NHẬP MÔN ĐẠI SỐ TUYẾN TÍNH TÍNH TOÁN

TS. Nguyen Thanh Binh
ThS. Tran Thi My Huynh

Tp. Hồ Chí Minh - 2020

Mục lục

Chương 1

Các khái niệm cơ bản

1.1 Ma trận và vector

1.1.1 Ma trận

Cho $\mathbb{R}$ là tập hợp các số thực. Khi đó, $\mathbb{R}^{m\times n}$ là không gian vector của các ma trận thực có m dòng và n cột

$$A \in \mathbb{R}^{m\times n} \Longleftrightarrow A = (a_{ij}) = \begin{bmatrix} a_{11} & \dots & a_{1n} \\ \vdots & & \vdots \\ a_{m1} & \dots & amn \end{bmatrix},\ a_{ij} \in \mathbb{R}$$

Ngoài ra, chúng ta còn sử dụng $[A]_{ij}$ hay $A(i,j)$ để chỉ những phần tử của một ma trận.

1.1.2 Các phép toán trên ma trận

Các phép toán cơ bản trên ma trận gồm:

- Ma trận chuyển vị ($\mathbb{R}^{m\times n} \to \mathbb{R}^{n\times m}$),

$$C = A^T \Longrightarrow c_{ij} = a_{ji}$$

- Cộng hai ma trận ($\mathbb{R}^{m\times n} \times \mathbb{R}^{m\times n} \to \mathbb{R}^{m\times n}$),

$$C = A + B \Longrightarrow c_{ij} = a_{ij} + b_{ij}$$

- Nhân một số với ma trận ($\mathbb{R} \times \mathbb{R}^{m\times n} \to \mathbb{R}^{m\times n}$),

$$C = \alpha A \Longrightarrow c_{ij} = \alpha a_{ij},$$

- Nhân hai ma trận ($\mathbb{R}^{m\times p} \times \mathbb{R}^{p\times n} \to \mathbb{R}^{m\times n}$),

$$C = AB \Longrightarrow c_{ij} = \sum_{k=1}^{r} a_{ik} b_{kj}.$$

- Nhân ma trận theo từng điểm ($\mathbb{R}^{m\times n} \times \mathbb{R}^{m\times n} \to \mathbb{R}^{m\times n}$),

$$C = A. * B \Longrightarrow c_{ij} = a_{ij} b_{ij}$$

- Phép chia theo từng điểm ($\mathbb{R}^{m\times n} \times \mathbb{R}^{m\times n} \to \mathbb{R}^{m\times n}$),

$$C = A./B \Longrightarrow c_{ij} = a_{ij}/b_{ij}.$$

1.1.3 Khái niệm vector

Cho $\mathbb{R}^n$ là không gian vector của các vector thực có n phần tử

$$x \in \mathbb{R}^n \Longleftrightarrow x = \begin{bmatrix} x_1 \\ \vdots \\ x_n \end{bmatrix}, x_i \in \mathbb{R}$$

trong đó, x_i như là thành phần thứ i của vector x.

Chú ý, ta đồng nhất $\mathbb{R}^n$ với $\mathbb{R}^{n\times 1}$ nên mỗi phần tử của $\mathbb{R}^n$ là một vector *cột*. Mặc khác, những phần tử của $\mathbb{R}^{1\times m}$ là những vector *dòng*:

$$x \in \mathbb{R}^{1\times n} \Longleftrightarrow x = [x_1, \ldots, x_n].$$

Nếu x là một vector cột thì $y = x^T$ là một vector dòng.

1.1.4 Các phép toán trên vector

Cho $a \in \mathbb{R}, x \in \mathbb{R}^n$ và $y \in \mathbb{R}^n$. Khi đó,các phép toán cơ bản trên vector gồm:

- Nhân một số với một vector ,

$$z = ax \Longrightarrow z_i = ax_i,$$

- Cộng hai vector

$$z = x + y \Longrightarrow z_i = x_i + y_i,$$

- Tích vô hướng của hai vector (hay *tích trong*),

$$c = x^T y \Rightarrow c = \sum_{i=1}^{n} x_i y_i,$$

- Nhân vector theo từng điểm

$$z = x.*y \Rightarrow z_i = x_i y_i$$

- Chia vector theo từng điểm

$$z = x./y \Rightarrow z_i = x_i/y_i$$

1.1.5 Tích vô hướng của 2 vector trong Matlab

Cho $x, y \in \mathbb{R}^n$, thuật toán sau sẽ tính tích vô hướng $c = x^T y$

Thuật toán 1.1 (Tích vô hướng)

```
1: c = 0
2: for i = 1 : n do
3:     c = c + x(i)y(i)
4: end for
```

1.1.6 Dấu hai chấm (:)

Cho $A \in \mathbb{R}^{m\times n}$. Khi đó, dòng thứ k của ma trận A

$$A(k,:) = [a_{k1}, \ldots, a_{kn}].$$

và cột thứ k của A

$$A(:,k) = \begin{bmatrix} a_{1k} \\ \ldots \\ a_{mk} \end{bmatrix}.$$

1.1.7 Ma trận phức

Không gian vectơ của các ma trận phức có m dòng và n cột được ký hiệu bởi $\mathbb{C}^{m\times n}$. Phép nhân với vô hướng, phép cộng và phép nhân của các ma trận phức tương ứng như trong ma trận thực. Tuy nhiên, phép chuyển vị trở thành chuyển vị liên hợp

$$C = A^H \Rightarrow c_{ij} = \overline{a_{ji}}.$$

Không gian vectơ của các vectơ phức n chiều được ký hiệu là $\mathbb{C}^n$. Tích vô hướng của hai vector x và y được cho bởi

$$s = x^H y = \sum_{i=1}^{n} \overline{x_i} y_i.$$

Cho $A = B + iC \in \mathbb{C}^{m\times n}$, phần thực và phần ảo của A tương ứng là $Re(A) = B$ và $Im(A) = C$. Liên hợp của A là ma trận $\overline{A} = (\overline{a}_{ij})$.

1.1.8 Các ma trận dải(band)

Một ma trận là thưa nếu phần lớn các phần tử của ma trận này là 0. Ma trận dải (band) là một trường hợp đặc biệt của ma trận thưa. Ta nói $A \in \mathbb{R}^{m\times n}$ có p băng thông (bandwith) dưới nếu $a_{ij} = 0$ với $i > j + p$ và q băng thông trên nếu $a_{ij} = 0$ với $j > i + q$. Ví dụ cho ma trận 8×5 có 1 băng thông dưới và 2 băng thông trên

$$\begin{bmatrix} \times & \times & \times & 0 & 0 \\ \times & \times & \times & \times & 0 \\ 0 & \times & \times & \times & \times \\ 0 & 0 & \times & \times & \times \\ 0 & 0 & 0 & \times & \times \\ 0 & 0 & 0 & 0 & \times \\ 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 \end{bmatrix}$$

trong đó $\times$ là ký hiệu các phần tử khác 0. Các cấu trúc dải xuất hiện thường xuyên được liệt kê trong Bảng 1.1.

1.1.9 Lưu trữ dải

Cho $A \in \mathbb{R}^{n\times n}$ có p băng thông dưới và q băng thông trên và giả sử $p, q < n$. Ma trận A có thể được lưu trữ trong một mảng $A.band$ có kích thước $(p + q + 1) \times n$ với quy ước

$$a_{ij} = A.band(i - j + q + 1, j) \tag{1.1.1}$$

Bảng 1.1: Thuật ngữ dải cho các ma trận $m \times n$

Loại ma trận	Băng thông dưới	Băng thông trên
Đường chéo	0	0
Tam giác trên	0	$n-1$
Tam giác dưới	$m-1$	0
Ba đường chéo	1	1
Hai đường chéo trên	0	1
Hai đường chéo dưới	1	0
Hessenberg trên	1	$n-1$
Hessenberg dưới	$m-1$	1

với mọi (i, j) nằm trong dải, ví dụ,

$$\begin{bmatrix} a_{11} & a_{12} & a_{13} & 0 & 0 & 0 \\ a_{21} & a_{22} & a_{23} & a_{24} & 0 & 0 \\ 0 & a_{32} & a_{33} & a_{34} & a_{35} & 0 \\ 0 & 0 & a_{43} & a_{44} & a_{45} & a_{46} \\ 0 & 0 & 0 & a_{54} & a_{55} & a_{56} \\ 0 & 0 & 0 & 0 & a_{65} & a_{66} \end{bmatrix} \Rightarrow \begin{bmatrix} * & * & a_{13} & a_{24} & a_{35} & a_{46} \\ * & a_{12} & a_{23} & a_{34} & a_{45} & a_{56} \\ a_{11} & a_{22} & a_{33} & a_{44} & a_{55} & a_{66} \\ a_{21} & a_{32} & a_{43} & a_{54} & a_{65} & * \end{bmatrix}.$$

với "*" là các hệ số không được sử dụng.

1.1.10 Ma trận đường chéo

Các ma trận với 0 băng thông dưới và 0 băng thông trên là ma trận đường chéo. Nếu $D \in \mathbb{R}^{m\times n}$ là ma trận đường chéo thì khi đó

$$D = diag(d_1, \ldots, d_q), q = min(m, n) \Longleftrightarrow d_i = d_{ii}.$$

Nếu $D = diag(d) \in \mathbb{R}^{n\times n}$ và $x \in \mathbb{R}^n$ thì $Dx = d. * x$. Nếu $A \in \mathbb{R}^{m\times n}$ thì phép nhân trái với $D = diag(d_1, \ldots, d_m) \in \mathbb{R}^{m\times m}$,

$$B = DA \Longleftrightarrow B(i, :) = d_i \cdot A(i, :), i = 1 : m$$

và phép nhân phải với $D = diag(d_1, \ldots, d_m) \in \mathbb{R}^{n\times n}$,

$$B = AD \Longleftrightarrow B(:, j) = d_j \cdot A(:, j), j = 1 : n.$$

1.1.11 Ma trận đối xứng

Ma trận $A \in \mathbb{R}^{n\times n}$ là đối xứng nếu $A^T = A$ và là phản đối xứng nếu $A^T = -A$. Tương tự, ma trận $A \in \mathbb{C}^{n\times n}$ là Hermit nếu $A^H = A$ và là phản Hermit nếu $A^H = -A$.

Ví dụ: Ma trận đối xứng:

$$\begin{bmatrix} 1 & 2 & 3 \\ 2 & 4 & 5 \\ 3 & 5 & 6 \end{bmatrix},$$

Ma trận Hermit:

$$\begin{bmatrix} 1 & 2-3i & 4-5i \\ 2+3i & 6 & 7-8i \\ 4+5i & 7+8i & 9 \end{bmatrix},$$

Ma trận phản đối xứng:

$$\begin{bmatrix} 0 & -2 & 3 \\ 2 & 0 & -5 \\ -3 & 5 & 0 \end{bmatrix},$$

Ma trận phản Hermit:

$$\begin{bmatrix} i & -2+3i & -4+5i \\ 2+3i & 6i & -7+8i \\ 4+5i & 7+8i & 9i \end{bmatrix}.$$

1.1.12 Ma trận hoán vị và ma trận đơn vị

Ta ký hiệu ma trận đơn vị $n \times n$ là I_n, ví du,

$$I_4 = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

Ta sử dụng ký hiệu e_i để chỉ cột thứ i của I_n. Nếu các dòng của I_n được sắp xếp lại thì ma trận kết quả được biểu diễn như là một ma trận hoán vị. Ví dụ,

$$P = \begin{bmatrix} 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 1 \\ 0 & 0 & 1 & 0 \\ 1 & 0 & 0 & 0 \end{bmatrix}. \tag{1.1.2}$$

1.1.13 Ma trận khối

Ma trận khối là ma trận mà các phần tử cũng là các ma trận. Chẳng hạn, một ma trận 8×15 của các vô hướng có thể được xem như là ma trận khối 2×3 với các phần tử là các ma trận 4×5.
Cho $A \in \mathbb{R}^{m \times n}$, *phân tích dạng dòng* của A là một mảng các vector dòng:

$$\iff A = \begin{bmatrix} r_1^T \\ \vdots \\ r_m^T \end{bmatrix}, r_k \in \mathbb{R}^n. \tag{1.1.3}$$

Ví dụ: phân tích dạng dòng của ma trận $\begin{bmatrix} 1 & 2 \\ 3 & 4 \\ 5 & 6 \end{bmatrix}$, ta xem A như là một tập hợp của các vector dòng với

$$r_1^T = [1 \quad 2], \qquad r_2^T = [3 \quad 4], \qquad r_3^T = [5 \quad 6].$$

Tương tự, ta cũng có *phân tích dạng cột* của ma trận A là một tập hợp các vector cột:

$$A \in \mathbb{R}^{m \times n} \iff A = [c_1 | \dots | c_n], c_k \in \mathbb{R}^m. \tag{1.1.4}$$

Ở ví dụ trên, ta dặt c_1 và c_2 lần lượt là cột thứ nhất và cột thứ hai của A:

$$c_1 = \begin{bmatrix} 1 \\ 3 \\ 5 \end{bmatrix}, \qquad c_2 = \begin{bmatrix} 2 \\ 4 \\ 6 \end{bmatrix}.$$

Phân tích dạng dòng và cột của một ma trận là các trường hợp đặc biệt của việc tạo khối ma trận. Tổng quát, ta có phân tích dạng dòng và cột của ma trận A có m dòng và n cột

$$A = \begin{bmatrix} A_{11} & \dots & A_{1r} \\ \vdots & & \vdots \\ A_{q1} & \cdots & A_{qr} \end{bmatrix} \begin{matrix} m_1 \\ \\ m_q \end{matrix}$$
$$\begin{matrix} n_1 & & n_r \end{matrix}$$

trong đó $m_1 + \ldots + m_q = m, n_1 + \ldots + n_r = n$, và $A_{\alpha\beta}$ ký hiệu khối (α, β) (ma trận con) có số chiều là $m_\alpha \times n_\beta$ và ta nói $A = (A_{\alpha\beta})$ là một ma trận khối $q \times r$.

Ta sử dụng các số hạng này để miêu tả các cấu trúc dải phổ biến cho các ma trận với các khối tương tự như các phần tử vô hướng. Do đó,

$$diag(A_{11}, A_{22}, A_{33}) = \begin{bmatrix} A_{11} & 0 & 0 \\ 0 & A_{22} & 0 \\ 0 & 0 & A_{33} \end{bmatrix}$$

là đường chéo khối,

$$L = \begin{bmatrix} L_{11} & 0 & 0 \\ L_{21} & L_{22} & 0 \\ L_{31} & L_{32} & L_{33} \end{bmatrix}, \; U = \begin{bmatrix} U_{11} & U_{12} & U_{13} \\ 0 & U_{22} & U_{23} \\ 0 & 0 & U_{33} \end{bmatrix}, \; T = \begin{bmatrix} T_{11} & T_{12} & 0 \\ T_{21} & T_{22} & T_{23} \\ 0 & T_{32} & T_{33} \end{bmatrix},$$

lần lượt là *ma trận tam giác khối dưới, tam giác khối trên*, và *ba đường chéo khối.*

1.1.14 Các phép toán ma trận khối

Các ma trận khối có thể được nhân với vô hướng và chuyển vị:

$$\mu \begin{bmatrix} A_{11} & A_{12} \\ A_{21} & A_{22} \\ A_{31} & A_{32} \end{bmatrix} = \begin{bmatrix} \mu A_{11} & \mu A_{12} \\ \mu A_{21} & \mu A_{22} \\ \mu A_{31} & \mu A_{32} \end{bmatrix},$$
$$\begin{bmatrix} A_{11} & A_{12} \\ A_{21} & A_{22} \\ A_{31} & A_{32} \end{bmatrix}^T = \begin{bmatrix} A_{11}^T & A_{21}^T & A_{31}^T \\ A_{12}^T & A_{22}^T & A_{32}^T \end{bmatrix}.$$

Chú ý, chuyển vị của khối (i, j) trở thành khối (j, i). Tương tự, ta có phép cộng hai ma trận khối

$$\begin{bmatrix} A_{11} & A_{12} \\ A_{21} & A_{22} \\ A_{31} & A_{32} \end{bmatrix} + \begin{bmatrix} B_{11} & B_{12} \\ B_{21} & B_{22} \\ B_{31} & B_{32} \end{bmatrix} = \begin{bmatrix} A_{11} + B_{11} & A_{12} + B_{12} \\ A_{21} + B_{21} & A_{22} + B_{22} \\ A_{31} + B_{31} & A_{32} + B_{32} \end{bmatrix}.$$

Phép nhân hai ma trận khối cần nhiều điều kiện về số chiều. Chẳng hạn, nếu

$$\begin{bmatrix} A_{11} & A_{12} \\ A_{21} & A_{22} \\ A_{31} & A_{32} \end{bmatrix} \begin{bmatrix} B_{11} & B_{12} \\ B_{21} & B_{22} \end{bmatrix} = \begin{bmatrix} A_{11}B_{11} + A_{12}B_{21} & A_{11}B_{12} + A_{12}B_{22} \\ A_{21}B_{11} + A_{22}B_{21} & A_{21}B_{12} + A_{22}B_{22} \\ A_{31}B_{11} + A_{32}B_{21} & A_{31}B_{12} + A_{32}B_{22} \end{bmatrix}$$

thì số cột của A_{11}, A_{21} và A_{31} phải bằng với số dòng của cả B_{11} và B_{12}. Tương tự, số dòng của A_{12}, A_{22} và A_{32} phải bằng với số dòng của cả B_{21} và B_{22}.

Mỗi khi cộng hoặc nhân một ma trận khối thì số dòng và cột của các khối thỏa mãn tất cả các ràng buộc cần thiết. Trong trường hợp đó, ta nói các toán hạng được *phân tích đúng với* định lý theo sau.

Định lý 1.1.1 *Nếu*

$$A = \begin{bmatrix} A_{11} & \dots & A_{1s} \\ \vdots & & \vdots \\ A_{q1} & \cdots & A_{qs} \end{bmatrix} \begin{matrix} m_1 \\ \\ m_q \end{matrix}, \qquad B = \begin{bmatrix} B_{11} & \dots & B_{1r} \\ \vdots & & \vdots \\ B_{s1} & \cdots & B_{sr} \end{bmatrix} \begin{matrix} p_1 \\ \\ p_s \end{matrix}$$
$$\begin{matrix} p_1 & & p_s \end{matrix} \qquad \begin{matrix} n_1 & & n_r \end{matrix}$$

và phân tích tích $C = AB$ như sau,

$$C = \begin{bmatrix} C_{11} & \dots & C_{1r} \\ \vdots & & \vdots \\ C_{q1} & \cdots & C_{qr} \end{bmatrix} \begin{matrix} m_1 \\ \\ m_q \end{matrix}$$
$$\begin{matrix} n_1 & & n_r \end{matrix}$$

thì for $\alpha = 1 : q$ và $\beta = 1 : r$ *ta có* $C_{\alpha\beta} = \sum_{\gamma=1}^{s} A_{\alpha\gamma} B_{\gamma\beta}$.

