

## Pracownia z analizy numerycznej

### Sprawozdanie do zadania P.2.3.

Prowadzący: mgr. Filip Chudy

Wrocław, 8 stycznia 2023, 21:37

### Spis treści

1. Wstęp	1
1.1. Metodologia	1
2. Rozwiązywanie równań liniowych	2
2.1. Eliminacja Gaussa	3
2.2. Rozkład $QR$	3
3. Algorytm eliminacji Gaussa	4
3.1. Wyniki	4
3.2. Złożoność obliczeniowa	4
4. Transformacja Householdera	5
4.1. Podstawy teoretyczne	5
4.2. Wyniki	6
4.3. Złożoność obliczeniowa	6
5. Podsumowanie	7

### 1. Wstęp

Od szkoły podstawowej przez gimnazjum (lub szkołę podstawową 2.0), liceum, przez studia i okazjonalnie po studiach wymagane od uczestników systemu szkolnictwa w Polsce wymagana jest umiejętność rozwiązywania równań. Zaczynamy od prostych, przyjemnych problemów - znajdź  $x$ . Można go wskazać paluszkiem, a można znaleźć jego wartość. Każdy jest w stanie to zrobić. Potem do  $x$  dochodzi  $y$  i powstaje chłopiec lub też układ równań. Znalezienie obu jest to zadanie trudniejsze, a czasem wręcz niemożliwe. Potem idziemy na studia i spotykamy  $x_k$  i to jest moment w którym z grzecznego, ułożonego chłopca układ równań staje się hydrą o  $n$  głowach, która kąsa i gryzie studenta który bez pomocy maszyny nie umie nawet pomnożyć  $7 \cdot 13$ . W takich momentach potrzebna jest interwencja boska lub maszynowa. Chciałoby się zakrzyknąć "Google, solve this set of equations for me!". Ale Google nie słucha lub nie umie. Wolfram milczy wymownie w kącie, oceniając wybory życiowe studenta. Wtedy właśnie student musi wziąć życie w swoje ręce, sięść przed komputerem, otworzyć nano i zacząć tworzyć kod, który rozwiąże tę  $n$ -głową hydrę i pokaże, że  $x$  jest zerem.

W poniższej pracy zbierzemy nasze studenckie siły i postaramy się pokonać hydrę, trafiając jak najbliżej każdej z jej  $n$  głów.

#### 1.1. Metodologia

W poniższej pracy zostaną porównane dwa sposoby doprowadzania macierzy kwadratowej  $A$  do postaci górnotrójkątnej: metoda eliminacji Gaussa oraz rozkład  $QR$  z transformacją Householdera.

Oba te algorytmy zostaną wykorzystane do rozwiązywania układu równań

$$Ax = b$$

dla odwracalnej macierzy  $A$  oraz dowolnego wektora  $x$ .

W ramach testowania praktycznego zastosowania naszych algorytmów, macierz  $A$  będzie losowo wygenerowaną macierzą  $A \in GL_n(\mathbb{R})$  dla  $n \in \{20, 100, 400\}$ . Dalej losowany będzie wektor  $x$  i na tej podstawie wyliczymy wektor  $b$ . Stosowane przez nas algorytmy dostaną jedynie macierz  $A$  oraz wektor  $b$ , a zwrócą wektor  $x'$  mający być rozwiązaniem jak wyżej. Co więcej, dla algorytmu rozkładu  $QR$  jesteśmy w stanie sprawdzić poprawność obliczonej formy macierzy  $A$ . Błąd dla każdej z metod będzie obliczany jako

$$e = -\log(\|x' - x\|).$$

Dla metody Householdera, która produkuje macierz ortogonalną  $Q$  oraz górnotrójkątną  $R$ , takie, że  $A = QR$ , porównane również zostaną macierze

$$A - QR$$

$$Q^T A - R$$

$$Q^T Q - I$$

poprzez wyznaczenie normy macierzy  $B$ , gdzie  $B$  to jedna z powyższych macierzy. Macierze będące wynikami powyższych przekształceń powinny być macierzami zerowymi, co wynika z przekształcenia  $A = QR$  oraz ortogonalności macierzy  $Q$  (wtedy  $Q^{-1} = Q^T$ ). Do obliczania normy macierzy  $B$  wykorzystana będzie biblioteczna funkcja `norm(A, p)` mająca zwracać  $p$ -normę macierzy  $A$ . Ponieważ jesteśmy w przestrzeni Euklidesowej, interesować nas będzie  $p = 2$ .

