} & \ddots & \ddots & \vdots \\ 0 & B_{32} & B_{33} & B_{34} & \ddots & \vdots \\ \vdots & \ddots & B_{43} & B_{44} & B_{45} & 0 \\ \vdots & \ddots & \ddots & B_{54} & B_{55} & B_{56} \\ 0 & \cdots & \cdots & 0 & B_{65} & B_{66} \end{bmatrix}$$

Macierz symetryczna:

Macierz A, taka, że $A = A^T$, gdzie macierz A jest macierzą kwadratową.

Macierz dodatnio określona:

Macierz posiadająca wszystkie wartości własne dodatnie i dla dowolnego wektora niezerowego x, zachodzi nierówność $x * A * x^T > 0$.

Związek rozkładu Cholesky'ego, a rozwiązywanie równań liniowych:

Cytowane z wykładu

Rozwiązywanie układów równań liniowych $\mathrm{Ax} = \mathrm{b}$. Podstawiając $A = \mathrm{L} * L^T$, otrzymujemy: $\mathrm{L} * L^T * \mathrm{x} = \mathrm{b}$. Rozwiązanie tego układu znajdujemy rozwiązując 2 układy z macierzami trójkątnymi: $\mathrm{L} * y = \mathrm{b}$ oraz $L^T * \mathrm{x} = \mathrm{y}$.

Związek rozkładu Cholesky'ego z wyznacznikiem macierzy:

Cytowane z wykładu

Obliczanie $det(A)$: $det(A) = det(\mathrm{L} * L^T) = det(\mathrm{L}) * \det(L^T) = det(L)^2 = \prod_{i=1}^{n} {l_{ii}}^2$

Opis działania programu i logika:

Uwaga: używałem format long

Cholesky:

Na początku podchodząc do tego zagadnienia, stwierdziłem, że warto byłoby napisać kod, który pozwoliłby nam na otrzymanie rozkładu cholesky'ego dla macierzy wejściowej. Podchodząc do tego problemu, warto sprawdzić, jak zachowuje się macierz trójdiagonalna, symetryczna, dodatnio określona podczas rozkładu. Wiemy, że ogólnie wyrazy macierzy L, wyglądają następująco:

$$l_{kk} = \sqrt{a_{kk} - \sum_{j=1}^{k-1} {l_{k,j}}^2}$$

$$l_{i,k} = \frac{a_{i,k} - \sum_{j=1}^{k-1} l_{k,j} * l_{i,j}}{l_{k,k}}$$

Wiemy, że: $a_{i,j} = 0, dla\ i \neq j, a\ także\ dla\ j \neq i \pm 1$. Macierz L jest dolnotrójkątna, zatem pomińmy $j \neq i + 1$. Obliczmy kilka pierwszych wyrazów dla naszej macierzy A:

$$l_{1,1} = \sqrt{a_{1,1}}, \quad l_{2,1} = \frac{a_{2,1}}{l_{1,1}}, \quad l_{2,2} = \sqrt{a_{2,2} - l_{2,1}}, \quad l_{3,1} = \frac{a_{3,1}}{l_{1,1}} = 0,$$

$$l_{3,2} = \frac{a_{3,2} - l_{3,1} * l_{2,1}}{l_{2,2}} = \frac{a_{3,2}}{l_{2,2}}$$

Jak widzimy elementy w macierzy L, znajdujące się poza główną przekątną i przekątną pod nią, będą zerowe. Natomiast elementy na nich możemy wyznaczyć ze wzorów:

$$l_{i,i} = \sqrt{a_{i,i} - {l_{i,i-1}}^2}$$

$$l_{i,i-1} = \frac{a_{i,i-1}}{l_{i-1,i-1}}$$

Dzięki takiemu skróceniu obliczeń jesteśmy w stanie przechowywać zarówno informacje o macierzy A w dwóch zmiennych: $a, b, gdzie\ a - wektor\ głównej\ przekątnej,\ b - wektor\ przekątnej\ pod\ i\ nad\ główną$. Również w ten sposób przechowujemy informacje o macierzy L , jako zmienne d, s. W ten prosty sposób używając tylko jednej pętli możemy wyznaczyć macierz L powstałą w wyniku rozkładu Cholesky'ego.