Chứng minh Giả sử $1 \leqslant \alpha \leqslant q$ và $1 \leqslant \beta \leqslant r$. Đặt $M = m_1 + \ldots + m_{\alpha-1}$ và $N = n_1 + \ldots + n_{\beta-1}$. Nếu $1 \leqslant i \leqslant m_\alpha$ và $1 \leqslant j \leqslant n_\beta$ thì

$$\begin{aligned} [C_{\alpha\beta}]_{ij} &= \sum_{k=1}^{p_1+\ldots+p_s} a_{M+i,k} b_{k,N+j} = \sum_{\gamma=1}^{s} \sum_{p_1+\ldots+p_{\gamma-1}+1}^{p_1+\ldots+p_\gamma} a_{M+i,k} b_{k,N+j} \\ &= \sum_{\gamma=1}^{s} \sum_{k=1}^{p_\gamma} [A_{\alpha\gamma}]_{ik} [B_{\gamma\beta}]_{kj} = \sum_{\gamma=1}^{s} [A_{\alpha\gamma} B_{\gamma\beta}]_{ij} = \left[\sum_{\gamma=1}^{s} A_{\alpha\gamma} B_{\gamma\beta} \right]_{ij}. \end{aligned}$$

Do đó, $C_{\alpha\beta} = A_{\alpha,1} B_{1,\beta} + \ldots + A_{\alpha,s} B_{s,\beta}$.

Nếu $A_{11}B_{11} + A_{12}B_{21} \neq B_{11}A_{11} + B_{21}A_{12}$ thì thao tác trên ma trận khối chính là thao tác trên ma trận ban đầu với a_{ij} và b_{ij} được viết như là A_{ij} và B_{ij}.

1.1.15 Các ma trận con

Cho $A \in \mathbb{R}^{m\times n}$. Nếu $\alpha = [\alpha_1, \ldots, \alpha_s]$ và $\beta = [\beta_1, \ldots, \beta_t]$ là các vector nguyên với các phần tử phân biệt thỏa mãn $1 \leqslant \alpha_i \leqslant m$ và $1 \leqslant \beta_i \leqslant n$ thì

$$A(\alpha, \beta) = \begin{bmatrix} a_{\alpha_1,\beta_1} & \cdots & a_{\alpha_1,\beta_t} \\ \vdots & \ddots & \vdots \\ a_{\alpha_s,\beta_1} & \cdots & a_{\alpha_s,\beta_t} \end{bmatrix}$$

là ma trận con của A có s dòng và t cột. Ví dụ, cho $A \in \mathbb{R}^{8\times 6}$, $\alpha = \begin{bmatrix} 2 & 4 & 6 & 8 \end{bmatrix}$, và $\beta = \begin{bmatrix} 4 & 5 & 6 \end{bmatrix}$,

$$A(\alpha, \beta) = \begin{bmatrix} a_{24} & a_{25} & a_{26} \\ a_{44} & a_{45} & a_{46} \\ a_{64} & a_{65} & a_{66} \\ a_{84} & a_{85} & a_{86} \end{bmatrix}.$$

Nếu $\alpha = \beta$ thì $A(\alpha, \beta)$ là *ma trận con chính (principal submatrix)*. Nếu $\alpha = \beta = 1 : k$ và $1 \leqslant k \leqslant min\{m, n\}$ thì $A(\alpha, \beta)$ là *ma trận con chính dẫn đầu (leading principal submatrix)*

Nếu $A \in \mathbb{R}^{m\times n}$ và

$$A = \begin{bmatrix} A_{11} & \dots & A_{1s} \\ \vdots & & \vdots \\ A_{q1} & \cdots & A_{qs} \end{bmatrix} \begin{matrix} m_1 \\ \\ m_q \end{matrix}$$
$$\begin{matrix} n_1 & & n_s \end{matrix}$$

thì kí hiệu dấu hai chấm có thể được sử dụng để xác định các khối riêng biệt. Đặc biệt,

$$A_{ij} = A(\tau + 1 : \tau + m, \mu + 1 : \mu + n_j)$$

trong đó $\tau = m_1 + \ldots + m_{i-1}$ và $\mu = n_1 + \ldots + n_{j-1}$.

1.2 Phép nhân ma trận với vector

1.2.1 Định nghĩa

Cho x là một vector cột n chiều và cho A là ma trận có m dòng và n cột. Khi đó tích ma trận với vector $b = Ax$ là một vector cột m chiều xác định như sau:

$$b_i = \sum_{j=1}^{n} a_{ij}x_j, \quad i = 1, \ldots, m. \tag{1.2.1}$$

Ánh xạ $x \mapsto Ax$ là *tuyến tính*, nghĩa là $x, y \in \mathbb{C}^n$ và $\alpha \in \mathbb{C}$ bất kì,

$$A(x + y) = Ax + Ay,$$
$$A(\alpha x) = \alpha Ax.$$

Ngược lại, mọi ánh xạ tuyến tính từ $\mathbb{C}^n$ vào $\mathbb{C}^m$ có thể được biểu diễn như phép nhân ma trận $m \times n$.

1.2.2 Nhân ma trận với vector

Cho a_j là cột thứ j của A và là một vector m chiều. Khi đó 1.2.1 có thể được viết lại

$$b = Ax = \sum_{j=1}^{n} x_j a_j. \tag{1.2.2}$$

Phương trình này có thể được viết dưới dạng như sau:

$$[b] = [a_1|a_2|\ldots|a_n] \begin{bmatrix} x_1 \\ x_2 \\ \vdots \\ x_n \end{bmatrix} = x_1 [a_1] + x_2 [a_2] + \ldots + x_n [a_n].$$

1.2.3 Ví dụ: ma trận Vandermonde

Cố định một chuỗi các số $\{x_1, x_2, \ldots, x_m\}$. Nếu p, q là các đa thức bậc nhỏ hơn n và α là một vô hướng, thì $p + q$ và αp cũng là các đa thức bậc nhỏ hơn n. Hơn nữa, các giá trị của các đa thức này tại các điểm x_i thỏa mãn các tính chất tuyến tính sau:

$$(p + q)(x_i) = p(x_i) + q(x_i)$$
$$(\alpha p)(x_i) = \alpha(p(x_i)).$$

Do đó ánh xạ từ các vector của các hệ số của các đa thức p bậc nhỏ hơn n tới các vector $(p(x_1), p(x_2), \ldots, p(x_m))$ của các giá trị đa thức được lấy mẫu là tuyến tính. Ánh xạ tuyến tính bất kì có thể được biểu diễn như phép nhân ma trận. Thực vậy, nó biểu diễn bằng ma trận Vandermonde

$$A = \begin{bmatrix} 1 & x_1 & x_1^2 & \ldots & x_1^{n-1} \\ 1 & x_2 & x_2^2 & \ldots & x_2^{n-1} \\ \vdots & \vdots & \vdots & & \vdots \\ 1 & x_m & x_m^2 & \ldots & x_m^{n-1} \end{bmatrix}.$$

Nếu c là vector cột của các hệ số của p,

$$c = \begin{bmatrix} c_0 \\ c_1 \\ c_2 \\ \vdots \\ c_{n-1} \end{bmatrix}, \quad p(x) = c_0 + c_1 x + c_2 x^2 + \ldots + c_{n-1} x^{n-1},$$

thì tích Ac, với mọi i từ 1 tới m

$$(Ac)_i = c_0 + c_1 x_i + c_2 x_i^2 + \ldots + c_{n-1} x_i^{n-1} = p(x_i). \tag{1.2.3}$$

Trong ví dụ này, tích ma trận với vector Ac không cần được thông qua tổng m vô hướng phân biệt, mỗi vô hướng cho một tổ hợp tuyến tính khác nhau của các phần tử của c, như 1.2.1. Hơn nữa, A có thể được xem như là một ma trận cột, mỗi cột cho các giá trị được lấy mẫu của một đơn thức,

$$A = \left[1 | x | x^2 | \ldots | x^{n-1}\right], \tag{1.2.4}$$

và tích Ac được hiểu như là tổng của các vector đơn trong dạng (1.2.2) mà nó là một tổ hợp tuyến tính của các đơn thức này,

$$Ac = c_0 + c_1 x + c_2 x^2 + \ldots + c_{n-1} x^{n-1} = p(x).$$

1.2.4 Nhân ma trận với ma trận

Cho A là ma trận $l \times m$ và C là ma trận $m \times n$, thì $B = AC$ là ma trận $l \times n$, với các phần tử xác định bởi

$$b_{ij} = \sum_{k=1}^{m} a_{ik} c_{kj} \tag{1.2.5}$$

với b_{ij}, a_{ik} và c_{kj} tương ứng là các phần tử của B, A và C. Được viết dưới dạng cột, tích

$$[b_1 | b_2 | \ldots | b_n] = [a_1 | a_2 | \ldots | a_m] [c_1 | c_2 | \ldots | c_n],$$

và (1.2.5) trở thành

$$b_j = A c_j = \sum_{k=1}^{m} c_{kj} a_k. \tag{1.2.6}$$

Do đó b_j là tổ hợp tuyến tính của các cột a_k với các hệ số c_{kj}.

Ví dụ 1.2.1. (Tích ngoài). Tích của vector cột u có m chiều với vector dòng v có n chiều; kết quả là một ma trận $m \times n$

$$[u] \begin{bmatrix} v_1 & v_2 & \ldots & v_n \end{bmatrix} = [v_1 u | v_2 u | \ldots | v_n u] = \begin{bmatrix} v_1 u_1 & \ldots & v_n u_1 \\ \vdots & & \vdots \\ v_1 u_m & \ldots & v_n u_m \end{bmatrix}.$$

Các cột là bội của cùng vector u, và tương tự, các dòng là bội của cùng vector v.

Ví dụ 1.2.2. Xét $B = AR$, với R là ma trận tam giác trên $n \times n$ mà $r_{ij} = 1$ với $i \leqslant j$ và $r_{ij} = 0$ với $i > j$. Tích này có thể được viết

$$[b_1|\dots|b_n] = [a_1|\dots|a_n] \begin{bmatrix} 1 & \dots & 1 \\ & \ddots & \vdots \\ & & 1 \end{bmatrix}.$$

Công thức theo dạng cột trong (1.2.6) cho

$$b_j = Ar_j = \sum_{k=1}^{j} a_k. \tag{1.2.7}$$

Cột thứ j của B là tổng của j cột đầu tiên của A.

1.2.5 Range và không gian đầy đủ

Vùng (range) của ma trận A, được viết $range(A)$, là tập hợp các vector có dạng Ax với x bất kỳ. Công thức (1.2.2) cho ta một đặc trưng của $range(A)$.

Định lý 1.2.1 *$range(A)$ là không gian được sinh bởi các cột của A.*

Chứng minh Do (1.2.2), Ax bất kỳ là một tổ hợp tuyến tính các cột của A. Ngược lại, một vector y bất kỳ trong không gian sinh bởi các cột của A có thể được viết như là một tổ hợp tuyến tính của các cột, $y = \sum_{j=1}^{n} x_j a_j$. Do đó, y nằm trong $range(A)$.

Trong Định lý 1.2.1, vùng của ma trận A cũng được gọi là *không gian cột của A*.

Không gian đầy đủ của $A \in \mathbb{C}^{m\times n}$, được viết là $null(A)$, là tập hợp các vector x thỏa mãn $Ax = 0$, với 0 là vector không trong $\mathbb{C}^m$. Các phần tử của mỗi vector $x \in null(A)$ cho khai triển các hệ số của 0 như là một tổ hợp tuyết tính các cột của ma trận A: $0 = x_1a_1 + x_2a_2 + \dots + x_na_n$.

1.2.6 Hạng

Hạng cột (column rank) của một ma trận là số chiều không gian cột của nó. Tương tự, *hạng dòng (row rank)* của một ma trận là số chiều của không gian sinh bởi các dòng của nó. Hạng dòng thường bằng với hạng cột nên để đơn gian ta gọi là *hạng* của một ma trận.

Ma trận *hạng đầy đủ (full rank)* $m \times n$ là ma trận có hạng có thể lớn nhất (nhỏ hơn m và n). Nghĩa là một ma trận hạng đầy đủ với $m \geqslant n$ phải có n cột độc lập tuyến tính. Ma trận như vậy cũng được xác định bởi tính chất mà ánh xạ xác định nó là đơn ánh.

Định lý 1.2.2 *Một ma trận $A \in \mathbb{C}^{m\times n}$ với $m \geqslant n$ có hạng đầy đủ nếu và chỉ nếu nó ánh xạ 2 vector không phân biệt thành cùng một vector.*

Chứng minh ($\Longrightarrow$) Nếu A là một ma trận hạng đầy đủ thì các cột của nó là độc lập tuyến tính, nên chúng hình thành một cơ sở cho $range(A)$. Nghĩa là mọi $b \in range(A)$ có duy nhất mở rộng tuyến tính các cột của A. Do đó, theo (1.2.2), mọi $b \in range(A)$ có duy nhất x sao cho $b = Ax$.

($\Longleftarrow$) Ngược lại, nếu A không có hạng đầy đủ thì các cột của a_j của nó là phụ thuộc tuyến tính, và $\sum_{j=1}^{n} c_j a_j = 0$ là một tổ hợp tuyến tính không tầm thường. Vector c khác 0 hình thành từ các hệ số c_j thỏa $Ac = 0$. Nhưng khi đó A ánh xạ các vector phân biệt thành cùng một vector vì với x bất kỳ, $Ax = A(x + c)$.

1.2.7 Nghịch đảo

Ma trận *khả nghịch* hoặc *không suy biến* là ma trận vuông hạng đầy đủ. Vì m cột của một ma trận không suy biến $m \times m$ tạo thành một cơ sở cho toàn bộ không gian $\mathbb{C}^m$ nên một vector bất kỳ được biểu diễn duy nhất dưới dạng là một tổ hợp tuyến tính của chúng. Đặc biệt, vector đơn vị chính tắc e_j với 1 ở vị trí thứ j và 0 ở những vị trí còn lại

$$e_j = \sum_{i=1}^{m} z_{ij} a_i. \tag{1.2.8}$$

Cho Z là ma trận với các phần tử là z_{ij}, và cho z_j là cột thứ j của Z. Khi đó, (1.2.8) có thể được viết $e_j = Az_j$. Phương trình này có dạng của (1.2.6) và được viết lại như sau

$$[e_1|\dots|e_m] = I = AZ,$$

với I là ma trận *đơn vị* $m \times m$. Ma trận Z là *ma trận nghịch đảo* của A. Ma trận không suy biến vuông A bất kỳ có duy nhất một nghịch đảo, được ký hiệu bởi A^{-1}, thỏa $AA^{-1} = A^{-1}A = I$.

Định lý sau cho số điều kiện tương đương khi ma trận vuông A không suy biến.

Định lý 1.2.3 *Cho $A \in \mathbb{C}^{m\times m}$, các điều kiện sau là tương đương:*

(a) A có nghịch đảo A^{-1},

(b) $rank(A) = m$,

(c) $range(A) = \mathbb{C}^m$,

(d) $null(A) = \{0\}$,

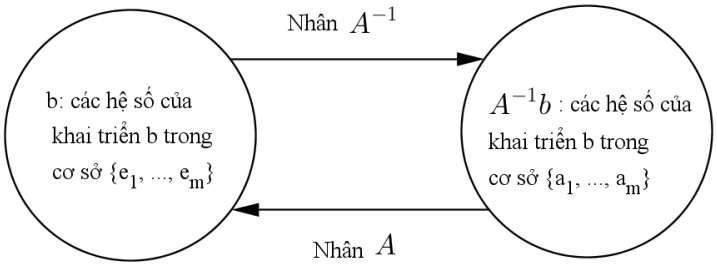
(e) 0 không là trị riêng của A,

(f) 0 không là giá trị suy biến của A,

(g) $det(A) \neq 0$.

1.2.8 Nhân ma trận nghịch đảo với vector

Theo (1.2.6), tích $x = A^{-1}b$ là vector khai triển tuyến tính duy nhất các hệ số của b trong cơ sở các cột của A. Nhân với A^{-1} là một phép toán *chuyển cơ sở*



1.3 Vector và ma trận trực giao

1.3.1 Phụ hợp

Liên hợp phức của một vô hướng z, được ký hiệu bởi $\overline{z}$ hoặc z^*, có được bằng việc phủ định phần ảo của nó. Cho z là số thực, $\overline{z} = z$.

Liên hợp Hermit hay *phụ hợp* của một ma trận A có kích thước $m \times n$, được ký hiệu bởi A^*, là ma trận $n \times m$ mà phần tử i, j của nó là liên hợp phức của phần tử j, i của A. Ví dụ,

$$A = \begin{bmatrix} a_{11} & a_{12} \\ a_{21} & a_{22} \\ a_{31} & a_{32} \end{bmatrix} \Longrightarrow A^* = \begin{bmatrix} \overline{a_{11}} & \overline{a_{21}} & \overline{a_{31}} \\ \overline{a_{12}} & \overline{a_{22}} & \overline{a_{32}} \end{bmatrix}$$

Nếu $A = A^*$ thì A là ma trận *hermitian*. Theo định nghĩa, một ma trận hermit phải là ma trận vuông. Cho A là ma trận thực thì ma trận phụ hợp chỉ đơn giản là hoán đổi các dòng và các cột của A. Trong trường hợp này, ma trận phụ hợp cũng là *ma trận chuyển vị* A^T. Do đó, nếu một ma trận thực là hermit, nghĩa là $A = A^T$, thì nó cũng là *ma trận đối xứng*.

1.3.2 Tích trong

Tích trong của hai vector cột $x, y \in \mathbb{C}^m$ là tích phụ hợp của x với y:

$$x^* y = \sum_{i=1}^{m} \overline{x_i} y_i. \tag{1.3.1}$$

Độ dài Euclidean của x được viết $\|x\|$ (chuẩn vector) và được xác định như căn bậc hai của tích trong của x với chính nó:

$$\|x\| = \sqrt{x^* x} = (\sum_{i=1}^{m} |x_i|^2)^{1/2}. \tag{1.3.2}$$

Cos của góc α giữa x và y được biểu diễn trong các số hạng của tích trong:

$$\cos \alpha = \frac{x^* y}{\|x\| \|y\|}. \tag{1.3.3}$$

Tích trong là *song tuyến tính*, nghĩa là nó tuyến tính theo từng vector riêng biệt

$$\begin{aligned} (x_1 + x_2)^* y &= x_1^* y + x_2^* y, \\ x^*(y_1 + y_2) &= x^* y_1 + x^* y_2, \\ (\alpha x)^*(\beta y) &= \overline{\alpha} \beta x^* y. \end{aligned}$$

Ta cũng sẽ thường xuyên sử dụng tính chất này cho các ma trận hay các vector bất kỳ A và B có các chiều tương thích,

$$(AB)^* = B^* A^*. \tag{1.3.4}$$

Tương tự cho tích của các ma trận vuông khả nghịch,

$$(AB)^{-1} = B^{-1} A^{-1}. \tag{1.3.5}$$

Ký hiệu A^{-*} là một tốc ký của $(A^*)^{-1}$ hay $(A^{-1})^*$; hai ký hiệu này là tương đương, được kiểm tra bằng việc áp dụng (1.3.4) với $B = A^{-1}$.

1.3.3 Các vector trực giao

Hai vector x và y được gọi là *trực giao* nếu $x^*y = 0$. Nếu x và y là các vector thực thì điều này có nghĩa là chúng nằm ở các góc phải trong $\mathbb{R}^m$. Tập hợp các vector X và tập hợp các vector Y là trực giao (hay X trực giao Y) nếu với mọi $x \in X$ là trực giao với mọi $y \in Y$.

Tập hợp các vector S khác không là *trực giao* nếu các phần tử của nó là trực giao từng đôi một, nghĩa là, nếu $x, y \in S, x \neq y \Longrightarrow x^*y = 0$. Tập hợp các vector là *trực chuẩn* nếu nó trực giao và với mọi $x \in S$, $\|x\| = 1$.