Wszystkie obliczenia zostaną wykonane na `BigFloatach` w trzech precyzjach: 68, 419 oraz 2005.

## 2. Rozwiązywanie równań liniowych

Mając dany układ równań liniowych:

$$\begin{cases} a_{11}x_1 + a_{12}x_2 + \dots + a_{1n}x_n = b_1 \\ a_{21}x_1 + a_{22}x_2 + \dots + a_{2n}x_n = b_2 \\ \dots \\ a_{n1}x_1 + a_{n2}x_2 + \dots + a_{nn}x_n = b_n \end{cases}$$

możemy go opisać w postaci macierzy. Macierz główna tego układu równań to macierz zawierająca wszystkie współczynniki przy zmiennych  $[X]$ :

$$A = \begin{bmatrix} a_{11} & a_{12} & \dots & a_{1n} \\ a_{21} & a_{22} & \dots & a_{2n} \\ \dots & \dots & \dots & \dots \\ a_{n1} & a_{n2} & \dots & a_{nn} \end{bmatrix}.$$

Jeśli do macierzy głównej dołączymy wektor zawierający wszystkie wyrazy wolne  $[B]$ , to dostaniemy macierz rozszerzoną tego układu:

$$A|B = \left[ \begin{array}{cccc|c} a_{11} & a_{12} & \dots & a_{1n} & b_1 \\ a_{21} & a_{22} & \dots & a_{2n} & b_2 \\ \dots & \dots & \dots & \dots & \dots \\ a_{n1} & a_{n2} & \dots & a_{nn} & b_n \end{array} \right].$$

Zapisanie układu równań w postaci macierzowej ma wiele zalet. Jesteśmy w stanie w szybki sposób sprawdzić, czy równanie ma jednoznaczne rozwiązanie przez sprawdzenie czy wyznacznik macierzy głównej nie jest zerowy, gdyż jeśli  $AX = B$ , to  $A^{-1}AX = X = A^{-1}B$ . Musi więc istnieć macierz odwrotna. Sprawia to również, że zapis układu jest bardziej czytelny oraz pozwala ułatwić operowanie na takim układzie równań za pomocą komputera.

## 2.1. Eliminacja Gaussa

Metoda eliminacji Gaussa jest algorytmem stosowanym do rozwiązywania układu równań. Polega ona na doprowadzeniu macierzy do postaci schodkowej, tzn. zawierającej niezerowe wartości tylko na głównej przekątnej. W algorytmie dozwolone są tylko operacje na wierszach i kolumnach, czyli dodawanie lub odejmowanie od wiersza (kolumny) wielokrotności innego wiersza (kolumny) oraz zamienianie kolejności dwóch wierszy (kolumn).

Alternatywnie, na kursie algebry liniowej poznaliśmy metodę na odwracanie macierzy za pomocą eliminacji Gaussa. Wtedy z lewej strony wpisujemy oryginalną macierz, z prawej macierz identyczności i dokonując operacji wierszowych na całości staramy się doprowadzić lewą macierz do macierzy identyczności. Wtedy to co, powstanie z prawej strony będzie szukaną macierzą odwrotną. W poniższej pracy nie skorzystamy z tej wariacji na temat metody eliminacji Gaussa.

## 2.2. Rozkład $QR$

Każdą macierz  $A$   $m \times n$  o wyrazach rzeczywistych taką, że  $\text{rank}(A) = n$ , można zapisać jako  $A = QR$ , gdzie  $R$  jest macierzą górnortrójkątną, a  $Q$  ma kolumny ortogonalne. Ponieważ my będziemy rozważać macierze  $A$  będące reprezentacją jednoznacznych układów równań, to interesują nas tylko  $A \in GL_n(\mathbb{R})$ .