Rozwiąż:

Funkcja rozwiązuje metodą eliminacji Gaussa dla macierzy dolno i górno- trójkątnych układy równań L $*$ y $=$ b oraz L^T $*$ x $=$ y, w sposób przedstawiony tak jak na wykładzie:

Jednym z prostszych do rozwiązania układów równań liniowych jest układ z macierzą trójkątną (górną lub dolną). Przyjmijmy, że A jest macierzą trójkątną górną, tj.

$$A = \begin{pmatrix} a_{11} & a_{12} & \dots & a_{1n} \\ 0 & a_{22} & \dots & a_{2n} \\ \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & \dots & a_{nn} \end{pmatrix}.$$

W przypadku macierzy trójkątnych dolnych rozważania są analogiczne.
Układ równań z tą macierzą można zapisać tak:

$$\begin{aligned} a_{11}x_1 + a_{12}x_2 + \ldots + a_{1n}x_n &= b_1 \\ a_{22}x_2 + \ldots + a_{2n}x_n &= b_2 \\ &\ldots \\ a_{nn}x_n &= b_n. \end{aligned}$$

Jak rozwiązać taki układ, wszyscy się na pewno domyślają. Z ostatniego równania obliczamy x_n, wstawiamy do przedostatniego, obliczamy x_{n-1}, następnie obie wartości (x_n i x_{n-1}) wstawiamy do poprzedniego równania, itd.

Algorytm:

$$\begin{aligned} &x_n := \frac{b_n}{a_{nn}} \\ &\text{for} \quad k = n-1, n-2, \ldots, 1 \\ &\qquad x_k := \frac{b_k - \sum_{j=k+1}^{n} a_{kj}x_j}{a_{kk}} \\ &\text{end} \end{aligned}$$

Z tym, że każdy element poza główną diagonalą i przekątną pod/nad nią jest zerowy, zatem suma sprowadza się tylko do jednego iloczynu, co widać w kodzie.

Wyznacznik_cholesky:

Funkcja wyznacza w sposób opisany wyżej wyznacznik macierzy A, używając tylko przekątnych tej macierzy jako wektorów do obliczenia wektorów powstałych podczas wywołania funkcji cholesky(). Następnie oblicza ona wyznacznik podnosząc do kwadratu każdy element na diagonali macierzy L

Wbudowane:

Aby sprawdzić poprawność wyników otrzymanych z funkcji rozwiąż(), postanowiłem zaimplementować funkcje rozwiązującą układ równań liniowych przy użyciu wbudowanych metod chol() i \. Aby użyć pierwszej z nich i otrzymać macierz L, utworzyłem macierz A z wektorów a, b i wywołałem chol(A, 'lower'). Następnie otrzymałem wektory y i wynik przez zastosowanie operatora \, który rozwiązuje układy równań jako funkcja wbudowana. Po sprawdzeniu wielu danych mogę stwierdzić, że zaimplementowana funkcja rozwiąż() nie tylko zwraca dane z bardzo dużą dokładnością, jak również jest dużo bardziej wydajna od funkcji wbudowanych, co zostało przedstawione w dalszej części raportu.

Testy i przykłady użycia:

Aby przetestować poprawność działania zaimplementowanych przeze mnie funkcji, postanowiłem sprawdzić ich podatność na błędy numeryczne, a także ich wydajność czasową na tle wbudowanych. Do tego przedstawiam 7 przykładowych zestawów, które testują po kolei różne warianty danych wejściowych: $a - główna\ przekątna, b - przekątna\ nad/pod\ główną\ i\ c - wyrazy\ wolne$:

- $Zestaw\ 1 - małe\ dane$: a = [1,5,5,5] b = [2,2,2] c = [1,2,3,4];
- $Zestaw\ 2 - duża\ macierz,\ małe\ liczby$: a = 0.00001 + (0.25 - 0.00001) * rand(1, 100), b = -0.0000001 * ones(1,99), c = rand(1,100);
- $Zestaw\ 3 - duże\ liczby$: a = randi([50,1000], 1, 10), b = randi([-25,-1], 1, 9), c = randi([50,1000], 1, 10);
- $Zestaw\ 4 - duża\ macierz\ i\ duże\ liczby$: a = randi([100,1000], 1, 100), b = randi([-10,-1], 1, 99), c = randi([100,1000], 1, 100);
- $Zestaw\ 5 - diagonalna$: a = randi([100,1000], 1, 35), b = zeros(1, 34), c = randi([100,1000], 1, 35);
- $Zestaw\ 6 - potężne\ dane$: a = randi([1000,10000], 1, 1000000), b = randi([-100,-5], 1, 999999), c = randi([1000,10000], 1, 1000000);
- $Zestaw\ 7 - potężna\ macierz\ z\ małymi\ wartościami$: a = 0.00001 + (0.25 - 0.00001) * rand(1, 1000000), b = -0.000001 * ones(1,999999), c = 0.00001 + (0.25 - 0.00001) * rand(1, 1000000);

Wybrane dane nie są przypadkowe, pokazują jak wyniki mogą się zmieniać w zależności od wielkości macierzy, a także od wartości w niej zawartych. Pozwala nam to ocenić czy funkcja rozwiąż() jest podatna na błędy numeryczne np. odejmowanie bliskich siebie liczb, czy błędy reprezentacji. Chcąc dokonać analizy otrzymanych wyników oczywiście moglibyśmy porównywać wyniki linijka po linijce, jednakże dla dużych danych byłoby to bardzo nieczytelne (można porównać sobie np. zestaw drugi i display(table2). Zatem do analizy poprawności i wydajności naszych metod postanowiłem podejść w następujących krokach:

- Obliczanie maksymalnego błędu bezwzględnego względem funkcji wbudowanej używającej rozkładu cholesky'ego
- Obliczanie maksymalnego błędu bezwzględnego na danych względem zwykłego rozwiązania układu równań liniowych
- Obliczanie czasu wykonywania każdej z 3 funkcji

Również podobną analizę dokonałem dla obliczania wyznacznika macierzy używając funkcji wyznacznik_cholesky() a także standardowej funkcji det(). W tym przypadku nie działamy jednak na danych wektorowych, tylko na skalarach, zatem nie musimy szukać maksymalnego błędu.