Định lý 1.3.1 *Các vector trong một tập hợp trực giao S là độc lập tuyến tính.*

Chứng minh Nếu các vector trong S không độc lập tuyến tính thì tồn tại $v_k \in S$ bất kỳ được biểu diễn dưới dạng tổ hợp tuyến tính của $v_1, \ldots, v_n \in S$,

$$v_k = \sum_{\substack{i=1 \\ i \neq k}}^{n} c_i v_i.$$

Vì $v_k \neq 0, v_k^* v_k = \|v_k\|^2 > 0$. Sử dụng tính song tuyến tính của tích trong và tính trực giao của S, ta tính

$$v_k^* v_k = \sum_{\substack{i=1 \\ i \neq k}}^{n} c_i v_k^* v_i = 0,$$

mâu thuẫn với giả thuyết các vector trong S là khác không.

Như là một hệ quả của Định lý 1.3.1, nếu một tập trực giao $S \subseteq \mathbb{C}^m$ chứa m vector thì nó là một cơ sở của $\mathbb{C}^m$.

1.3.4 Các thành phần của một vector

Ý tưởng quan trọng nhất từ các khái niệm của tích trong và trực giao là các tích trong có thể được sử dụng để phân tích các vector tùy ý thành các thành phần trực giao.

Ví dụ, giả sử $\{q_1, q_2, \ldots, q_n\}$ là một tập trực giao, v là một vector bất kỳ. Con số $q_j^* v$ là một vô hướng. Khi đó, ta có vector

$$r = v - (q_1^* v)q_1 - (q_2^* v)q_2 - \ldots - (q_n^* v)q_n \tag{1.3.6}$$

là trực giao với $\{q_1, q_2, \ldots, q_n\}$. Điều này có thể được kiểm tra bằng việc tính $q_i^* v$

$$q_i^* v = q_i^* v - (q_1^* v)(q_i^* q_1) - \ldots - (q_n^* v)(q_i^* q_n).$$

Rút gọn tổng này, vì $q_i^* q_j = 0$ với $i \neq j$:

$$q_i^* v = q_i^* v - (q_i^* v)(q_i^* q_i) = 0.$$

Do đó ta thấy v được phân tích thành $n + 1$ thành phần trực giao:

$$v = r + \sum_{i=1}^{n} (q_i^* v)q_i = r + \sum_{i=1}^{n} (q_i q_i^*)v. \tag{1.3.7}$$

Trong phân tích này, r là phần của v trực giao với tập các vector $\{q_1, q_2, \ldots, q_n\}$, hay không gian con sinh bởi tập các vector này và $(q_i^* v)q_i$ là phần của v trong phương của q_i.

Nếu $\{q_i\}$ là cơ sở của $\mathbb{C}^m$ thì n phải bằng m và r phải là vector không, nên v được phân tích đầy đủ thành m thành phần trực giao trong các phương của q_i:

$$v = \sum_{i=1}^{m} (q_i^* v) q_i = \sum_{i=1}^{m} (q_i q_i^*) v. \tag{1.3.8}$$

Hai công thức (1.3.7) và (1.3.8), một với $(q_i^* v)q_i$ và một với $(q_i q_i^*)v$, là tương đương nhau nhưng chúng có các giải thích khác nhau. Trong trường hợp đầu tiên, ta xem v như một tổng của các hệ số $q_i^* v$ nhân với các vector q_i. Trong trường hợp hai, ta xem v như một tổng của các phép chiếu trực giao của v vào các phương khác nhau q_i. Phép chiếu thứ i được thực hiện bởi ma trận hạng một rất đặc biệt $q_i q_i^*$.

1.3.5 Các ma trận Unita

Một ma trận vuông $Q \in \mathbb{C}^{m\times m}$ là *unita* (trong trường hợp thực, *trực giao*) nếu $Q^* = Q^{-1}$, nghĩa là, nếu $Q^*Q = I$. Trong các dạng cột của Q, tích này được viết như sau

$$\begin{bmatrix} \overline{q_1^*} \\ \overline{q_2^*} \\ \overline{\vdots} \\ \overline{q_m^*} \end{bmatrix} \left[\begin{array}{c|c|c|c} q_1 & q_2 & \cdots & q_m \end{array} \right] = \begin{bmatrix} 1 & & & \\ & 1 & & \\ & & \ddots & \\ & & & 1 \end{bmatrix}.$$

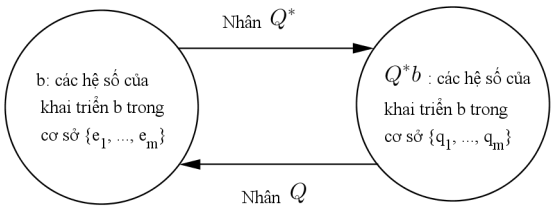
Mặt khác, $q_i^* q_j = \delta_{ij}$, và các cột của một ma trận unita Q tạo thành một cơ sở trực giao của $\mathbb{C}^m$. Ký hiệu δ_{ij} là *Kronecker delta*, bằng 1 nếu $i = j$ và bằng 0 nếu $i \neq j$.

1.3.6 Nhân với ma trận unita

Nếu Λ là một ma trận unita Q thì Λx và $\Lambda^{-1} b$ trở thành Qx và $Q^* b$. Như trong những mục trước, Qx là tổ hợp tuyến tính của các cột của Q với các hệ số x. Ngược lại,

$Q^* x$ là vector của các hệ số của khai triển b trong cơ sở các cột của Q.

Dưới dạng biểu đồ, nó trông giống như hình 1.3.6



Các quá trình của phép nhân với một ma trận unita hoặc phụ hợp của nó bảo toàn cấu trúc hình học trong ý nghĩa Euclidean bởi vì các tích trong được bảo toàn. Đó là, với ma trận unita Q,

$$(Qx)^*(Qy) = x^* y, \tag{1.3.9}$$

được kiểm tra bởi (1.3.4). Tính bất biến của tích trong nghĩa là các góc giữa các vector được bảo toàn, và chiều dài của chúng là

$$\|Qx\| = \|x\|. \tag{1.3.10}$$

Trong trường hợp thực, phép nhân với ma trận trực giao Q tương ứng với phép quay cố định (nếu $detQ = 1$) hoặc phép đối xứng (nếu $detQ = -1$) của không gian vector.

1.4 Trực chuẩn

1.4.1 Các chuẩn vector

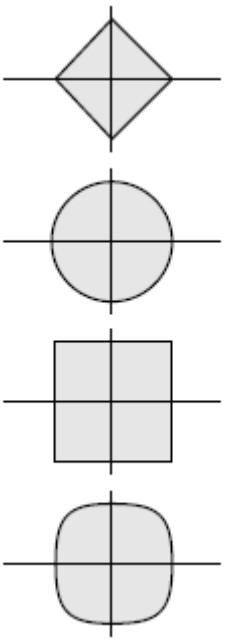
Một *chuẩn* là một hàm $\|\cdot\| : \mathbb{C}^m \to \mathbb{R}$ thỏa mãn 3 điều kiện theo sau: Với mọi vector x và y và với mọi vô hướng $\alpha \in \mathbb{C}$,

$$\begin{aligned} &(1) \quad \|x\| \geqslant 0, \text{ và} \|x\| = 0 \text{ nếu } x = 0, \\ &(2) \quad \|x + y\| \leqslant \|x\| + \|y\|, \\ &(3) \quad \|\alpha x\| = |\alpha| \|x\|. \end{aligned} \tag{1.4.1}$$

Trong đó, (1) chuẩn của một vector khác không là dương, (2) chuẩn của một tổng vector là không vượt quá tổng của các chuẩn của các phần của nó - *bất đẳng thức tam giác*, và (3) chia tỉ lệ chuẩn một vector vô hướng của nó bằng số lượng giống nhau.

Quả cầu đơn vị đóng $\{x \in \mathbb{C}^m : \|x\| \leqslant 1\}$ tương ứng với mỗi chuẩn được minh họa ở hình bên dưới cho trường hợp $m = 2$.

$$\begin{aligned}
\|x\|_1 &= \sum_{i=1}^{m} |x_i|, \\
\|x\|_2 &= \left(\sum_{i=1}^{m} |x_i|^2 \right)^{1/2} = \sqrt{x^* x}, \\
\|x\|_\infty &= \max_{1 \leqslant i \leqslant m} |x_i|, \\
\|x\|_p &= \left(\sum_{i=1}^{m} |x_i|^p \right)^{1/p} \qquad (1 \leqslant p < \infty).
\end{aligned} \tag{1.4.2}$$



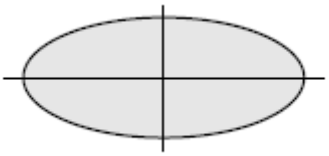
Tổng quát, cho chuẩn $\|\cdot\|$ bất kỳ, một chuẩn có trọng số có thể được viết như

$$\|x\|_W = \|Wx\|. \tag{1.4.3}$$

W ở đây là ma trận đường chéo mà phần tử đường chéo ở vị trí thứ i là trọng số $w_i \neq 0$. Ví dụ, một chuẩn 2 có trọng số $\|\cdot\|_W$ trong $\mathbb{C}^m$ được thiết lập như sau:

$$\|x\|_W = \left(\sum_{i=1}^{m} |w_i x_i|^2 \right)^{1/2}. \tag{1.4.4}$$

Ta cũng có thể tổng quát hóa ý tưởng các chuẩn có trọng số bằng việc cho phép W là một ma trận không suy biến tùy ý, không cần thiết là đường chéo.



1.4.2 Các chuẩn ma trận bao gồm bởi các chuẩn vector

Ma trận $m \times n$ có thể được xem như một vector trong không gian mn chiều mà mỗi phần tử mn của ma trận là một tọa độ độc lập. Một chuẩn mn chiều bất kỳ có thể được sử dụng cho việc đo "kích thước" của một ma trận như vậy.

Cho các chuẩn vector $\|\cdot\|_{(n)}$ và $\|\cdot\|_{(m)}$ trong miền xác định và vùng của $A \in \mathbb{C}^{m\times n}$ tương ứng, chuẩn ma trận $\|A\|_{(m,n)}$ là số nhỏ nhất C sao cho bất đẳng thức sau đúng với mọi $x \in \mathbb{C}^n$:

$$\|Ax\|_{(m)} \leqslant C\|x\|_{(n)}. \tag{1.4.5}$$

Mặt khác, $\|A\|_{(m,n)}$ là cận trên đúng của tỉ số $\|Ax\|_{(m)}/\|x\|_{(n)}$ với mọi vector $x \in \mathbb{C}^n$ - thừa số lớn nhất mà A có thể "giãn" một vector x. Ta nói rằng $\|\cdot\|_{(m,n)}$ là chuẩn ma trận được bao gồm bởi $\|\cdot\|_{(n)}$ và $\|\cdot\|_{(m)}$.

Bởi vì điều kiện (3) của (1.4.1), tác động của A được xác định bởi tác động của nó trong các vector đơn vị. Do đó, chuẩn ma trận có thể được xác định một cách tương đương với các ảnh của các vector đơn vị dưới A:

$$\|Ax\|_{(m,n)} = \sup_{\substack{x\in\mathbb{C}^n \\ x\neq 0}} \frac{||\mathrm{Ax}||_{(m)}}{||x||_{(n)}} = \sup_{\substack{x\in\mathbb{C}^n \\ ||x||_{(n)}=1}} ||\mathrm{Ax}||_{(m)}. \tag{1.4.6}$$

1.4.3 Các ví dụ

Ví dụ 1.4.1. Ma trận

$$A = \begin{bmatrix} 1 & 2 \\ 0 & 2 \end{bmatrix} \tag{1.4.7}$$

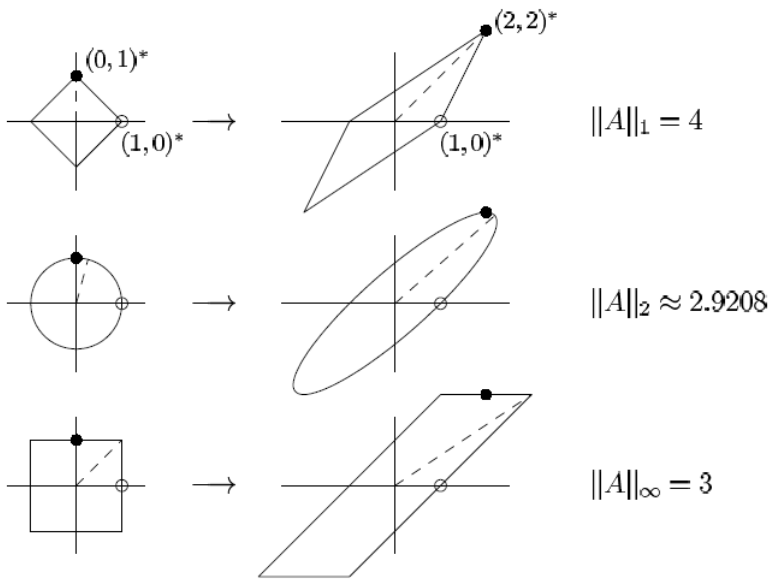
ánh xạ từ $\mathbb{C}^2$ vào $\mathbb{C}^2$. Nó cũng ánh xạ từ $\mathbb{R}^2$ vào $\mathbb{R}^2$.

Hình 1.1 miêu tả tác động của A vào các quả cầu đơn vị của $\mathbb{R}^2$ xác định bởi chuẩn 1, chuẩn 2 và chuẩn ∞. Không quan tâm tới chuẩn, A ánh xạ $e_1 = (1, 0)^*$ thành cột đầu tiên của A, cụ thể chính là e_1, và $e_2 = (0, 1)^*$ thành cột thứ 2 của A, cụ thể là $(2, 2)^*$. Trong chuẩn 1, vector đơn vị x được khuếch đại hầu hết cho A là $(0, 1)^*$ (hoặc phủ định của nó), và thừa số khuếch đại là 4. Trong chuẩn ∞, vector đơn vị x được khuếch đại hầu hết cho A là $(1, 1)^*$ (hoặc phủ định của nó), và thừa số khuếch đại là 3. Trong chuẩn 2, vector đơn vị được khuếch đại hầu hết cho A là vector được bao gồm bởi đường đứt nét trong hình (hoặc phủ định của nó), và nhân tố khuếch đại là xấp xỉ 2.9208. (Chú ý rằng nó phải ít nhất là $\sqrt{8} \approx 2.8284$, vì $(0, 1)^*$ ánh xạ thành $(2, 2)^*$.)

Ví dụ 1.4.2. Chuẩn p của ma trận đường chéo. Cho D là một ma trận đường chéo

$$D = \begin{bmatrix} d_1 & & & \\ & d_2 & & \\ & & \ddots & \\ & & & d_m \end{bmatrix}$$

Khi đó, trong dòng thứ hai của Hình 1.1, ảnh của hình cầu đơn vị chuẩn 2 của D là một ellip m chiều mà các chiều dài bán trục của nó được cho bởi các số $|d_i|$. Các vector đơn vị khuếch



Hình 1.1: Các quả cầu đơn vị ứng với các chuẩn 1, 2 và ∞

đại lớn nhất cho D mà nó được ánh xa tới bán trục dài nhất của ellip, của chiều dài $max_i\{|d_i|\}$. Do đó, ta có $\|D\|_2 = max_{1\leqslant i\leqslant m}\{|d_i|\}$. Kết quả này cho chuẩn 2 tổng quát hóa cho chuẩn p bất kỳ: nếu D là đường chéo thì $\|D\|_p = max_{1\leqslant i\leqslant m}|d_i|$.

Ví dụ 1.4.3. Chuẩn 1 của một ma trận. Nếu A là một ma trận $m \times n$ bất kỳ thì $\|A\|_1$ là bằng với "tổng cột lớn nhất" của A. Ta giải thích kết quả này như sau. Viết A dưới dạng các cột của nó

$$A = [a_1|\dots|a_n], \tag{1.4.8}$$

với mỗi a_j là vector m chiều. Xét quả cầu đơn vị chuẩn 1 có hình dạng giống kim cương trong $\mathbb{C}^n$, được minh hoa như trong 1.4.2. Đó là tập hợp $\{x \in \mathbb{C}^n : \sum_{j=1}^n |x_j| \leqslant 1\}$. Một vector Ax bất kỳ trong hình này thỏa mãn

$$\|Ax\|_1 = \|\sum_{j=1}^n x_j a_j\|_1 \leqslant \sum_{j=1}^n |x_j|\|a_j\|_1 \leqslant max_{1\leqslant j\leqslant n}\|a_j\|_1.$$

Do đó, chuẩn 1 ma trận thỏa mãn $\|A\|_1 \leqslant max_{1\leqslant j\leqslant n}\|a_j\|_1$. Do chọn $x = e_j$ với j cực đại hóa $\|a_j\|_1$, do đó chuẩn ma trận là

$$\|A\|_1 = max_{1\leqslant j\leqslant n}\|a_j\|_1. \tag{1.4.9}$$

Ví dụ 1.4.4. Chuẩn ∞ của một ma trận. Do nhiều đối số giống nhau nên chuẩn ∞ của một ma trận $m \times n$ là tương đương với "tổng dòng lớn nhất",

$$\|A\|_\infty = max_{1\leqslant j\leqslant n}\|a_j^*\|_1, \tag{1.4.10}$$

với a_j^* là dòng thứ i của A.

1.4.4 Bất đẳng thức Cauchy - Schwarz và Holder

Việc tính toán chuẩn p của ma trận với $p \neq 1, \infty$ là khó khăn hơn, và để xấp xỉ bài toán này, ta chú ý rằng các tích trong có thể được chặn bằng việc sử dụng chuẩn p. Cho p và q thỏa mãn $\frac{1}{p} + \frac{1}{q} = 1$, với $1 \leqslant p, q \leqslant \infty$. Khi đó *bất đẳng thức Holder* phát biểu rằng, với các vector x và y bất kỳ,

$$|x^*y| \leqslant \|x\|_p\|y\|_q. \tag{1.4.11}$$

Bất đẳng thức Cauchy - Schwarz là trường hợp đặc biệt $p = q = 2$:

$$|x^*y| \leqslant \|x\|_2\|y\|_2. \tag{1.4.12}$$

Các kết quả này có thể được tìm thấy trong các sách Đại số tuyến tính.

Ví dụ 1.4.5. Chuẩn 2 của một vector dòng. Xét một ma trận A chứa một dòng đơn. Ma trận này có thể được viết như $A = a^*$, với a là một vector cột. Theo bất đẳng thức Cauchy - Schwarz, với x bất kỳ, ta có $\|Ax\|_2 = |a^*x| \leqslant \|a\|_2\|x\|_2$ mà $\|Aa\|_2 = \|a\|_2^2$. Do đó, ta có

$$\|A\|_2 = \sup_{x\neq 0}\{\|Ax\|_2/\|x\|_2\} = \|a\|_2.$$

Ví dụ 1.4.6. Chuẩn 2 của tích ngoài. Tổng quát, xét tích ngoài hạng 1 $A = uv^*$, với u là một vector m chiều và v là một vector n chiều. Cho vector n chiều x bất kỳ, ta có thể chặn $\|Ax\|_2$ như sau

$$\|Ax\|_2 = \|uv^*x\|_2 = \|u\|_2|v^*x| \leqslant \|u\|_2\|v\|_2\|x\|_2. \tag{1.4.13}$$

Do đó $\|A\|_2 \leqslant \|u\|_2\|v\|_2$. Dấu "=" xảy ra khi $x = v$.