Zauważmy, że jeśli  $A$  ma niezerowy wyznacznik, to  $A$  nie może mieć liniowo zależnych kolumn. W takim razie, wektory  $a_1, \dots, a_n$  odpowiadające kolumnom  $A$  są bazą przestrzeni  $\mathbb{R}^n$  jako maksymalny możliwy układ wektorów liniowo niezależnych. Możemy na ich podstawie stworzyć bazę ortonormalną  $u_1, \dots, u_n$  przez proces Grama-Schmidta. Wtedy dla  $k = 1, \dots, n$

$$u_k = a_k - \sum_{i=1}^{k-1} \frac{\langle u_i, a_k \rangle}{\langle u_i, u_i \rangle} u_i.$$

Co więcej, dla dowolnego  $a_k$  z oryginalnej bazy możemy go zapisać za pomocą kombinacji liniowej wektorów z bazy ortonormalnej:

$$a_k = \sum_{i=1}^n c_i u_i = \sum_{i=1}^n c_i \sum_{j=1}^{i-1} \left[ a_k - \sum_{i=1}^{k-1} \frac{\langle u_i, a_k \rangle}{\langle u_i, u_i \rangle} u_i \right]$$

a ponieważ  $a_1, \dots, a_n$  były wektorami lnz, to dla  $i > k$   $c_i = 0$ . Niech  $r_k$  to będzie wektor zawierający współczynniki  $c_i$  dla wektora  $a_k$ :

$$r_k = \begin{bmatrix} c_1 \\ c_2 \\ \dots \\ c_k \\ 0 \\ \dots \\ 0 \end{bmatrix}$$

Czyli mamy, że

$$a_k = \begin{bmatrix} u_1 & u_2 & \dots & u_n \end{bmatrix} r_k$$

i dalej

$$A = \begin{bmatrix} u_1 & u_2 & \dots & u_n \end{bmatrix} \begin{bmatrix} r_1 & r_2 & \dots & r_n \end{bmatrix}.$$

Zauważamy, że  $R = \begin{bmatrix} r_1 & r_2 & \dots & r_n \end{bmatrix}$  to macierz górnotrójkątna, a  $Q$  to macierz ortogonalna.

Niech teraz  $A$  to macierz główna rozważanego układu równań,  $Q, R$  to macierze z jej rozkładu,  $X$  niech będzie wektorem wartości szukanych, a  $B$  niech będzie wektorem wyrazów wolnych. Wtedy

$$\begin{aligned} AX &= B \\ (QR)X &= B \end{aligned}$$

i ponieważ dla macierzy ortonormalnych mamy  $Q^{-1} = Q^T$ , to w prosty sposób możemy zamienić powyższy układ na

$$RX = Q^T B.$$

### 3. Algorytm eliminacji Gaussa

#### 3.1. Wyniki

Precyzja	$20 \times 20$	$100 \times 100$	$400 \times 400$
68	41.78	35.1	31.15
419	286.01	280.42	276.44
2005	1474.83	1471.28	1466.6

Tabela 1. Liczba cyfr znaczących normy wektora  $(Ax' - b)$ , gdzie  $x'$  to wektor uzyskany z metody eliminacji Gaussa.

W Tabeli 1 umieszczone zostały ilości cyfr znaczących wartości  $\|x' - x\|$ . Bez względu na precyzję obliczeń większe macierze dawały nieco mniej precyzyjne wyniki. Co ciekawe, różnica ta, w okolicach 4 cyfr znaczących, ulegała jedynie niezauważalnym zmianom między różnymi precyzjami obliczeń.

#### 3.2. Złożoność obliczeniowa

W zaimplementowanym przez nas algorytmie przy każdym kroku dokonujemy jednego dzielenia skalaru oraz jednego mnożenia przez skalar i odejmowania wektora o długości  $n$ , gdzie  $n$  to szerokość naszej macierzy. Czyli za każdym razem wykonujemy

$$a = 1 + 2n$$


operacji. Co więcej, w algorytmie zewnętrzna pętla wykonuje się  $(n - 1)$  razy, natomiast wewnętrzna -  $n + 1 - (i + 1)$  razy, gdzie  $i$  to jest aktualna pozycja zewnętrznej pętli. Czyli wykonujemy

$$\sum_{i=1}^{n-1} (n - i)a = \frac{a(n - 2)^2}{2} = \frac{(1 + 2n)(n - 2)^2}{2} = O(n^3)$$

operacji w trakcie całego algorytmu.