Wyniki i wnioski z analizy danych testowych znajdują się w kolejnej części.

Wyniki + wizualizacja:

1. Wyniki otrzymane podczas rozwiązywania układu równań:

	Iter_time	Pro time	WB time	wb - iter max err	pro - iter max err
Zestaw1	0.0021951	0.0001895	0.0011063	0	0
Zestaw2	0.0002285	0.0002709	0.0002167	0	0
Zestaw3	0.0001129	0.000138	0.0001147	4.4409e-16	1.3878e-17
Zestaw4	0.0002272	0.0002379	0.0002014	0	2.2204e-16
Zestaw5	0.0001719	0.0001174	0.0001314	4.4409e-16	0
Zestaw6	0.54746	NaN	NaN	NaN	NaN
Zestaw7	0.49577	NaN	NaN	NaN	NaN

Jak widzimy dla małych zestawów danych otrzymujemy bardzo zbliżone wyniki, jeśli chodzi o czas wykonywania, lecz także błędy są bardzo małe. Ciekawiej jest już zdecydowanie w zestawie 6 i 7. Tak duże macierze, rzędu milion są niereprezentowalne w pamięci komputera, więc funkcje chol(), a także operator \ , stają się bezużyteczne. Natomiast funkcja iteracyjna, pozwala nam na rozwiązywanie układów równań, wysokich rzędów w dość krótkim czasie. Widzimy również jakie znacznie ma dobór liczb w macierzy. Niestety dla tych dużych danych nie jesteśmy w stanie szybko obliczyć w matlabie czy wyniki są poprawne, stąd wszędzie uzupełniłem tabelkę wartościami NaN.

2. Wyniki otrzymane podczas liczenia wyznacznika:

	wyznaczniki	z det	error	time wyz	time dets
Zestaw 1	1	1	0	0.0002126	9.18e-05
Zestaw 2	1.3546e-106	1.3546e-106	1.2102e-121	0.0002889	0.0002125
Zestaw 3	2.7642e+27	2.7642e+27	0	0.0001312	5.9e-05
Zestaw 4	5.8647e+270	5.8647e+270	8.446e+255	0.0004159	0.0002082
Zestaw 5	2.744e+96	2.744e+96	4.7428e+80	0.000153 6	7.97e-05
Zestaw 6	Inf	NaN	NaN	0.5832	NaN
Zestaw 7	0	NaN	NaN	0.49933	NaN

Analizując tabelkę z wyznacznikami łatwo zauważyć, że ta metoda jest już bardziej problematyczna. Dla małych danych/liczb spisuje się dobrze, lecz już dla większych zaczynają pojawiać się problemy. Pozornie wyniki otrzymane przez nas wydają się podobne, lecz w odejmowaniu zwracają bardzo dużą liczbę. Fakt może to być zarówno błąd numeryczny, ale mogą się one faktycznie różnić na dalszych elementach. Również dla największych danych następują problemy, mianowicie w zestawie 6 matlab przyjął nieskończoność, gdyż mnożone przez nas liczby były na tyle duże i analogicznie dla zestawu 7 tylko tam były za małe i zwraca zero. Zatem metoda wyznaczania wyznacznika macierzy jest dobra w przybliżeniu, lecz może prowadzić do błędów i nie jest w 100% dokładna dla niektórych danych.

Wnioski:

Reasumując, zaimplementowana przeze mnie metoda rozwiązywania równań liniowych przez rozkład cholesky'ego , jak i pozostałe funkcje działają zgodnie z założeniami i zwracają wyniki mocno zbliżone do wyników z funkcji wbudowanych. Błędy numeryczne zwykle nie wpływają na zmianę oczekiwanego wyniku, mogą to ewentualnie robić w kontekście obliczania wyznacznika macierzy. Takie podejście do rozwiązywania układów równań i liczenia det, dla macierzy trójdiagonalnych jest bardzo skuteczne, gdyż pozwala nam na wykonywania obliczeń, na naprawdę dużych danych, które dużo by ważyły w pamięci komputera. Praca nad tym projektem zdecydowanie była ucząca, pogłębiłem swoją wiedzę o modyfikacjach układów równań, co na pewno może zaprocentować na drugim kolokwium. Dziękuję za poświęcony czas i uwagę.