1.4.5 Việc chặn $\|AB\|$ trong chuẩn ma trận được bao gồm

Cho $\|\cdot\|_{(l)}, \|\cdot\|_{(m)}$ và $\|\cdot\|_{(n)}$ là các chuẩn tương ứng trong $\mathbb{C}^l, \mathbb{C}^m$, và $\mathbb{C}^n$, và cho A là một ma trận $l \times m$ và B là ma trận $m \times n$. Với $x \in \mathbb{C}^n$ bất kỳ, ta có

$$\|ABx\|_{(l)} \leqslant \|A\|_{(l,m)}\|Bx\|_{(m)} \leqslant \|A\|_{(l,m)}\|B\|_{(m,n)}\|x\|_{(n)}.$$

Do đó chuẩn được bao gồm của AB phải thỏa mãn

$$\|AB\|_{(l,n)} \leqslant \|A\|_{(l,m)}\|B\|_{(m,n)}. \tag{1.4.14}$$

Tổng quát, bất đẳng thức này không là đẳng thức. Ví dụ, bất đẳng thức $\|A^n\| \leqslant \|A\|^n$ đúng với ma trận vuông bất kỳ trong chuẩn ma trận bất kỳ được bao gồm bởi một vector chuẩn, nhưng $\|A^n\| = \|A\|^n$ không đúng trong trường hợp tổng quát với $n \geqslant 2$.

1.4.6 Các chuẩn ma trận tổng quát

Tổng quát, một chuẩn ma trận phải thỏa mãn 3 điều kiện chuẩn vector 1.4.1 áp dụng trong không gian vector mn chiều của các ma trận:

$$\begin{aligned}
&(1) \quad \|A\| \geqslant 0, \text{ và} \|A\| = 0 \text{chỉ nếu } A = 0,\\
&(2) \quad \|A + B\| \leqslant \|A\| + \|B\|,\\
&(3) \quad \|\alpha A\| = |\alpha|\|A\|.
\end{aligned} \tag{1.4.15}$$

Chuẩn ma trận phổ biến nhất không được bao gồm bởi một chuẩn vector là *chuẩn Hilbert - Schmidt* hoặc *chuẩn Frobenius*, xác định bởi

$$\|A\|_F = \left(\sum_{i=1}^{m}\sum_{j=1}^{n}|a_{ij}|^2\right)^{1/2}. \tag{1.4.16}$$

Chuẩn này giống như chuẩn 2 của ma trận khi nó được xem như là một vector mn chiều. Công thức của chuẩn Frobenius cũng có thể được viết dưới dạng các cột hoặc các dòng riêng biệt. Ví dụ, nếu a_j là cột thứ j của A, ta có

$$\|A\|_F = \left(\sum_{j=1}^{n}\|a_j\|_2^2\right)^{1/2}. \tag{1.4.17}$$

Tương tự cho các dòng, ta có

$$\|A\|_F = \sqrt{tr(A^*A)} = \sqrt{tr(AA^*)}, \tag{1.4.18}$$

với $tr(B)$ là *vết* của B, tổng các phần tử trên đường chéo của nó.

Như chuẩn ma trận được bao gồm, chuẩn Frobenius có thể được sử dụng để chặn các tích của các ma trận. Cho $C = AB$ với các phần tử c_{ik}, và cho a_i^* là dòng thứ i của A và b_j là cột thứ j của B. Khi đó $c_{ij} = a_i^* b_j$, nên theo bất đẳng thức Cauchy - Schwarz ta có $|c_{ij}| \leqslant \|a_i\|_2 \|b_j\|_2$. Bình phương cả hai vế và tính tổng trên tất cả chỉ số i, j, ta được

$$\begin{aligned}
\|AB\|_F^2 &= \sum_{i=1}^{n}\sum_{j=1}^{m} |c_{ij}|^2 \\
&\leqslant \sum_{i=1}^{n}\sum_{j=1}^{m} (\|a_i\|_2 \|b_j\|_2)^2 \\
&= \sum_{i=1}^{n} (\|a_i\|_2)^2 \sum_{j=1}^{m} (\|b_j\|_2)^2 = \|A\|_F^2 \|B\|_F^2.
\end{aligned}$$

1.4.7 Bất biến dưới phép nhân Unita

Một trong số những tính chất đặc biệt của chuẩn 2 ma trận là, như chuẩn 2 vector, tính bất biến dưới phép nhân các ma trận Unita. Tính chất này cũng đúng cho chuẩn Frobenius.

Định lý 1.4.1 *Cho $A \in \mathbb{C}^{m\times n}$ bất kỳ và ma trận Unita $Q \in \mathbb{C}^{m\times m}$, ta có*

$$\|QA\|_2 = \|A\|_2, \ \|QA\|_F = \|A\|_F.$$

Chứng minh Vì $\|Qx\|_2 = \|x\|_2$ với mọi x, theo (1.3.10), tính bất biến trong chuẩn 2 theo sau từ (1.4.6). Cho chuẩn Frobenius ta có thể sử dụng (1.4.18).

Theo Định lý 1.4.1, nếu Q được tổng quát hóa thành ma trận hình chữ nhật với các cột trực giao, nghĩa là, $Q \in \mathbb{C}^{p\times m}$ với $p > m$. Tương tự tính đồng nhất cũng đúng cho phép nhân các ma trận Unita trong vế phải, hoặc tổng quát hơn, nhân các ma trận hình chữ nhật với các dòng trực giao.

1.5 Phân tích giá trị suy biến

Phân tích giá trị suy biến (Singular Value Decomposition - SVD) là phân tích ma trận mà tính toán của nó là một bước trong nhiều thuật toán. Nhiều bài toán của Đại số tuyến tính có thể được hiểu tốt hơn nếu đầu tiên ta đặt câu hỏi: cái gì xảy ra nếu ta dùng SVD?

1.5.1 Quan sát hình học

Phân tích các giá trị suy biến (SVD) được thúc đẩy bởi lập luận hình học sau:

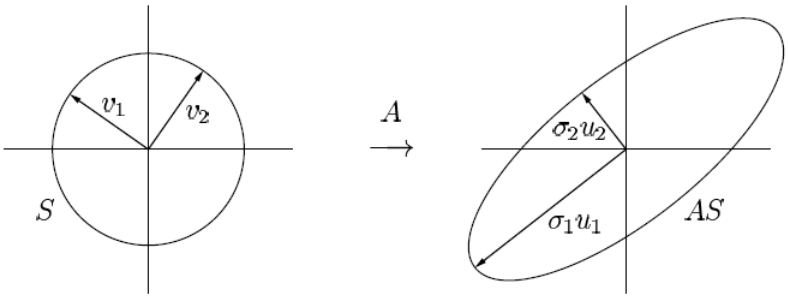
Ảnh của một quả cầu đơn vị dưới ma trận $m \times n$ bất kỳ là một siêu ellip.

SVD có thể áp dụng được cho cả ma trận thực và ma trận phức. Tuy nhiên, trong việc mô tả hình học, ma trận sử dụng là ma trận thực.

Thuật ngữ "siêu ellip" có thể là xa lạ, nhưng đó là sự tổng quát hóa m chiều của một ellip. Ta có thể định nghĩa một siêu ellip trong $\mathbb{R}^m$ như là một mặt thu được bằng việc kéo căng

quả cầu đơn vị trong $\mathbb{R}^m$ bởi các thừa số $\sigma_1, \ldots, \sigma_m$ (có thể là 0) trong các phương trực giao bất kỳ $u_1, \ldots, u_m \in \mathbb{R}^m$. Cho thuật lợi, ta lấy u_i là các vector đơn vị, nghĩa là, $\|u_i\|_2 = 1$. Các vector $\{\sigma_i u_i\}$ là *các bán trục chính* của siêu ellip, với các độ dài $\sigma_1, \ldots, \sigma_m$. Nếu A có hạng r, một cách chính xác r của các độ dài σ_i sẽ trả ra giá trị khác 0, và đặc biệt, nếu $m \geqslant n$, tối đa n trong số chúng sẽ là khác 0.

Ảnh của quả cầu đơn vị mà ta muốn nói là quả cầu Euclidean thông thường trong không gian n chiều, nghĩa là, quả cầu đơn vị trong chuẩn 2; ta ký hiệu nó là S. Khi đó AS, ảnh của S dưới ánh xạ A, là một siêu ellip như vừa được xác định.



Hình 1.2: SVD của ma trận 2×2

Cho S là quả cầu đơn vị trong $\mathbb{R}^n$, và lấy $A \in \mathbb{R}^{m \times n}$ bất kỳ với $m \geqslant n$. Để đơn giản, giả sử A có hạng đầy đủ là n. Ảnh AS là một siêu ellip trong $\mathbb{R}^m$. Bây giờ ta định nghĩa một vài tính chất của A liên quan tới hình dạng của AS. Ý tưởng được miêu tả trong Hình 1.2.

Đầu tiên, ta định nghĩa *n giá trị suy biến* của A. Các giá trị này là các độ dài của n bán trục chính của AS, được viết $\sigma_1, \sigma_2, \ldots, \sigma_n$. Theo qui ước giả sử rằng các giá trị suy biến được đánh số trong thứ tự giảm dần, $\sigma_1 \geqslant \sigma_2 \geqslant \ldots \geqslant \sigma_n > 0$.

Tiếp theo, ta định nghĩa *n vector suy biến trái* của A. Các vector này là các vector đơn vị $\{u_1, u_2, \ldots, u_n\}$ trực giao với phương của các bán trục chính của AS, tương ứng với các giá trị suy biến. Do đó vector $\sigma_i u_i$ là bán trục chính lớn nhất thứ i của AS.

Cuối cùng, ta định nghĩa *n vector suy biến phải* của A. Các vector này là các vector đơn vị của $\{v_1, v_2, \ldots, v_n\} \in S$ mà chúng là ảnh ngược của các bán trục chính của AS, được ký hiệu bởi $Av_j = \sigma_j u_j$.

Các thuật ngữ "trái" và "phải" trong các định nghĩa trên là rõ ràng bất tiện. Chúng đến từ các vị trí cả của các thừa số U và V trong (1.5.2) và (1.5.3) bên dưới. Bất tiện ở đây là kéo dãn giống như Hình 1.2, các vector suy biến trái tương ứng với không gian bên phải, và các vector suy biến phải tương ứng với không gian bên trái! Ta có thể giải bài toán này bằng việc hoán đổi hai nửa của hình, với ánh xạ A chỉ hướng từ phải sang trái, nhưng điều này sẽ đi ngược lại các thói quen.

1.5.2 SVD được giảm

Như đề cập ở trên,ta có phương trình liên hệ các vector suy biến phải $\{v_j\}$ với các vector suy biến trái $\{u_j\}$

$$Av_j = \sigma_j u_j, \ 1 \leqslant j \leqslant n. \tag{1.5.1}$$

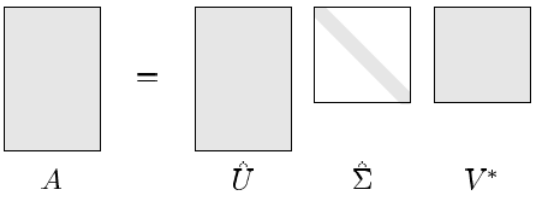
Tập hợp các phương trình vector này có thể được biểu diễn như một phương trình ma trận,hay $AV = \widehat{U}\widehat{\sum}$. Trong phương trình ma trận này, $\widehat{\sum}$ là ma trận đường chéo $n \times n$ với các phần tử thực dương (vì A được giả sử có hạng đầy đủ n), $\widehat{U}$ là ma trận $m \times n$ với các cột trực

$$\begin{bmatrix} & & \\ & A & \\ & & \end{bmatrix} \begin{bmatrix} | & | & & | \\ v_1 & v_2 & \cdots & v_n \\ | & | & & | \end{bmatrix} = \begin{bmatrix} | & | & & | \\ u_1 & u_2 & \cdots & u_n \\ | & | & & | \end{bmatrix} \begin{bmatrix} \sigma_1 & & & \\ & \sigma_2 & & \\ & & \ddots & \\ & & & \sigma_n \end{bmatrix},$$

giao, và V là ma trận $n \times n$ với các cột trực giao. Do đó V là ma trận Unita, và ta có thể nhân V^* bên phải nó để được

$$A = \widehat{U}\widehat{\sum}V^*. \tag{1.5.2}$$

Phân tích này của A được gọi là *phân tích giá trị suy biến được giảm* hay *SVD được giảm* của A. Dưới dạng biểu đồ, nó trông giống điều này



Hình 1.3: SVD bị giảm ($m \geqslant n$)

1.5.3 SVD đầy đủ

Các cột của $\widehat{U}$ là n vetor trực giao trong không gian m chiều $\mathbb{C}^m$. Trừ khi $m = n$, chúng không hình thành một cơ sở của $\mathbb{C}^m$, $\widehat{U}$ cũng không là ma trận Unita. Tuy nhiên, bằng việc nối thêm $m - n$ cột trực giao, $\widehat{U}$ có thể được mở rộng thành một ma trận Unita.

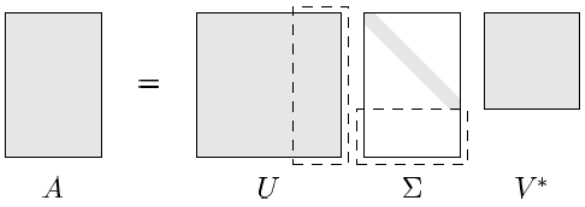
Nếu $\widehat{U}$ được thay thế bởi U trong (1.5.2) thì $\widehat{\sum}$ sẽ phải thay đổi như vậy. Cho tích giữ nguyên không thay đổi, các cột $m - n$ cuối của U sẽ được nhân với 0. Do đó, cho $\sum$ là ma trận $m \times n$ gồm có $\widehat{\sum}$ trong $n \times n$ khối bên trên với $m - n$ dòng 0 bên dưới. Phân tích giá trị suy biến đầy đủ hay SVD đầy đủ của A

$$A = U \sum V^*. \tag{1.5.3}$$

với U là ma trân Unita $m \times m$, V là ma trận Unita $n \times n$, và $\sum$ là ma trận đường chéo $m \times n$ với các phần tử thực dương. Dưới dạng biểu đồ

Nếu A có hạng không đầy đủ thì phân tích (1.5.3) vẫn là thích hợp. Tất cả các thay đổi đó là không phải là n mà là r vector suy biến trái của A được xác định bởi hình học của siêu ellip. Để xây dựng ma trận Unita U, ta đưa vào $m - r$ thay cho $m - n$ cột trực giao tùy ý. Ma trận V cũng sẽ cần $n - r$ cột trực giao tùy ý để mở rộng r cột xác định bởi hình học. Ma trận $\sum$ sẽ có r phần tử đường chéo dương, với $n - r$ còn lại bằng 0.

Vì vậy, SVD bị giảm (1.5.2) cũng làm số chiều các ma trận A nhỏ hơn hạng đầy đủ. Ta có thể lấy ma trận $\widehat{U}$ là $m \times n$, với các số chiều của $\sum$ là $n \times n$ với một vài số 0 trên đường chéo, hoặc nén $\widehat{U}$ thành $m \times r$ và $\widehat{\sum}$ thành $r \times r$ và dương ngặt trên đường chéo.



Hình 1.4: SVD đầy đủ ($m \geqslant n$)

1.5.4 Định nghĩa

Cho m và n tùy ý và cho $A \in \mathbb{C}^{m\times n}$, *phân tích giá trị suy biến* (SVD) của A là một phân tích

$$A = U \sum V^* \tag{1.5.4}$$

với

$$\begin{aligned} &U \in \mathbb{C}^{m\times m} \text{ là Unita,} \\ &V \in \mathbb{C}^{n\times n} \text{ là Unita,} \\ &\textstyle\sum \in \mathbb{C}^{m\times n} \text{ là đường chéo.} \end{aligned}$$

Hơn nữa, giả sử các phần tử trên đường chéo σ_j của $\sum$ là không âm và sắp thứ tự không tăng, nghĩa là, $\sigma_1 \geqslant \sigma_2 \geqslant \ldots \geqslant \sigma_p \geqslant 0$, với $p = \min(m, n)$.

Chú ý, ma trận đường chéo $\sum$ có hình dạng giống như A khi A không là ma trận vuông, nhưng U và V thường là các ma trận Unita vuông.

Rõ ràng ảnh của quả cầu đơn vị trong $\mathbb{R}^n$ dưới ánh xạ $A = U \sum V^*$ phải là một siêu ellip trong $\mathbb{R}^m$. Ánh xạ Unita V^* bảo toàn quả cầu, ma trận đường chéo $\sum$ kéo quả cầu thành một siêu ellip được căn lề với cơ sở chính tắc, và ánh xạ Unita cuối cùng U quay hoặc phản xạ ngoài siêu ellip không thay đổi hình dạng của nó. Do đó, nếu ta có thể chứng minh mọi ma trận có một SVD thì ta sẽ chứng minh ảnh của quả cầu đơn vị dưới ánh xạ tuyến tính bất kỳ là một siêu ellip.

1.5.5 Sự tồn tại và tính duy nhất

Định lý 1.5.1 *Mọi ma trận $A \in \mathbb{C}^{m\times n}$ có một phân tích giá trị suy biến (1.5.4). Hơn nữa, các giá trị suy biến $\{\sigma_j\}$ được xác định duy nhất, và, nếu A là ma trận vuông và σ_j là phân biệt, các vector suy biến trái $\{u_j\}$ và phải $\{v_j\}$ được xác định duy nhất thành các ký hiệu phức (nghĩa là, các thừa số vô hướng phức của giá trị tuyệt đối 1).*

Chứng minh Để chứng minh sự tồn tại của SVD, ta tách phương tác động lớn nhất của A, và khi đó tiếp tục phương pháp quy nạp theo số chiều của A.

Đặt $\sigma_1 = \|A\|_2$. Theo đối số tính compact, ở đây phải là các vector $v_1 \in \mathbb{C}^n$ và $u_1 \in \mathbb{C}^m$ với $\|v_1\|_2 = \|u_1\|_2 = 1$ và $Av_1 = \sigma_1 u_1$. Xét các khai triển bất kỳ của v_1 thành một cơ sở trực giao $\{v_j\}$ của $\mathbb{C}^n$ và khai triển của u_1 thành cơ sở trực giao $\{u_j\}$ của $\mathbb{C}^m$, và cho U_1 và V_1 ký hiệu là các ma trận Unita với các cột tương ứng là u_j và v_j. Khi đó ta có

$$U_1^* A V_1 = S = \begin{bmatrix} \sigma_1 & w^* \\ 0 & B \end{bmatrix}, \tag{1.5.5}$$

với 0 là vector cột $m-1$ chiều, w^* là vector dòng $n-1$ chiều, và B có $(m-1)\times(n-1)$ chiều. Hơn nữa,

$$\left\| \begin{bmatrix} \sigma_1 & w^* \\ 0 & B \end{bmatrix} \begin{bmatrix} \sigma_1 \\ w \end{bmatrix} \right\|_2 \geqslant \sigma_1^2 + w^*w = (\sigma_1^2 + w^*w)^{1/2} \left\| \begin{bmatrix} \sigma_1 \\ w \end{bmatrix} \right\|_2,$$

kéo theo $\|S\|_2 \geqslant (\sigma_1^2 + w^*w)^{1/2}$. Vì U_1, V_1 là ma trận Unita và $\|S\|_2 = \|A\|_2 = \sigma_1$ nên điều này suy ra $w = 0$.