## 4. Transformacja Householdera

### 4.1. Podstawy teoretyczne

Transformacja Householdera [  : Householder transformation ] to liniowe przekształcenie poprzez odbicie punktu wokół płaszczyzny, lub hiperpłaszczyzny, która zawiera początek układu współrzędnych. Płaszczyzna wokół której obracamy jest zdefiniowana przez jednostkowy wektor  $u$  do niej normalny, a więc odbicie względem niej to  $x$  pomniejszony o dwa rzuty na  $u$ :

$$x' = x - 2u\langle x, u \rangle = x - 2u(u^*x)$$

co dla przestrzeni rzeczywistej wynosi

$$x' = x - 2u(u^T x).$$

Macierz tego odbicia to

$$P = I - 2uu^*$$

i jest ona Hermitowska:

$$P^* = (I - 2uu^*)^* = I^* - (2uu^*)^* = I - 2(uu^*)^* = I - 2(u^*)^*u^* = I - 2uu^* = P$$

oraz unitarna (czyli  $P^*P = PP^* = I$ ):

$$\begin{aligned} P^*P &= P^2 = (I - 2uu^*)^2 = I - 4uu^* + 4(uu^*)^2 = \\ &= I - 4uu^* + 4u(u^*u)u^* = I - 4uu^* + 4u\langle u, u \rangle u^* = \\ &= I - 4uu^* + 4uu^* = I \end{aligned}$$

a więc w przypadku rzeczywistym dostajemy macierz symetryczną i ortogonalną, czyli taką jakiej szukamy.

Niech teraz  $A$  będzie macierzą  $m \times m$ , której formę  $QR$  chcemy znaleźć, a  $a_1, \dots, a_m$  będą wektorami odpowiadającymi jej kolumnom. Dalej, niech  $e_1, \dots, e_m$  będą wektorami ze standardowej bazy przestrzeni  $\mathbb{R}^m$  i ustalmy

$$\begin{aligned} v &= a_1 - \|a_1\|e_1 \\ u &= \frac{v}{\|v\|}. \end{aligned}$$

Wektor  $u$  jest jednostkowym wektorem pewnej płaszczyzny przechodzącej przez początek układu współrzędnych, możemy więc dla niego znaleźć macierz Householdera

$$P'_1 = I - 2uu^*.$$

Zauważmy, że

$$P'_1 a_1 = \begin{pmatrix} \|a_1\| \\ 0 \\ 0 \\ \dots \\ 0 \end{pmatrix}$$

czyli zaczynamy tworzyć macierz górnotrójkątną. Proces transformacji Householdera możemy powtórzyć dla macierzy  $P_1 A$  bez pierwszej kolumny i wiersza, co da nam macierz  $P'_2$  która dla  $P'_2 a_2$  daje wektor niezerowy tylko na pierwszej współrzędnej. Jednak  $P'_2$  jest  $(m-1) \times (m-1)$ , więc musimy ją rozciągnąć,

choćby dodając identyczność w lewym górnym rogu. Rozciągając tę procedurę na przypadek ogólny, mamy

$$P_k = \begin{pmatrix} I_{k-1} & * \\ 0 & P'_k \end{pmatrix}$$

gdzie  $I_{k-1}$  to identyczność ale na  $\mathbb{R}^{k-1}$ .

Szukana przez nas macierz górnotrójkątna ma zatem postać

$$R = P_m \dots P_1 A$$

natomiast szukana macierz ortogonalna to

$$Q = (P_m \dots P_1)^{-1} = P_1^{-1} \dots P_m^{-1} = P_1 \dots P_m$$

z faktu, że każda z macierzy  $P_k$  jest unitarna i hermitowska.

Dla polepszenia wyników algorytmu definicja wektora  $v$  musi brać pod uwagę znak lewego górnego rogu macierzy (minora) którą będziemy poddawać transformacji Householdera. Chcemy zawsze mnożyć normę  $a_1$  przez znak przeciwny do wspomnianego elementu macierzy.