Nếu $n = 1$ hoặc $m = 1$ thì ta đã hoàn thành. Mặt khác, ma trận con B là tác động của A vào không gian con trực giao với v_1. Theo giả thiết quy nạp, B có một SVD $B = U_2 \sum_2 V_2^*$. Bây giờ ta dễ dàng kiểm tra

$$A = U_1 \begin{bmatrix} 1 & 0 \\ 0 & U_2 \end{bmatrix} \begin{bmatrix} \sigma_1 & 0 \\ 0 & \sum_2 \end{bmatrix} \begin{bmatrix} 1 & 0 \\ 0 & V_2 \end{bmatrix}^* V^*$$

là một SVD của A, hoàn thành chứng minh sự tồn tại.

Cho tính duy nhất, chứng minh hình học là không phức tạp: nếu độ dài các bán trục của một siêu ellip là phân biệt, khi đó các bán trục của chúng được xác định bởi hình học, lên tới các ký hiệu. Về phương diện đại số, ta có thể chứng minh như sau. Đầu tiên, từ (1.5.4)ta chú ý rằng σ_1 là được xác định duy nhất bởi điều kiện mà nó bằng $\|A\|_2$. Giả sử, có một vector độc lập tuyến khác w với $\|w\|_2 = 1$ và $\|Aw\|_2 = \sigma_1$. Xác định một vector đơn vị v_2 mà nó trực giao với v_1 là một tổ hợp tuyến tính của v_1 và w

$$v_2 = \frac{w - (v_1^*w)v_1}{\|w - (v_1^*w)v_1\|_2}.$$

Vì $\|A\|_2 = \sigma_1, \|Av_2\|_2 \leqslant \sigma_1$ nhưng điều này phải bằng nhau cho trường hợp khác. Vì $w = v_1c + v_2s$ với các hằng số c và s bất kỳ thỏa $|c|^2 + |s|^2 = 1$, ta sẽ có $\|Aw\|_2 \leqslant \sigma_1$. Vector v_2 này là vector suy biến phải thứ hai của A ứng với giá trị suy biến σ_1 nên tồn tại một vector y (tương đương với $n-1$ thành phần cuối của $V_1^*v_2$) thỏa $\|y\|_2 = 1$ và $\|By\|_2 = \sigma_1$. Nếu vector suy biến v_1 là không duy nhất thì giá trị suy biến σ_1 tương ứng là không đơn giane. Để hoàn thành chứng minh tính duy nhất ta chú ý, như được cho biết ở trên, σ_1, v_1 và u_1 được xác định, phần còn lại của SVD được xác định bởi tác động của A vào không gian trực giao với v_1. Vì v_1 là duy nhất nên không gian trực giao này được xác định duy nhất, và tính duy nhất của các giá trị và vector suy biến còn lại theo sau phương pháp quy nạp.

1.5.6 Sự thay đổi của các cơ sở

Cho $b \in \mathbb{C}^m$ bất kỳ có thể được khai triển trong cơ sở của các vector suy biến trái của A (các cột của U), và $x \in \mathbb{C}^n$ bất kỳ có thể được khai triển trong cơ sở của các vector suy biến phải của A (các cột của V). Các vector tọa độ cho các khai triển này là

$$b' = U^*b,\ x' = V^*x.$$

Theo (1.5.3), $b = Ax$ có thể được biểu diễn theo b' và x'

$$b = Ax \Longleftrightarrow U^*b = U^*Ax = U^*U\sum V^*x \Longleftrightarrow b' = \sum x'.$$

Khi $b = Ax$, ta có $b' = \sum x'$. Do đó A rút gọn thành ma trận đường chéo $\sum$ khi vùng được biểu diễn trong cơ sở các cột của U và miền xác định được biểu diễn trong cơ sở các cột của V.

1.5.7 SVD so với phân tích trị riêng

Một ma trận vuông đầy đủ không quan trọng A có thể được biểu diễn như là một ma trận đường chéo của các trị riêng Λ, nếu vùng và miền xác định được biểu diễn trong một cơ sở của các vector riêng.

Nếu các cột của ma trận $X \in \mathbb{C}^{m\times m}$ chứa các vector riêng độc lập tuyến tính của $A \in \mathbb{C}^{m\times m}$, *phân tích trị riêng* của A là

$$A = X\Lambda X^{-1}, \tag{1.5.6}$$

với Λ là ma trận đường chéo $m \times m$ mà các phần tử của nó là các trị riêng của A. Cho b, $x \in \mathbb{C}^m$ thỏa mãn $b = Ax$,

$$b^{'} = X^{-1}b,\ x^{'} = X^{-1}x,$$

thì các vector được khai triển mới $b^{'}$ và $x^{'}$ thỏa mãn $b^{'} = \Lambda x^{'}$.

Có sự khác nhau cơ bản giữa SVD và phân tích trị riêng. Một là SVD sử dụng hai cơ sở khác nhau (các tập hợp của các vector suy biến trái và phải), trong khi đó phân tích trị riêng sử dụng đúng một cơ sở (các vector riêng). Thứ hai là SVD sử dụng cơ sở trực giao, trong khi đó phân tích trị riêng sử dụng một cơ sở nói chung không phải là trực giao. Thứ ba là không phải tất cả các ma trận (ngay cả ma trận vuông) đều có phân tích trị riêng, nhưng tất cả các ma trận (ngay cả ma trận vuông) có phân tích giá trị suy biến, như được thiết lập trong Định lý 1.5.1. Trong các ứng dụng, các trị riêng hướng về các bài toán có liên quan tới các dạng được lặp lại của A, như ma trận lũy thừa A^k hay các hàm mũ e^{tA}, trong khi các vector suy biến hướng về các bài toán có liên quan tới xử lý của chính A hoặc nghịch đảo của nó.

1.5.8 Các tính chất ma trận thông qua SVD

Giả sử A có $m \times n$ chiều. Cho p là số nhỏ nhất của m và n, $r \leqslant p$ ký hiệu số các giá trị suy biến khác 0 của A, và cho $< x, y, \ldots, z >$ là không gian sinh bởi các vector $x, y, \ldots, z$. Khi đó, ta có các định lý sau

Định lý 1.5.2 *$rank(A) = r$, số các giá trị suy biến khác 0.*

Chứng minh Hạng của một ma trận đường chéo là bằng số các phần tử khác 0 của nó, và trong phân tích $A = U\sum V^*$, U và V là hạng đầy đủ. Do đó, $rank(A) = rank(\sum) = r$.

Định lý 1.5.3 *$range(A) =< u_1, \ldots, u_r >$ và $null(A) =< v_{r+1}, \ldots, v_n >$.*

Chứng minh Đây là chuỗi mà $range(\sum) =< e_1, \ldots, e_r > \subseteq \mathbb{C}^m$ và $null(\sum) =< e_{r+1}, \ldots, e_n > \subseteq \mathbb{C}^n$.

Định lý 1.5.4 *$\|A\|_2 = \sigma_1$ và $\|A\|_F = \sqrt{\sigma_1^2 + \sigma_2^2 + \ldots + \sigma_r^2}$.*

Chứng minh Kết quả đầu tiên đã được thiết lập trong chứng minh của Định lý 1.5.1 vì $A = U\sum V^*$ với ma trận Unita U và V, $\|A\|_2 = \|\sum\|_2 = max\{|\sigma_j|\} = \sigma_1$ (do Định lý 1.4.1). Kết quả thứ hai, do Định lý 1.4.1 và nhận xét theo sau, chuẩn Frobenius là bất biến dưới phép nhân Unita nên $\|A\|_F = \|\sum\|_F$, và do (1.4.16), ta có công thức như trên.

Định lý 1.5.5 *Các giá trị suy biến khác không của A là các căn bậc hai của các trị riêng khác không của A^*A hay AA^*. (Các ma trận này có cùng các trị riêng khác không.)*

Chứng minh Từ kết quả tính toán

$$A^*A = (U\sum V^*)^*(U\sum V^*) = V\overset{*}{\sum}U^*U\sum V^* = V(\overset{*}{\sum}\sum)V^*,$$

ta thấy A^*A tương tự với $\sum^*\sum$ và do đó có cùng n trị riêng. Các trị riêng của ma trân đường chéo $\sum^*\sum$ là $\sigma_1^2, \sigma_2^2, \ldots, \sigma_p^2$, với $n-p$ trị riêng 0 thêm vào nếu $n > p$. Tính toán tương tự với m trị riêng của AA^*.

Định lý 1.5.6 *Nếu $A = A^*$ thì các giá trị suy biến của A là giá trị tuyệt đối của các trị riêng của A.*

Chứng minh Một ma trận Hermit là một tập đầy đủ các vector riêng trực giao và tất cả các trị riêng này là thực. Một phát biểu tương đương là (**??**) đúng với X là một ma trận Unita Q bất kỳ và Λ là ma trận đường chéo thực. Khi đó ta có thể viết

$$A = Q\Lambda Q^* = Q|\Lambda|sign(\Lambda)Q^*, \tag{1.5.7}$$

với $|\Lambda|$ và $sign(\Lambda)$ là các ma trận đường chéo mà các phần tử của nó tương ứng là $|\lambda_j|$ và $sign(\lambda_j)$. (Ta có thể đặt $sign(\Lambda)$ bên trái Λ thay vì bên phải.) Vì $sign(\Lambda)Q^*$ là Unita khi Q là Unita, (1.5.7) là SVD của A, với các giá trị suy biến bằng với các phần tử trên đường chéo của $|\Lambda|, |\lambda_j|$. Nếu được như mong muốn thì các số này có thể được sắp xếp thành thứ tự không tăng bằng việc thêm các ma trận hoán vị phù hợp như là các thừa số trong vế trái ma trận Unita của (1.5.7), Q và vế phải ma trận Unita, $sign(\Lambda)Q^*$.

Định lý 1.5.7 *Cho $A \in \mathbb{C}^{m\times m}$, $|det(A)| = \prod_{i=1}^m \sigma_i$.*

Chứng minh Định thức của tích các ma trận vuông là tích các định thức của các thừa số. Hơn nữa, do công thức $U^*U = I$ và tính chất $det(U^*) = (det(U))^*$ nên trị tuyệt đối của định thức của một ma trận Unita thường là 1. Do đó,

$$|det(A)| = |det(U\sum V^*)| = |det(U)||det(\sum)||det(V^*)| = |det(\sum)| = \prod_{i=1}^m \sigma_i.$$

1.5.9 Xấp xỉ ma trận hạng thấp

Định lý 1.5.8 *A là tổng của r ma trận hạng 1:*

$$A = \sum_{j=1}^r \sigma_j u_j v_j^*. \tag{1.5.8}$$

Chứng minh Nếu ta viết $\sum$ như là tổng của r ma trận $\sum_j$, với $\sum_j = diag(0, \ldots, 0, \sigma_j, 0, \ldots, 0)$, thì (1.5.8) theo sau từ (1.5.3).

Có nhiều cách để biểu diễn một ma trận A có m dòng và n cột như là tổng của các ma trận hạng 1. Ví dụ, A có thể được viết như tổng của m dòng của nó, hoặc tổng của n cột của nó, hoặc tổng của mn phần tử của nó. Cho ví dụ khác, khử Gauss giảm A thành tổng của ma trận hạng 1 đầy đủ, một ma trận hạng 1 mà dòng và cột đầu tiên của nó là 0, v.v.

Tuy nhiên, công thức (1.5.8) biểu diễn phân tích thành các ma trận hạng 1 với tích chất sâu hơn: *tổng riêng phần thứ ν thu giữ nhiều năng lượng của A ngay khi có thể thực hiện được.* Phát biểu này đúng với "năng lượng" xác định bởi hoặc là chuẩn 2 hoặc là chuẩn Frobenius. Ta có thể làm nó tỉ mỉ bằng việc đưa ra công thức một bài toán xấp xỉ tốt nhất của một ma trận A bằng các ma trận có hạng nhỏ hơn.

Định lý 1.5.9 *Cho ν bất kỳ với $0 \leqslant \nu \leqslant r$, định nghĩa*

$$A_\nu = \sum_{j=1}^{\nu} \sigma_j u_j v_j^*; \tag{1.5.9}$$

Nếu $\nu = p = \min\{m, n\}$ thì định nghĩa $\sigma_{\nu+1} = 0$. Khi đó

$$\|A - A_\nu\|_2 = \inf_{\substack{B \in \mathbb{C}^{m\times n} \\ rank(B) \leqslant \nu}} \|A - B\|_2 = \sigma_{\nu+1}.$$

Chứng minh Giả sử có B bất kỳ với $rank(B) \leqslant \nu$ sao cho $\|A - B\|_2 < \|A - A_\nu\|_2 = \sigma_{\nu+1}$. Khi đó, có một không gian con $W \subseteq \mathbb{C}^n$ có $(n - \nu)$ chiều sao cho $w \in W \Rightarrow Bw = 0$. Do đó, với $w \in W$ bất kỳ, ta có $Aw = (A - B)w$ và

$$\|Aw\|_2 = \|(A - B)w\|_2 \leqslant \|A - B\|_2\|w\|_2 < \sigma_{\nu+1}\|w\|_2.$$

Do đó, W là không gian con $n-\nu$ chiều với $\|Aw\| < \sigma_{\nu+1}\|w\|$. Nhưng có không gian con $(\nu+1)$ chiều với $\|Aw\| \geqslant \sigma_{\nu+1}\|w\|$, cụ thể là không gian sinh bởi $\nu + 1$ vector suy biến trái đầu tiên của A. Vì tổng các chiều của các không gian này vượt quá n, nên phải có một vector khác 0 nằm trong cả hai, mâu thuẫn.

Định lý 1.5.9 có một giải thích hình học. Xấp xỉ tốt nhất của một siêu ellip bởi một đoạn thẳng là gì? Lấy đoạn thẳng là trục dài nhất. Cái gì là xấp xỉ tốt nhất một ellipsoid 2 chiều? Lấy ellipsoid sinh bởi trục dài nhất và trục dài thứ hai. Tiếp tục trong kiểu này, tại mỗi bước ta cải thiện sự xấp xỉ bằng việc thêm vào xấp xỉ trục lớn nhất của một siêu ellipsoid mà nó chưa tính đến lúc này. Sau r bước, ta đã thu được tất cả của A. Ý tưởng này có các phân nhánh trong các lĩnh vực khác hẳn nhau như nén ảnh và giải tích hàm.

Ta phát biểu kết quả tương tự cho chuẩn Frobenius mà không chứng minh.

Định lý 1.5.10 *Cho ν bất kỳ với $0 \leqslant \nu \leqslant r$, ma trận A_ν của (1.5.8) cũng thỏa mãn*

$$\|A - A_\nu\|_F = \inf_{\substack{B \in \mathbb{C}^{m\times n} \\ rank(B) \leqslant \nu}} \|A - B\|_F = \sqrt{\sigma_{\nu+1}^2 + \ldots + \sigma_r^2}.$$

Bài tập

1. Cho $A \in \mathbb{C}^{m\times n}$, $r \in \mathbb{C}^r$. Hãy đưa ra thuật toán tính cột đầu tiên của $M = (A - x_1 I)\ldots(A - x_r I)$.

2. Chứng minh rằng nếu một ma trận A vừa là ma trận tam giác vừa là ma trận Unita thì A là ma trận đường chéo.

3. Chứng minh rằng nếu $Q = Q_1 + iQ_2$, với $Q_1, Q_2 \in \mathbb{C}^{n\times n}$ thì ma trận $2n \times 2n$

$$Z = \begin{bmatrix} Q_1 & -Q_2 \\ Q_2 & Q_1 \end{bmatrix}$$

   là trực giao.

4. Chứng minh rằng ma trận bất kỳ trong $\mathbb{C}^{m\times n}$ là giới hạn của một chuỗi các ma trận hạng đầy đủ.

5. Chứng minh rằng nếu $A \in \mathbb{C}^{m\times n}$ có hạng là r thì $\|A(A^TA)^{-1}A^T\|_2 = 1$.

6. Chứng minh rằng nếu thêm một dòng khác 0 vào ma trận A thì giá trị suy biến lớn nhất và nhỏ nhất của A đều tăng.

7. Viết chương trình tính định thức của một ma trận vuông. So sánh kết quả chương trình với hàm det trên Matlab.

8. Viết chương trình tạo ngẫu nhiên ma trận đối xứng cấp N.

9. Viết chương trình tạo ngẫu nhiên ma trận trực chuẩn cấp N.

10. Cho ma trận
$$A = \begin{bmatrix} 3 & 0 \\ 0 & -2 \end{bmatrix}$$
   a) Tính SVD của ma trận A.
   b) Viết chương trình vẽ các vector suy biến phải của ma trận V trong phân tích SVD như Hình 1.2.
   c) Viết chương trình vẽ các vector suy biến trái của ma trận U trong phân tích SVD như Hình 1.2.

Chương 2

Phân tích QR và bình phương tối tiểu

2.1 Phép chiếu

2.1.1 Phép chiếu

Một *phép chiếu* là một ma trận vuông P thỏa mãn

$$P^2 = P. \tag{2.1.1}$$

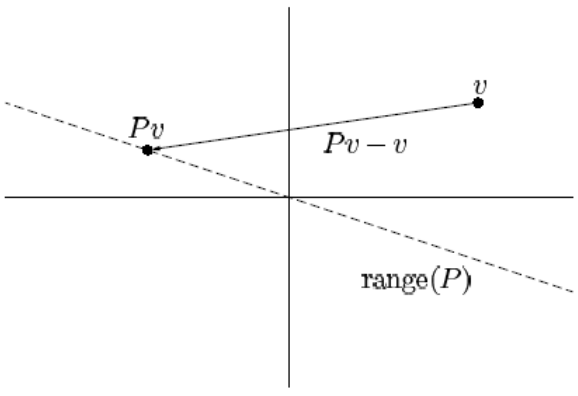
(Ma trận như vậy cũng được nói là ma trận *lũy đẳng*.) Định nghĩa này bao gồm cả phép chiếu trực giao và không trực giao. Để tránh lộn xộn ta sử dụng thuật ngữ *phép chiếu nghiêng* trong trường hợp không trực giao.

Thuật ngữ phép chiếu có được thông qua việc nếu người ta chiếu sánh sáng vào không gian con $range(P)$ từ phương thẳng, thì Pv sẽ là bóng được chiếu bởi vector v.

Quan sát thấy rằng nếu $v \in range(P)$ thì nó nằm một cách chính xác trong cái bóng của nó, và áp dụng các kết quả của phép chiếu trong chính v. Theo toán học, ta có $v = Px$ với x bất kỳ và

$$Pv = P^2x = Px = v.$$

Ánh sáng chiếu vào phương như thế nào khi $v \neq Pv$? Tổng quát, câu trả lời phụ thuộc vào v,



Hình 2.1: Phép chiếu nghiêng

nhưng cho v đặc biệt bất kỳ, nó dễ dàng được suy ra bằng việc vẽ đường từ v tới $Pv, Pv - v$ (Hình 2.1). Việc áp dụng phép chiếu tới vector này cho một kết quả

$$P(Pv - v) = P^2v - Pv = 0.$$

nghĩa là $Pv - v \in null(P)$. Phương của ánh sáng có thể là khác nhau cho v khác nhau, nhưng nó thường được miêu tả bởi một vector trong $null(P)$.

2.1.2 Phép chiếu bù

Nếu P là phép chiếu thì $I - P$ cũng là một phép chiếu, cũng là một lũy đẳng:

$$(I - P)^2 = I - 2P + P^2 = I - P.$$

Ma trận $I - P$ được gọi là *phép chiếu bù* tới P.

Phép chiếu $I - P$ vào không gian đầy đủ của P. Ta biết rằng $range(I - P) \supseteq null(P)$, bởi vì nếu $Pv = 0$, ta có $(I - P)v = v$. Ngược lại, $range(I - P) \subseteq null(P)$ vì với v bất kỳ, ta có $(I - P)v = v - Pv \in null(P)$. Do đó, với phép chiếu P bất kỳ,

$$range(I - P) = null(P). \tag{2.1.2}$$

Bằng việc viết $P = I - (I - P)$ ta suy ra phần bù

$$null(I - P) = range(P). \tag{2.1.3}$$

Ta cũng thấy rằng $null(I - P) \cap null(P) = \{0\}$: vector v bất kỳ trong cả 2 tập thỏa mãn $v = v - Pv = (I - P)v = 0$. Mặt khác,

$$range(P) \cap null(P) = \{0\}. \tag{2.1.4}$$

Một phép chiếu tách $\mathbb{C}^m$ thành 2 không gian. Ngược lại, cho S_1 và S_2 là hai không gian con của $\mathbb{C}^m$ sao cho $S_1 \cap S_2 = \{0\}$ và $S_1 + S_2 = \mathbb{C}^m$, với $S_1 + S_2$ là không gian sinh của S_1 và S_2, nghĩa là, tập các vector $s_1 + s_2$ với $s_1 \in S_1$ và $s_2 \in S_2$. (Một cặp như vậy được nói là *các không gian con bù*). Khi đó có một phép chiếu P thỏa mãn $range(P) = S_1$ và $null(P) = S_2$. Ta nói rằng P là phép chiếu *vào S_1 dọc theo S_2*. Phép chiếu này và bù của nó có thể được xem như là lời giải duy nhất của bài toán theo sau:

Cho v, tìm các vector $v_1 \in S_1$ và $v_2 \in S_2$ sao cho $v_1 + v_2 = v$.

Phép chiếu Pv cho v_1, và phép chiếu bù $(I - P)v$ cho v_2. Các vector này là duy nhất bởi vì tất cả các lời giải phải có dạng

$$(Pv + v_3) + ((I - P)v - v_3) = v,$$

rõ ràng v_3 phải nằm trong cả S_1 và S_2, nghĩa là, $v_3 = 0$.

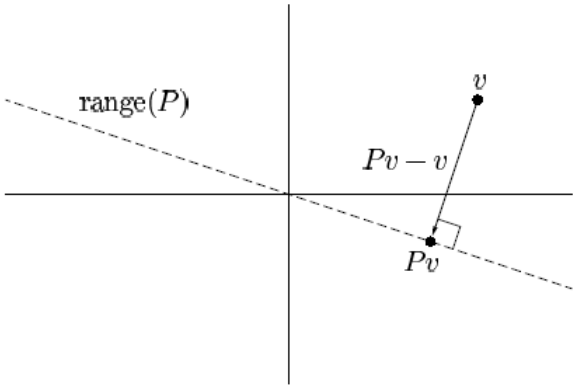
Giả sử ma trận $m \times m$ A có một tập đầy đủ các vector riêng $\{v_j\}$, như trong (**??**), nghĩa là $\{v_j\}$ là một cơ sở của $\mathbb{C}^m$. Chúng thường được liên quan tới các bài toán kết hợp với các khai triển của các vector trong cơ sở này. Cho $x \in \mathbb{C}^m$, ví dụ, thành phần của x trong phương của một vector riêng đặc biệt v là gì? Câu trả lời là Px, với P là phép chiếu hạng 1 nào đó.

2.1.3 Phép chiếu trực giao

Một *phép chiếu trực giao* (Hình 2.2) là một phép chiếu lên một không gian con S_1 dọc theo không gian S_2, với S_1 và S_2 là trực giao. (Lưu ý, Các phép chiếu trực giao không là các ma trận trực giao!)

Định nghĩa đại số: một phép chiếu trực giao là phép chiếu bất kỳ mà nó là Hermit, thỏa mãn $P^* = P$ như trong (2.1.1).

Định lý 2.1.1 *Một phép chiếu P là trực giao nếu và chỉ nếu $P = P^*$.*



Hình 2.2: Phép chiếu trực giao

Chứng minh ($\Rightarrow$) Nếu $P = P^*$ thì tích trong của vector $Px \in S_1$ và vector $(I - P)y \in S_2$ là 0:

$$x^*P^*(I - P)y = x^*(P - P^2)y = 0.$$

Khi đó phép chiếu là trực giao.

($\Rightarrow$) Giả sử P chiếu lên S_1 dọc theo S_2, với $S_1 \perp S_2$ và S_1 có số chiều là n. Khi đó SVD của P có thể được xây dựng như sau. Cho $\{q_1, q_2, \ldots, q_m\}$ là một cơ sở trực giao của $\mathbb{C}^m$, với $\{q_1, \ldots, q_n\}$ là 1 cơ sở của S_1 và $\{q_{n+1}, \ldots, q_m\}$ là một cơ sở của S_2. Cho $j \leqslant n$, ta có $Pq_j = q_j$, và cho $j > n$, ta có $Pq_j = 0$. Cho Q là ma trận Unita mà cột thứ j là q_j. Khi đó ta có

$$PQ = \left[\begin{array}{c|c|c|c|c} q_1 & \ldots & q_n & 0 & \ldots \end{array}\right],$$

để cho

$$Q^*PQ = \begin{bmatrix} 1 & & & & \\ & \ddots & & & \\ & & 1 & & \\ & & & 0 & \\ & & & & \ddots \end{bmatrix} = \sum,$$

một ma trận đường chéo với 1 nằm ở n phần tử đầu tiên và 0 nằm ở những nơi khác. Khi đó ta đã xây dựng một phân tích giá trị suy biến của P:

$$P = Q \sum Q^*. \tag{2.1.5}$$

(Chú ý đây cũng là một phân tích trị riêng (1.5.6). Từ đây ta thấy P là Hermit, vì $P^* = (Q \sum Q^*)^* = Q \sum^* Q^* = Q \sum Q^* = P$.

2.1.4 Phép chiếu với cơ sở trực giao

Vì phép chiếu trực giao có một vài giá trị suy biến bằng 0 (ngoại trừ trường hợp tầm thường $P = I$), nên các cột của Q trong (2.1.5) và sử dụng SVD được giảm, ta thu được biểu thức đơn giản

$$P = \widehat{Q}\widehat{Q}^*, \tag{2.1.6}$$

với các cột của $\widehat{Q}$ là trực giao.

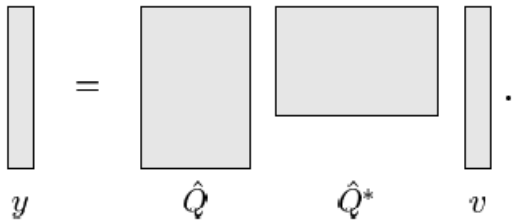
Trong (2.1.6), ma trận $\widehat{Q}$ không cần thiết đến từ SVD. Cho $\{q_1, \ldots, q_n\}$ là một tập bất kì của n vecotor trực giao trong $\mathbb{C}^m$, và cho $\widehat{Q}$ là ma trận $m \times n$ tương ứng. Từ (1.3.7), ta biết rằng

$$v = r + \sum_{i=1}^{n} (q_i q_i^*) v$$

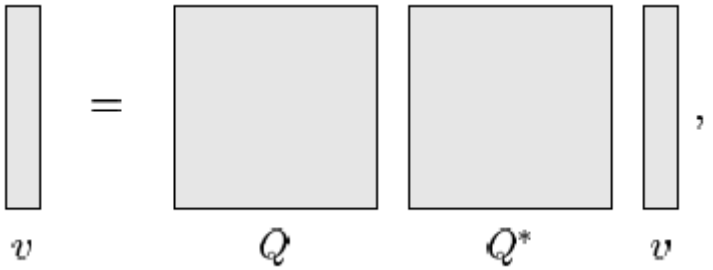
biểu diễn phân tích của vector $v \in \mathbb{C}^m$ thành một phân tích trong không gian cột của $\widehat{Q}$ cộng với một phân tích trong không gian trực giao. Do đó ánh xạ

$$v \mapsto \sum_{i=1}^{n} (q_i q_i^*) v \tag{2.1.7}$$

là một phép chiếu trực giao vào $range(\widehat{Q})$, và trong dạng ma trận, nó có thể được viết $y = \widehat{Q}\widehat{Q}^* v$



Do đó tích $\widehat{Q}\widehat{Q}^*$ bất kỳ thường là một phép chiếu vào không gian cột của $\widehat{Q}$, bất chấp thu được $\widehat{Q}$ như thế nào, miễn là các cột của nó là trực giao. Có thể $\widehat{Q}$ được thu được bằng việc giảm một vài cột và dòng từ phân tích đầy đủ $v = \widehat{Q}\widehat{Q}^* v$ và có thể là không.



Phần bù của một phép chiếu trực giao cũng là một phép chiếu trực giao (chứng minh: $I - \widehat{Q}\widehat{Q}^*$ là hermit). Các phép chiếu bù vào không gian trực giao tới $range(\widehat{Q})$.

Một trường hợp đặc biệt quan trọng của các phép chiếu trực giao là phép chiếu trực giao hạng 1 tách thành phần trong một phương q

$$P_q = qq^*. \tag{2.1.8}$$

Các phần bù của chúng là các phép chiếu trực giao giao hạng $m-1$ mà chúng ước lượng thành phần trong phương của q:

$$P_{\perp q} = I - qq^*. \tag{2.1.9}$$

Phương trình (2.1.8) và (2.1.9)cho rằng q là một vector đơn vị. Cho môt vector a khác không tùy ý, các công thức tương tự là

$$P_a = \frac{aa^*}{a^*a}, \tag{2.1.10}$$

$$P_{\perp a} = I - \frac{aa^*}{a^*a}. \tag{2.1.11}$$

2.1.5 Phép chiếu với cơ sở tùy ý

Một phép chiếu trực giao vào một không gian con của $\mathbb{C}^m$ cũng có thể được xây dựng với một cơ sở tùy ý, không cần thiết là trực giao. Giả sử không gian con được sinh bởi các vector độc lập tuyến tính $\{a_1, \ldots, a_n\}$, và cho A là ma trận $m \times n$ mà cột thứ j của nó là a_j.

Ngẫu nhiên từ v tới phép chiếu trực giao $y \in range(A)$ của nó, $y - v$ phải trực giao với $range(A)$. Điều này tương đương với phát biểu y phải thỏa mãn $a_j^*(y - v) = 0$ với mọi j. Vì $y \in range(A)$ nên ta có thể đặt $y = Ax$ và viết điều kiện này như $a_j^*(Ax - v) = 0$ với mọi j, mà nó tương đương với $A^*(Ax - v) = 0$ hoặc $A^*Ax = A^*v$. Dễ dàng thấy vì A có hạng đầy đủ nên A^*A là không suy biến. Do đó

$$x = (A^*A)^{-1}A^*v. \tag{2.1.12}$$

Cuối cùng, phép chiếu của $v, y = Ax$, là $y = A(A^*A)^{-1}A^*v$. Do đó phép chiếu trực giao vào $range(A)$ có thể được biểu diễn bởi công thức

$$P = A(A^*A)^{-1}A^*A. \tag{2.1.13}$$

Chú ý điều này là tổng quá hóa nhiều chiều của (2.1.10). Trong trường hợp trực giao $A = \widehat{Q}$, số hạng trong dấu ngoặc đơn trở thành đơn vị và ta được (2.1.6).

2.2 Phân tích QR

2.2.1 Phân tích QR được giảm

Các không gian cột của ma trận A là các không gian *liên tiếp* được sinh bởi các cột $a_1, a_2, \ldots$ của A:

$$< a_1 > \subseteq < a_1, a_2 > \subseteq < a_1, a_2, a_3 > \subseteq \ldots$$

Do đó, $< a_1 >$ là không gian 1 chiều sinh bởi a_1, $< a_1, a_2 >$ là không gian 2 chiều sinh bởi a_1 và a_2, ... Ý tưởng của phân tích QR là xây dựng một chuỗi các vector trực giao $q_1, q_2, \ldots$ mà nó sinh ra các không gian liên tiếp này.

Giả sử $A \in \mathbb{C}^{m\times n} (m \geqslant n)$ có hạng đầy đủ n. Ta muốn chuỗi $q_1, q_2, \ldots$ có tính chất

$$< q_1, q_2, \ldots, q_j > = < a_1, a_2, \ldots, a_j >,\ j = 1, \ldots, n. \tag{2.2.1}$$

Từ mục 1.2, ta có điều kiện

$$\left[\begin{array}{c|c|c|c} a_1 & a_2 & \ldots & a_n \end{array}\right] = \left[\begin{array}{c|c|c|c} q_1 & q_2 & \ldots & q_n \end{array}\right] \begin{bmatrix} r_{11} & r_{12} & \ldots & r_{1n} \\ & r_{22} & & \\ & & \ddots & \vdots \\ & & & r_{nn} \end{bmatrix}, \tag{2.2.2}$$

với các phần tử đường chéo r_{kk} khác 0 - nếu (2.2.2) đúng thì $a_1, \ldots, a_k$ có thể được biểu diễn như là tổ hợp tuyến tính của $q_1, \ldots, q_k$, và nghịch đảo của khối $k \times k$ ở trên bên trái của ma trận tam giác. Do đó, $q_1, \ldots, q_k$ có thể được biểu diễn như tổ hợp tuyến tính của $a_1, \ldots, a_k$. Các phương trình này có dạng

$$\begin{aligned} a_1 &= r_{11}q_1, \\ a_2 &= r_{12}q_1 + r_{22}q_2, \\ a_3 &= r_{13}q_1 + r_{23}q_2 + r_{33}q_3, \\ &\vdots \\ a_n &= r_{1n}q_1 + r_{2n}q_2 + \ldots + r_{nn}q_n. \end{aligned} \tag{2.2.3}$$

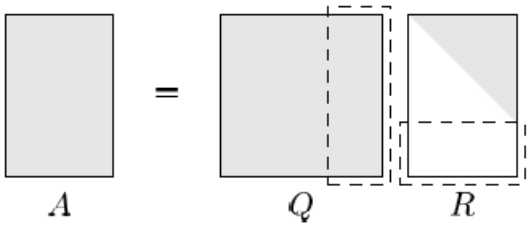
Khi đó, ta có

$$A = \widehat{Q}\widehat{R}, \tag{2.2.4}$$

với $\widehat{Q}$ là ma trận $m \times n$ với các cột trực giao và $\widehat{R}$ là ma trận tam giác trên $n \times n$. Phân tích như vậy được gọi là *phân tích QR được giảm của A*.

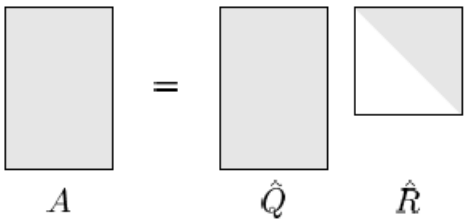
2.2.2 Phân tích QR đầy đủ

Phân tích QR đầy đủ của $A \in \mathbb{C}^{m\times n}(m \geqslant n)$ là việc thêm $m - n$ cột trực giao vào $\widehat{Q}$ sao cho nó trở thành ma trận Unita Q $m \times m$. Tương tự SVD được giảm thành SVD đầy đủ ở trong mục trước. Các dòng 0 được thêm vào $\widehat{R}$ để nó trở thành ma trận R có $m \times n$, vẫn là ma trận tam giác trên. Phân tích QR đầy đủ và được giảm có quan hệ như sau Trong phân tích



Hình 2.3: Phân tích QR đầy đủ ($m \geqslant n$)

QR đầy đủ, Q là ma trận $m \times m$, R là ma trận $m \times n$, và $m - n$ cột cuối cùng của Q được nhân với 0 trong R (bao bọc bởi các đường đứt nét). Trong phân tích QR được giảm, các cột và các dòng không được nói đến bị loại bỏ. Ma trận $\widehat{Q}$ là ma trận $m \times n$, $\widehat{R}$ là ma trận $n \times n$, và không dòng nào của $\widehat{R}$ là 0.



Hình 2.4: Phân tích QR được giảm

Chú ý trong phân tích QR đầy đủ, các cột q_j với $j > n$ trực giao với $range(A)$. Giả sử A là ma trận có hạng đầy đủ n thì chúng tạo thành một cơ sở trực giao cho $range(A)^{\perp}$ (không gian trực giao với $range(A)$), hoặc tương đương, cho $null(A^*)$.

2.2.3 Trực giao hóa Gram - Schmidt

Phương trình (2.2.3) đưa ra một phương pháp cho việc tính phân tích QR được giảm. Cho $a_1, a_2, \ldots,$ ta có thể xây dựng các vector $q_1, q_2, \ldots$ và các phần tử r_{ij} bằng một qúa trình trực giao hóa liên tiếp, được biết như *trực giao hóa Gram - Schmidt*.

Tại bước thứ j, ta mong tìm một vector đơn vị $q_j \in < a_1, \ldots, a_j >$ trực giao với $q_1, \ldots, q_{j-1}$. Khi điều này xảy ra, ta đã xét kỹ thuật trực giao hóa cần thiết trong (1.3.6). Từ phương trình

đó, ta thấy rằng

$$v_j = a_j - (q_1^* a_j)q_1 - (q_2^*)q_2 - \ldots - (q_{j-1}^* a_j)q_{j-1} \tag{2.2.5}$$

là một loạivector được yêu cầu, ngoại trừ nó không được trực chuẩn hóa. Nếu ta chia cho $\|v_j\|_2$ thì kết quả là một vector phù hợp q_j.

Ta viết lại 2.2.3 thành dạng

$$\begin{aligned} q_1 &= \frac{a_1}{r_{11}}, \\ q_2 &= \frac{a_2 - r_{12}q_1}{r_{22}}, \\ q_3 &= \frac{a_3 - r_{13}q_1 - r_{23}q_2}{r_{33}}, \\ &\vdots \\ q_n &= \frac{a_n - \sum_{i=1}^{n-1} r_{in}q_i}{r_{nn}}. \end{aligned} \tag{2.2.6}$$

Từ (2.2.5), một định nghĩa xấp xỉ cho các hệ số r_{ij} trong các tử số của (2.2.6) là

$$r_{ij} = q_i^* a_j \ (i \neq j). \tag{2.2.7}$$

Các hệ số r_{ij} trong các mẫu số được chọn cho sự trực chuẩn hóa:

$$|r_{ij}| = \|a_j - \sum_{i=1}^{j-1} r_{ij}q_i\|_2. \tag{2.2.8}$$

Chú ý dấu của r_{ij} không được xác định nên ta có thể chọn $r_{ij} > 0$, trong trường hợp mà ta sẽ hoàn thành phân tích $A = \widehat{Q}\widehat{R}$ mà $\widehat{R}$ có các phần tử dương trên đường chéo.