## 4.2. Wyniki

W Tabeli 2 umieszczona została ilość cyfr znaczących dla wektora  $\|x' - x\|$  w zależności od wielkości macierzy  $A$  oraz precyzji arytmetyki. Różnica między precyzją obliczeń dla różnych wielkości macierzy jest w okolicach 3 cyfr znaczących i zachowuje się między różnymi precyzjami, co jest wynikiem napawającym nadzieją. Z Tabeli 4 widzimy też, że otrzymana przez nas macierz  $Q$  jest bardzo bliska bycia macierzą ortogonalną, przynajmniej pod względem normy, to znaczy

$$\|Q\| = \sup_{\|x\| \leq 1} \frac{\|Qx\|}{\|x\|}$$

jest niewiele większe od zera.

Co ciekawe, liczba cyfr znaczących różnicy między otrzymanym przez nas rozwiązaniem równania a rozwiązaniem poprawnym (Tabela 2) oraz norma macierzy będących przekształceniem  $A = QR$  (Tabele 3,4,5) jest na tym samym poziomie precyzji.

Precyzja	$20 \times 20$	$100 \times 100$	$400 \times 400$
68	42.66	39.46	36.86
419	286.19	282.58	280.144
2005	1476.08	1473.03	1470.28

Tablusia 2. Liczba cyfr znaczących normy wektora  $(Ax' - b)$ , gdzie  $x'$  to wektor uzyskany z metody transformacji Householdera.

## 4.3. Złożoność obliczeniowa

W naszej implementacji najpierw wykonujemy  $n$  iteracji zewnętrznej pętli. Wewnątrz której mamy pętle po kolumnach oraz pętle po wierszach.

W każdym kroku obu tych pętli wykonujemy stałą liczbę następujących operacji: mnożenie wektora przez skalar, dodawania/odejmowanie wektorów, mnożenie wektora przez wektor o takim rozmiarze ale, który został transponowany. Wszystkie te operacje mają złożoność  $O(n)$ . I jako że jest ich stała liczba, to każda iteracja tych pętli ma złożoność  $O(n)$

Precyzja	$20 \times 20$	$100 \times 100$	$400 \times 400$
68	41.96	40.68	38.67
419	286.50	284.19	282.22
2005	1476.00	1471.28	1470.08

Tabela 3. Rząd wielkości normy macierzy  $A - QR$  (obliczany z  $-\log \|A - QR\|$ ).

Precyzja	$20 \times 20$	$100 \times 100$	$400 \times 400$
68	43.48	42.18	40.85
419	286.91	285.47	284.14
2005	1477.15	1475.61	1474.25

Tabela 4. Rząd wielkości normy macierzy  $Q^T Q - I$  (obliczany z  $-\log \|Q^T Q - I\|$ ).

Precyzja	$20 \times 20$	$100 \times 100$	$400 \times 400$
68	43.20	40.97	38.91
419	286.59	284.3	282.25
2005	1476.83	1474.34	1472.37

Tabela 5. Rząd wielkości normy macierzy  $Q^T A - R$  (obliczany z  $-\log \|Q^T A - R\|$ ).

Żeby użyć tego algorytmu do rozwiązania układu równań, trzeba potem użyć algorytmu Backward substitution, ma on złożoność  $O(n^2)$ .

Czyli złożoność całego algorytmu to

$$\sum_{i=1}^n \left( \sum_{j=i}^n O(N) + \sum_{j=i}^n O(N) \right) + O(n^2) = O(n^3)$$

## 5. Podsumowanie

W porównaniu Tabeli 1 oraz Tabeli 2 widzimy niewielką różnicę na zysk algorytmu wykorzystującego transformację Householdera. Miał on ogółem większą precyzję niż algorytm wykorzystujący metodę eliminacji Gaussa. Różnica w precyzji między macierzami różnej wielkości była również mniejsza w rozkładzie  $QR$ . Sugeruje to nieco większa stabilność numeryczną metody Householdera.

Złożoność obu algorytmów była taka sama - oba wykonują  $O(n^3)$  operacji, jednak przy metodzie Householdera potrzebujemy większej liczby zmiennych. Z drugiej strony, metoda Householdera zwraca więcej informacji o macierzy  $A$  niż metoda eliminacji Gaussa.

Jeżeli potrzebujemy bardzo wysokiej precyzji obliczeń lepszym wyborem jest metoda Householdera. Jednak jeśli liczy się prostota implementacji oraz zmniejszenie ilości używanej przez algorytm pamięci, to metoda eliminacji Gaussa jest bardzo dobrym rozwiązaniem.