Thuật toán được thể hiện trong (2.2.6) - (2.2.8) là bước lặp Gram - Schmidt. Theo toán học, nó đưa ra một dây truyền đơn giản để hiểu và chứng minh các tính chất khác nhau của các phân tích QR. Theo số học, nó trả ra kết quả là không ổn định bởi vì việc làm tròn các sai số trong máy tính. Để nhấn mạnh tính không ổn định, các nhà phân tích số xem điều này như *bước lặp Gram - Schmidt cổ điển*, đối lặp với *bước lặp Gram - Schmidt được sửa đổi.*

Thuật toán 2.1 Gram - Schmidt cổ điển (không ổn định)

```
for j = 1 to n do
    v_j = a_j
    for i = 1 to j − 1 do
        r_ij = q_i^* a_j
        v_j = v_j − r_ij q_i
    end for
    r_ij = ‖v_j‖_2
    q_j = v_j / r_jj
end for
```

2.2.4 Sự tồn tại và tính duy nhất

Tất cả các ma trận có các phân tích QR, và dưới các hạn chế phù hợp, chúng là duy nhất. Ta bắt đầu kết quả tồn tại đầu tiên.

Định lý 2.2.1 *Mọi $A \in \mathbb{C}^{m\times n}$ $(m \geqslant n)$ có một phân tích QR đầy đủ, do đó cũng có một phân tích QR được giảm.*

Chứng minh Giả sử A có hạng đầy đủ và ta muốn phân tích QR được giảm. Trong trường hợp này, chứng minh tồn tại được cung cấp bởi thuật toán Gram - Schmidt. Quá trình này sinh ra các cột trực giao của $\widehat{Q}$ và các phần tử của $\widehat{R}$ sao cho (2.2.4) đúng. Thất bại có thể xảy ra khi tại bước bất kì, v_j là 0 và do đó nó không thể được trực chuẩn hóa để đưa ra q_j.

Tuy nhiên, điều này sẽ kéo theo $a_j \in < q_1, \ldots, q_{j-1} > = < a_1, \ldots, a_{j-1} >$, mâu thuẫn với giả thuyết A có hạng đầy đủ.

Giả sử rằng A không có hạng đầy đủ. Khi đó ít nhất một bước j, từ(2.2.5) cho $v_j = 0$. Bây giờ, ta chọn một cách đơn giản q_j tùy ý để là vector được chuẩn hóa bất kỳ trực giao với $< q_1, \ldots, q_{j-1} >$, và khi đó tiếp tục quá trình Gram - Schmidt.

Cuối cùng, phân tích QR đầy đủ của một ma trận $m \times n$ với $m > n$ có thể được xây dựng bằng việc đưa ra các vector trực giao tùy ý trong mô hình tương tự. Quá trình Gram - Schmidt qua bước n, khi đó tiếp tục thêm vào $m - n$ bước, đưa ra các vector q_j tại mỗi bước.

Bây giờ ta chuyển sang tính duy nhất. Giả sử $A = \widehat{Q}\widehat{R}$ là một phân tích QR được giảm. Nếu cột thứ i của $\widehat{Q}$ được nhân với z và dòng thứ i của $\widehat{R}$ được nhân với z^{-1} với vô hướng z bất kì thỏa $|z| = 1$, ta được phân tích QR khác của A. Định lý tiếp theo khẳng định rằng nếu A có hạng đầy đủ thì điều này chỉ cách để thu được các phân tích QR được giảm phân biệt.

Định lý 2.2.2 *Mỗi $A \in \mathbb{C}^{m\times n}$ $(m \geqslant n)$ hạng đầy đủ có duy nhất một phân tích QR được giảm $A = \widehat{Q}\widehat{R}$ với $r_{ij} > 0$.*

Chứng minh Nhắc lại, chứng minh được cung cấp bởi bước lặp Gram - Schmidt. Từ (2.2.4), tính trực giao của các cột của $\widehat{Q}$, và tính chất tam giác trên của $\widehat{R}$, phân tích QR được giảm bất kỳ của A phải thỏa (2.2.6) - (2.2.8). Theo giả thuyết hạng đầy đủ, các mẫu số (2.2.8) của (2.2.6) là khác 0, và do đó tại mỗi bước j liên tiếp, các công thức này xác định một cách đầy đủ r_{ij} và q_j, ngoài trừ dấu của r_{ij} chưa được chỉ định trong (2.2.8). Điều này được cố định bằng điều kiện $r_{ij} > 0$ như trong Thuật toán 2.1, phân tích được xác định một cách đầy đủ.

2.2.5 Khi các vector trở thành các hàm liên tục

Giả sử ta thay thế $\mathbb{C}^m$ bằng $L^2[-1, 1]$, không gian vector của các hàm có giá trị phức trong [-1,1]. Ta sẽ không đưa ra các tính chất của không gian này. Tích trong của f và g có dạng

$$(f, g) = \int_{-1}^{1} \overline{f(x)} g(x) dx. \tag{2.2.9}$$

Ví dụ, xét "ma trận" theo sau mà "các cột" của nó là các đơn thức x^j:

$$A = \left[1 \middle| x \middle| x^2 \middle| \ldots \middle| x^{n-1} \right], \tag{2.2.10}$$

Mỗi cột là một hàm trong $L^2[-1, 1]$. Do đó, trong khi A là rời rạc như trong phương nằm ngang thông thường, nó liên tục trong phương thẳng đứng. Nó là mô hình liên tục của các ma trận Vandermonde (4.3.4) của mục Ví dụ (1.2.3).

"Phân tích QR liên tục" của A có dạng

$$A = QR = \left[q_0(x) \middle| q_1(x) \middle| \ldots \middle| q_{n-1}(x) \right] \begin{bmatrix} r_{11} & r_{12} & \ldots & r_{1n} \\ & r_{22} & & \\ & & \ddots & \vdots \\ & & & r_{nn} \end{bmatrix},$$

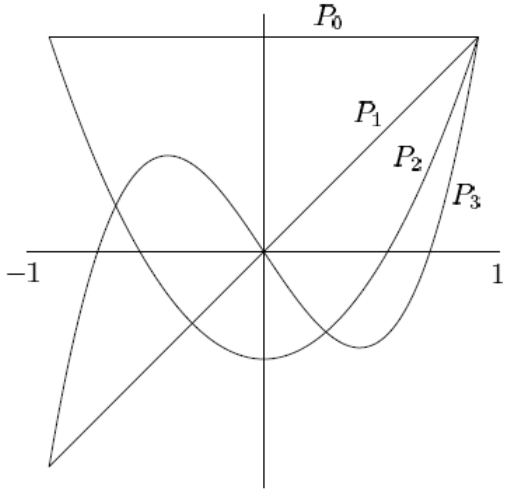
với các cột của Q là các hàm của x, trực giao đối với tích trong (2.2.9)

$$\int_{-1}^{1} \overline{q_i(x)} q_j(x) dx = \delta_{ij} = \begin{cases} 1 & \text{nếu } i = j, \\ 0 & \text{nếu } i \neq j. \end{cases}$$

Từ xây dựng Gram - Schmidt ta có thể thấy rằng q_j là một đa thức bậc j. Các đa thức này là bội vô hướng của *các đa thức Legendre*, P_j, mà chúng được trực chuẩn để $P_j(1) = 1$. Một vài P_j đầu tiên là

$$P_0(x) = 1,\ P_1(x) = x,\ P_2(x) = \frac{3}{2}x^2 - \frac{1}{2},\ P_3(x) = \frac{5}{2}x^3 - \frac{3}{2}x; \tag{2.2.11}$$

thấy trong Hình 2.5. Giống như các đơn thức $1, x, x^2, \ldots$, chuỗi này của các đa thức sinh ra các không gian của các đa thức bậc cao hơn liên tiếp nhau. Tuy nhiên, $P_0(x), P_1(x), P_2(x), \ldots$ là trực giao nhau. Thật vậy, tính toán với các đa thức như vậy tạo thành cơ sở trực giao của *các phương pháp phổ*, một trong những kỹ thuật mạnh nhất cho lời giải số của các phương trình đạo hàm riêng.



Hình 2.5: Bốn đa thức Legendre đầu tiên trong (2.2.11) ($[1, x, x^2, x^3]$)

"Phép chiếu ma trận" $\widehat{Q}\widehat{Q}^*$ 2.1.6 kết hợp với $\widehat{Q}$ là một "ma trận $[-1, 1] \times [-1, 1]$", nghĩa là, một toán tử tích phân

$$f(\cdot) \mapsto \sum_{j=0}^{n-1} q_j(\cdot) \int_{-1}^{1} \overline{q_j(x)} f(x) dx \tag{2.2.12}$$

ánh xạ các hàm trong $L^2[-1, 1]$ vào các hàm trong $L^2[-1, 1]$.

2.2.6 Giải phương trình $Ax = b$ bằng phân tích QR

Giả sử ta muốn giải phương trình $Ax = b$ cho biến x, với $A \in \mathbb{C}^{m \times m}$ là không suy biến. Nếu $A = QR$ là một phân tích QR thì ta có thể viết $QRx = b$, hoặc

$$Rx = Q^*b. \tag{2.2.13}$$

Vế bên phải của phương trình này tính dễ dàng, nếu biết Q, và hệ phương trình tuyến tính ẩn trong vế bên trái cũng giải dễ dàng bởi vì nó là tam giác. Giải phương trình $Ax = b$:

1. Tính phân tích QR $A = QR$.
2. Tính $y = Q^*b$.
3. Giải $Rx = y$ cho x.

2.3 Trực giao hóa Gram - Schmit

2.3.1 Phép chiếu Gram - Schmidt

Cho $A \in \mathbb{C}^{m\times n}, m \geqslant n$, là ma trận có hạng đầy đủ với các cột $\{a_j\}$. Trước đó, ta biểu diễn bước lặp Gram - Schmidt bằng các công thức (2.2.6) - (2.2.8). Xét chuỗi công thức

$$q_1 = \frac{P_1 a_1}{\|P_1 a_1\|}, \quad q_2 = \frac{P_2 a_2}{\|P_2 a_2\|}, \ldots, q_n = \frac{P_n a_n}{\|P_n a_n\|}. \tag{2.3.1}$$

Trong các công thức này, mỗi P_j là một phép chiếu trực giao. Đặc biệt, P_j là ma trận $m \times m$ có hạng $m - (j-1)$ mà nó chiếu trực giao $\mathbb{C}^m$ vào không gian trực giao với $< q_1, \ldots, q_{j-1} >$. (Trong trường hợp $j = 1$, $P_1 = I$). Ta thấy q_j được xác định như trong (2.3.1) là trực giao với $q_1, \ldots, q_{j-1}$, nằm trong không gian $< a_1, \ldots, a_j >$, và có chuẩn bằng 1. Khi đó, (2.3.1) tương đương với (2.2.6) - (2.2.8) và do đó tương đương với Thuật toán 2.1.

Cho $\widehat{Q}_{j-1}$ là ma trận $m \times (j-1)$ chứa $j-1$ cột đầu tiên của $\widehat{Q}$,

$$\widehat{Q}_{j-1} = \left[q_1 \middle| q_2 \middle| \cdots \middle| q_{j-1} \right]. \tag{2.3.2}$$

Khi đó P_j được cho bởi

$$P_j = I - \widehat{Q}_{j-1}\widehat{Q}^*_{j-1}. \tag{2.3.3}$$

2.3.2 Thuật toán Gram - Schmidt được sửa đổi

Trong thực hành, Các công thức Gram - Schmidt không được áp dụng như ta đã chỉ thị trong Thuật toán 2.1 và trong (2.3.1), với các tính toán của chuỗi này thì kết quả trả ra là số không ổn định. May thay, có sự sửa đổi đơn giản mà nó cải thiện các vấn đề này. Ta sẽ không thảo luận số ổn định lúc này mà sẽ thảo luận ở các mục sau. Hiện tại, nó là một thuật toán ổn định, không quá chính xác tới hiệu quả của việc làm tròn các sai số trong máy tính.

Với mỗi giá trị j, Thuật toán 2.1 tính phép chiếu trực giao đơn có hạng $m - (j-1)$,

$$v_j = P_j a_j. \tag{2.3.4}$$

Ngược lại, thuật toán Gram - Schmidt được sửa đổi tính kết quả giống nhau bằng một chuỗi $j-1$ phép chiếu có hạng $m-1$. Nhắc lại từ (2.1.9), $P_{\perp q}$ là phép chiếu trực giao có hạng $m-1$ lên không gian trực giao với một vector $q \in \mathbb{C}^m$ khác 0. Theo định nghĩa của P_j,

$$P_j = P_{\perp q_{j-1}} \ldots P_{\perp q_2} P_{\perp q_1}, \tag{2.3.5}$$

nhắc lại $P_1 = I$. Do đó một phát biểu tương đương với (2.3.4) là

$$v_j = P_{\perp q_{j-1}} \ldots P_{\perp q_2} P_{\perp q_1} a_j. \tag{2.3.6}$$

Thuật toán Gram - Schmidt được sửa đổi sử dụng (2.3.6) thay vì (2.3.4).

Theo toán học, (2.3.4) và (2.3.6) tương đương nhau. Tuy nhiên, các chuỗi phép toán số học bao hàm bởi công thức này là khác nhau. Thuật toán được sửa đổi tính v_j bằng việc đánh giá các công thức sau

$$\begin{aligned} v_j^{(1)} &= a_j, \\ v_j^{(2)} &= P_{\perp q_1} v_j^{(1)} = v_j^{(1)} - q_1 q_1^* v_j^{(1)}, \\ v_j^{(3)} &= P_{\perp q_1} v_j^{(2)} = v_j^{(2)} - q_2 q_2^* v_j^{(2)}, \\ &\vdots \ \vdots \\ v_j = v_j^{(j)} &= P_{\perp q_{j-1}} v_j^{(j-1)} = v_j^{(j-1)} - q_{j-1} q_{j-1}^* v_j^{(j-1)}. \end{aligned} \tag{2.3.7}$$

Trong số học tính toán độ chính xác hữu hạn, ta sẽ thấy rằng (2.3.7) đưa ra các sai số nhỏ hơn (2.3.4).

Khi thuật toán được thực thi, phép chiếu $P_{\perp q_i}$ có thể được ứng dụng thuận lợi cho $v_j^{(i)}$ với $j > i$ ngay sau khi q_i được biết.

Thuật toán 2.2 Gram - Schmidt được sửa đổi

```
for i = 1 to n do
    v_i = a_i
    for i = 1 to n do
        r_ii = ||v_i||
        q_i = v_i / r_ii
        for j = i + 1 to n do
            r_ij = q_i^* v_j
            v_j = v_j − r_ij q_i
        end for
    end for
end for
```

Trong thực hành, nó thường là để đặt v_i ghi đè a_i và q_i ghi đè v_i để lưu không gian.

2.3.3 Đếm số phép toán

Đếm số phép toán dấu chấm động - "flops" - mà thuật toán yêu cầu. Mỗi phép cộng, phép trừ, phép nhân, phép chia hoặc căn bậc hai đếm như là một phép toán dấu chấm động. Ta không phân biệt giữa số học thực và phức, mặc dù trong thực hành trong hầu hết các máy tính có sự khác nhau khá lớn.

Thật vậy, có nhiều chi phí của thuật toán hơn là đếm số phép toán. Trong máy tính xử lý đơn, thời gian thực thi bị ảnh hưởng bởi sự di chuyển dữ liệu giữa các phần tử của hệ thống cấp bậc trong bộ nhớ và việc cạnh tranh các công việc đang chạy trong cùng một xử lý. Trong các máy hệ thống đa xử lý việc này trở nên phức tạp hơn, sự giao tiếp giữa các xử lý thỉnh thoảng đưa thông tin quan trọng lớn hơn nhiều của các "tính toán" hiện nay.

Định lý 2.3.1 *Thuật toán 2.1 và 2.2 cần $\sim 2mn^2$ phép toán dấu chấm động để tính phân tích QR của một ma trận A có $m \times n$.*

Ký hiệu "$\sim$" có nghĩa tiệm cận thông thường của nó:

$$\lim_{m,n\to\infty} \frac{\text{số phép toán dấu chấm động}}{2mn^2} = 1.$$

Định lý 2.3.1 có thể được thiết lặp như sau. Để xác định, xét thuật toán Gram - Schmidt được sửa đổi (thuật toán 2.2). Khi m và n là lớn, các phép toán trong vòng lặp ở trong cùng:

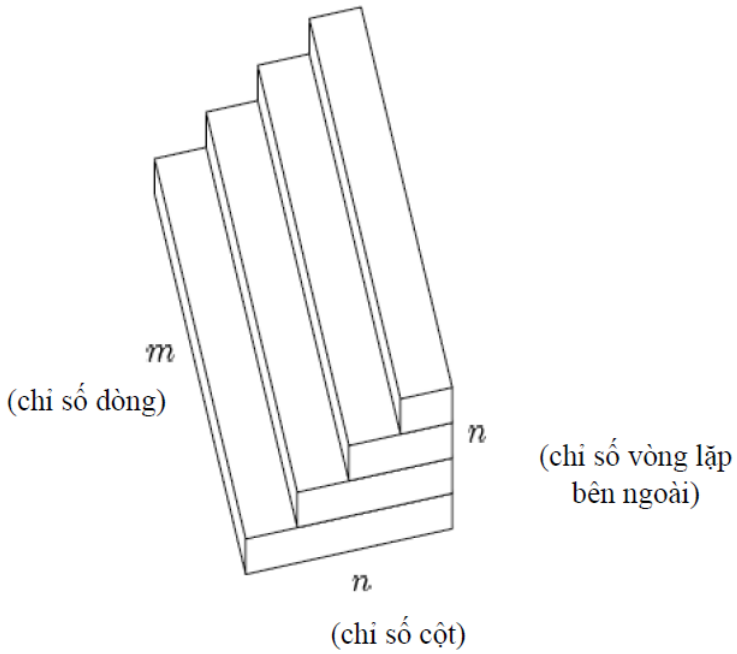
$$r_{ij} = q_i^* v_j,$$
$$v_j = v_j - r_{ij} q_i.$$

Dòng đầu tiên tính một tích trong $q_i^* v_j$ cần m phép nhân và $m-1$ phép cộng. Dòng thứ hai tính $v_j - r_{ij} q_i$ cần m phép nhân và m phép trừ. Tổng số việc được bao gồm môt bước lặp đơn bên trong là $\sim 4m$ phép toán dấu chấm động, hay 4 phép toán dấu chấm động trên phần tử vector cột. Do đó, số phép toán dấu chấm động cần cho thuật toán là tiệm cận

$$\sum_{i=1}^{n} \sum_{j=i+1}^{n} 4m \sim \sum_{i=1}^{n} (i) 4m \sim 2mn^2. \tag{2.3.8}$$

2.3.4 Đếm số phép toán theo hình học

Tại bước đầu tiên của vòng lặp bên ngoài, Thuật toán 2.2 chạy trong toàn bộ ma trận, việc trừ bôi của cột 1 từ các cột khác. Tại bước thứ hai, nó tính toán trong một ma trận con, trừ một bội của cột 2 từ cột $3, \ldots, n$. Tiếp tục cách này, tại mỗi bước số chiều cột rút lại bởi 1 cho tới bước cuối cùng, chỉ cột n được sửa đổi. Thủ tục này có thể được biểu diễn bằng biểu đồ theo sau: Hình chữ nhật $m \times n$ tại đáy tương ứng bước đầu tiên qua vòng lặp bên ngoài, hình



chữ nhật $m \times (n-1)$ ở trên nó tương ứng bước thứ hai, ...

Khi $m, n \to \infty$, số phép toán trực giao hoán Gram - Schmidt tỉ lệ với thể tích của hình ở trên. Hai bước của vòng lặp bên trong tương ứng với 4 phép toán tại vị trí mỗi ma trận nên hằng số của tỉ lệ thức là 4 flop . Ngay khi $m, n \to \infty$, hình hội tụ tới lăng trụ đều bên phải, với thể tích $mn^2/2$. Nhân với 4 flop trên 1 đơn vị thể tích

$$\text{Trực giao hóa Gram - Schmidt: } \sim 2mn^2 \text{ flop.} \tag{2.3.9}$$

2.3.5 Gram - Schmidt như trực giao hóa tam giác

Mỗi bước bên ngoài của thuật toán Gram - Schmidt được sửa đổi có thể được làm sáng tỏ như phép nhân phải với một ma trận tam giác trên vuông. Ví dụ, bắt đầu với A, bước lặp đầu tiên nhân cột đầu tiên a_1 với $1/r_{11}$ và khi đó trừ r_{1j} lần kết quả từ mỗi cột còn lại a_j. Điều này tương đương với phép nhân phải với ma trận R_1:

$$\left[\begin{array}{c|c|c|c} & & & \\ v_1 & v_2 & \dots & v_n \\ & & & \end{array}\right] \begin{bmatrix} \frac{1}{r_{11}} & \frac{-r_{12}}{r_{11}} & \frac{-r_{13}}{r_{11}} & \dots \\ & 1 & & \\ & & 1 & \\ & & & \ddots \end{bmatrix} = \left[\begin{array}{c|c|c|c} & & & \\ q_1 & v_2^{(2)} & \dots & v_n^{(2)} \\ & & & \end{array}\right].$$

Tổng quát, bước thứ i của Thuật toán 2.2 trừ r_{ij}/r_{ii} lần cột i của A từ các cột $j > i$ và thay thế cột i bằng $1/r_{ii}$ lần cột i. Điều này tương ứng với phép nhân ma trận với ma trận tam giác trên R_i:

$$R_2 = \begin{bmatrix} 1 & & & \\ & \frac{1}{r_{22}} & \frac{-r_{23}}{r_{22}} & \dots \\ & & 1 & \\ & & & \ddots \end{bmatrix}, \; R_3 = \begin{bmatrix} 1 & & & \\ & 1 & & \\ & & \frac{1}{r_{33}} & \\ & & & \ddots \end{bmatrix}, \dots$$

Tại bước lặp cuối cùng, ta có

$$A \underbrace{R_1 R_2 \dots R_n}_{\widehat{R}^{-1}} = \widehat{Q}. \tag{2.3.10}$$

Công thức này chứng minh rằng thuật toán Gram - Schmidt là một phương pháp của *trực giao hóa tam giác*. Nó áp dụng các phép toán tam giác vào bên phải của một ma trận để giảm nó thành một ma trận với các cột trực giao. Dĩ nhiên, trong thực hành, ta không làm thành các ma trận R_i và nhân chúng với nhau rõ ràng.

2.4 Matlab

2.4.1 Matlab

Matlab là một ngôn ngữ cho các tính toán toán học mà các loại dữ liệu cơ bản của nó là các vector và các ma trận. Nó tính toán tại mức toán học cao hơn, bao gồm hàng trăm phép toán như ma trận khả nghịch, phân tích giá trị suy biến, và biến đổi Fourier nhanh như các dòng lệnh được xây dựng. Nó cũng là một môi trường giải bài toán, xử lý các lời lệnh mức cao nhất bằng một bộ diễn dịch hơn là một trình biên dịch và cung cấp nội dòng truy cập tới đồ họa 2D và 3D.

Từ những năm 1980, Matlab đã trở thành một công cụ phổ biến của các nhà phân tích số và các kỹ sư trên thế giới. Với nhiều bài toán tính toán khoa học phạm vi lớn. Với thực nghiệm phạm vi trung bình và nhỏ trong, nó được chọn cho phương pháp số cho đại số tuyến tính.

Trong sách này, bây giờ ta sử dụng Matlab và đưa ra các thực nghiệm.

2.4.2 Thực nghiệm 1: Các đa thức Legendre rời rạc

Trong mục 2.2 ta xét "ma trận" Vandermonde với "các cột" gồm có các đơn thức $1, x, x^2$ và x^3 trong khoảng [-1,1]. Giả sử, ta rời rạc hóa [-1,1] bằng 257 điểm đều nhau. Các dòng theo sau của Matlab xây dựng ma trận này và tính phân tích QR được giảm của nó.

`x = (-128:128)'/128;`	x là một rời rạc hóa của [-1,1].
`A = [x.^0 x.^1 x.^2 x.^3];`	Xây dựng ma trận Vandermonde.
`[Q,R] = qr(A,0);`	Tìm phân tích QR được giảm của nó.

Trong dòng đầu tiên, dấu ' chuyển (-128:128) từ một dòng thành một vector cột. Trong dòng thứ 2, các chuỗi .^ cho biết các lũy thừa *entrywise*. Trong dòng thứ 3, qr là một hàm Matlab được xây dựng sẵn cho việc tính các phân tích QR, đối số 0 cho biết một phân tích QR được giảm hơn là phân tích QR đầy đủ. Phương pháp sử dụng ở đây không phải là trực giao hóa Gram - Schmidt mà là tam giác hóa Householder, được thảo luận trong mục tiếp theo, nhưng đây không phải là kết quả cho mục đích hiện tại. Trong cả 3 dòng, dấu chấm phẩy cuối ngăn không in kết quả của (x, A, Q và R).

Các cột của ma trận Q về cơ bản là 4 đa thức Legendre đầu tiên của Hình 2.5. Chúng khác nhau không đáng kể bởi vì tích trong liên tục trong đoạn [-1,1] mà nó xác định các đa thức Legendre đã được thay thế bởi mô hình rời rạc. Chúng cũng khác nhau trong trực chuẩn hóa vì một đa thức Legendre thỏa mãn $P_k(1) = 1$. Ta có thể cố định điều này bằng việc chia mỗi cột của Q với phần tử cuối cùng của nó. Các dòng theo sau của Matlab làm điều này bằng phép nhân phải với ma trận đường chéo 4×4.

`scale = Q(257,:);`	Chọn cột cuối cùng của Q.
`Q = Q*diag(1 ./scale);`	Thay đổi tỷ lệ các cột bằng các số này.
`plot(Q);`	Vẽ các cột thay đổi tỷ lệ của Q.

Kết quả tính toán của chúng ta là một đồ thị mà nó trông giống như Hình 2.5 (không cho thấy ở đây). Trong Fortran hay C, điều này sẽ phải lấy nhiều dòng code chứa nhiều vòng lặp và các vòng lặp lồng nhau. Trong 6 dòng của Matlab, không có vòng lặp đơn xuất hiện rõ ràng, mặc dù ít nhất một vòng lặp là rõ ràng trong mỗi dòng.

2.4.3 Thực nghiệm 2: Gram - Schmidt cổ điển với Gram - Schmidt được sửa đổi

Ví dụ thứ hai nghiên cứu sự khác nhau trong tính ổn định số giữa thuật toán Gram -Schmidt cổ điển và được sửa đổi.

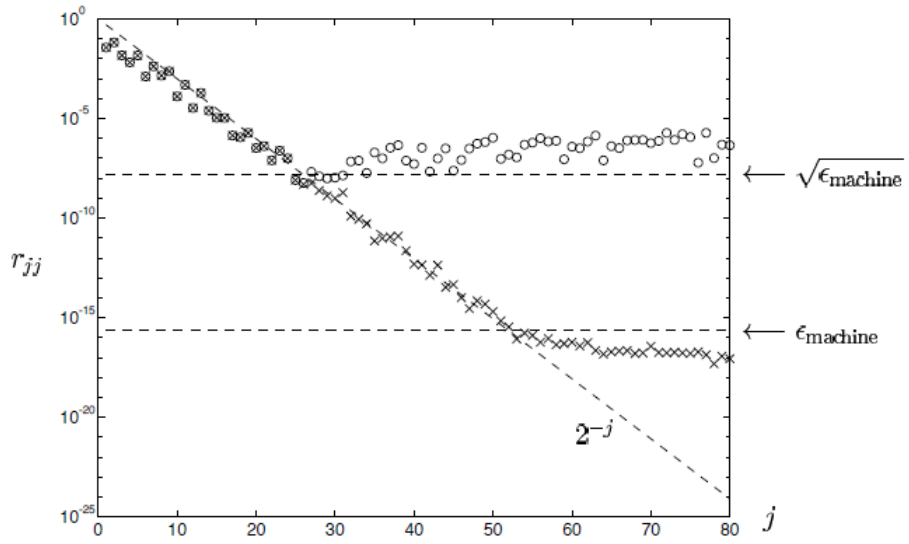
Đầu tiên, ta xây dựng một ma trận A với các vector suy biến ngẫu nhiên và các giá trị suy biến thay đổi thưa thớt được biết giữa 2 thừa số 2^{-1} và 2^{-80}.

`[U,X] = qr(randn(80));`	U là ma trận trực giao ngẫu nhiên.
`[V,X] = qr(randn(80));`	V là ma trận trực giao ngẫu nhiên.
`S = diag(2. ^(-1:-1:-80));`	S là ma trận đường chéo với các phần tử phân loại theo hàm mũ.
`A = U*S*V;`	A là ma trận với các phần tử là các giá trị suy biến.

Bây giờ ta sử dụng Thuật toán 2.1 và 2.2 để tính các phân tích QR của A. Trong đoạn code theo sau, *clgs* và *mgs* là các hàm thực thi trong Matlab của Thuật toán 2.1 và 2.2.

`[QC,RC] = clgs(A);`	Tính phân tích $Q^{(c)}R^{(c)}$ bằng Gram - Schmidt cổ điển.
`[QM, RM] = mgs(A);`	Tính phân tích $Q^{(m)}R^{(m)}$ bằng Gram - Schmidt được sửa đổi.

Cuối cùng, ta vẽ đồ thị các phần tử đường chéo r_{jj} được đưa ra bởi cả hai tính toán. Vì $r_{jj} = \|P_j a_j\|$, điều này cho chúng ta ảnh của kích thước của phép chiếu tại mỗi bước. Các kết quả được cho thấy trong Hình 2.6. Việc đầu tiên chú ý trong hình là sự giảm đều đều của r_{jj} với j, gần khớp với dòng 2^{-j}. Hiển nhiên r_{jj} không chính xác bằng giá trị suy biến thứ j của



Hình 2.6: Tính r_{jj} đối với j cho phân tích QR của một ma trận với các giá trị suy biến theo hàm mũ

A, nhưng nó là một xấp xỉ tốt phù hợp. Hiện tượng này có thể được giải thích như sau. SVD của A có thể được viết trong dạng (1.5.8) như

$$A = 2^{-1}u_1v_1^* + 2^{-2}u_2v_2^* + 2^{-3}u_3v_3^* + \ldots + 2^{-80}u_{80}v_{80}^*,$$

với $\{u_j\}$ và $\{v_j\}$ tương ứng là các vector suy biến trái và phải của A. Đặc biệt, cột thứ j của A có dạng

$$a_j = 2^{-1}\overline{v_{j1}}u_1 + 2^{-2}\overline{v_{j2}}u_2 + 2^{-3}\overline{v_{j3}}u_3 + \ldots + 2^{-80}\overline{v_{j,80}}u_{80}.$$

Vì các vector suy biến là ngẫu nhiên, nên ta có thể mong đợi rằng các số $\overline{v_{ji}}$ là tất cả độ dài tương tự, trong bậc $80^{-1/2} \approx 0.1$. Khi ta lấy phân tích QR, nó là hiển nhiên mà vector đầu tiên q_1 là xấp xỉ bằng u_1, với r_{11} bậc $2^{-1} \times 80^{-1/2}$. Trực giao hóa tại bước tiếp theo sẽ cho một vector q_2 xấp xỉ bằng u_2, với r_{22} trong bậc $2^{-2} \times 80^{-1/2}$, ...

Việc tiếp theo chú ý trong Hình 2.6 là r_{jj} không liên tục với $j = 80$. Đây là một chuỗi của việc làm tròn sai số trong máy tính. Với thuật toán Gram - Schmidt cổ điển, các sai số này không bao giờ nhỏ hơn khoảng 10^{-8}. Với thuật toán Gram - Schmidt được sửa đổi, chúng rút lại 8 bậc của độ dài, giảm xuống còn 10^{-16}, mức của *machine epsilon* cho máy tính sử dụng trong tính toán này. Machine epsilon được định nghĩa ở những mục sau.

Rõ ràng, một vài thuật toán là ổn định nhiều hơn các thuật toán khác. Quá trình Gram - Schmidt cổ điển là một những thuật toán không ổn định. Do đó nó hiếm khi được sử dụng, ngoại trừ thỉnh thoảng trong các máy tính song song ở chổ có ảnh hưởng lớn đến sự bất lợi của tính không ổn định.

2.4.4 Thực nghiệm 3: Sự hao hụt số của tính trực giao

Xét trường hợp của một ma trận

$$A = \begin{bmatrix} 0.70000 & 0.70711 \\ 0.70001 & 0.70711 \end{bmatrix} \tag{2.4.1}$$

trong một máy tính mà nó làm tròn các kết quả được tính toán tới 5 chữ số. Thuật toán Gram - Schmidt cổ điển và được sửa đổi là đồng nhất trong trường hợp 2×2. Tại bước $j = 1$, cột

đầu tiên được trực chuẩn, cho

$$r_{11} = 0.98996,\ q_1 = a_1/r_{11} = \begin{bmatrix} 0.70000/0.98996 \\ 0.70001/0.98996 \end{bmatrix} = \begin{bmatrix} 0.70710 \\ 0.70711 \end{bmatrix}$$

trong số học 5 chữ số. Tại bước $j = 2$, thành phần của a_2 trong phương của q_1 được tính và được trừ bên ngoài:

$$r_{12} = q_1^* a_2 = 0.70710 \times 0.70711 + 0.70711 \times 0.70711 = 1.0000,$$
$$v_2 = a_2 - r_{12} q_1 = \begin{bmatrix} 0.70711 \\ 0.70711 \end{bmatrix} - \begin{bmatrix} 0.70710 \\ 0.70711 \end{bmatrix} = \begin{bmatrix} 0.00001 \\ 0.00000 \end{bmatrix},$$

Tính v_2 trội hơn bởi các sai số. Q được tính cuối cùng là

$$Q = \begin{bmatrix} 0.70710 & 1.0000 \\ 0.70711 & 0.0000 \end{bmatrix},$$

Trong một máy tính với độ chính xác 16 chữ số, ta vẫn mất khoảng 5 chữ số của tính trực giao nếu ta áp dụng Gram - Schmidt được sửa đổi với ma trận 2.4.1. Hàm "eye" khởi tạo ma trận đơn vị của số chiều đươc chỉ định.

	Định nghĩa A.
`A = [.70000 .70711;.70001 .70711];`	Tính thừa số Q bằng Householder.
`[Q,R] = qr(A);`	Kiểm tra tính trực giao của Q.
`norm(Q'*Q -eye(2))`	Tính thừa số Q bằng G-S được sửa
`[Q,R] = mgs(A);`	đổi.
`norm(Q'*Q - eye(2))`	Kiểm tra tính trực giao của Q.

Các dòng không có dấu chấm phẩy đưa ra kết quả sau:

`ans = 2.3382e-16,  ans = 2.3014e - 11.`

2.5 Tam giác hóa Householder

Thuật toán Householder là một quá trình của "tam giác hóa trực giao", làm một ma trận tam giác bằng một chuỗi các phép toán ma trận Unita.

2.5.1 Householder và Gram - Schmidt

Như ta thấy trong (2.3), bước lặp Gram - Schmidt áp dụng liên tiếp các ma trận tam giác cơ bản R_k vào bên phải của A, để được ma trận kết quả

$$A \underbrace{R_1 R_2 \dots R_n}_{\widehat{R}^{-1}} = \widehat{Q}$$

có các cột trực giao. Tích $\widehat{R} = R_n^{-1} \dots R_2^{-1} R_1^{-1}$ cũng là ma trận tam giác trên, và do đó $A = \widehat{Q}\widehat{R}$ là một phân tích QR được sửa đổi của A.

Ngược lại, phương pháp Householder áp dụng liên tiếp các ma trận Unita Q_k vào bên trái của A nên ma trận kết quả

$$\underbrace{Q_n \dots Q_2 Q_1}_{\widehat{Q}^*} A = R$$

là ma trận tam giác trên. Tích $Q = Q_1^* Q_2^* \dots Q_n^*$ cũng là ma trận Unita, và do đó $A = QR$ là một phân tích QR đầy đủ của A.

Do đó, hai phương pháp có thể được tóm tắt như sau:

Gram - Schmidt: trực giao hóa tam giác,
Householder: tam giác hóa trực giao.

2.5.2 Tam giác hóa bằng việc đưa vào các số 0

Phương pháp Householder được đưa ra đầu tiên bởi Alston Householder trong năm 1958. Đây là một cách khéo léo của việc thiết kế các ma trận Unita Q_k sao cho $Q_n \dots Q_2 Q_1 A$ là ma trận tam giác trên.

Ma trận Q_k được chọn để đưa ra các số 0 bên dưới đường chéo trong cột thứ k trong khi bảo toàn các số 0 được đưa ra trước đó. Ví dụ, trong trường hợp 5×3, 3 phép toán Q_k được áp dụng. Trong các ma trận này, ký hiệu $\times$ biểu diễn một phần tử khác 0, và kiểu chữ đậm cho biết một phần tử vừa được thay đổi. Các phần tử để trống là 0.

$$\underset{A}{\begin{bmatrix} \times & \times & \times \\ \times & \times & \times \\ \times & \times & \times \\ \times & \times & \times \\ \times & \times & \times \end{bmatrix}} \xrightarrow{Q_1} \underset{Q_1 A}{\begin{bmatrix} \mathbf{\times} & \mathbf{\times} & \mathbf{\times} \\ \mathbf{0} & \times & \times \\ \mathbf{0} & \times & \times \\ \mathbf{0} & \times & \times \\ \mathbf{0} & \times & \times \end{bmatrix}} \xrightarrow{Q_2} \underset{Q_2 Q_1 A}{\begin{bmatrix} \times & \times & \times \\ & \mathbf{\times} & \mathbf{\times} \\ & \mathbf{0} & \times \\ & \mathbf{0} & \times \\ & \mathbf{0} & \times \end{bmatrix}} \xrightarrow{Q_3} \underset{Q_3 Q_2 Q_1 A}{\begin{bmatrix} \times & \times & \times \\ & \times & \times \\ & & \mathbf{\times} \\ & & \mathbf{0} \\ & & \mathbf{0} \end{bmatrix}} \tag{2.5.1}$$

Đầu tiên, Q_1 tính toán trong các dòng 1, ..., 5, đưa ra các số 0 nằm ở các vị trí (2,1), (3,1), (4,1) và (5,1). Tiếp theo, Q_2 tính toán trong các dòng 2, ..., 5, đưa ra các số 0 nằm ở các vị trí (3,2), (4,2) và (5,2) nhưng không triệt tiêu các số 0 được đưa ra bởi Q_1. Cuối cùng, Q_k tính toán trong các dòng 3, ..., 5, đưa ra các số 0 ở các vị trí (4,3), (5,3) không triệt tiêu bất kì số 0 nào được đưa ra trước đó.

Tổng quát, Q_k tính toán trong các dòng $k, \dots, m$. Bắt đầu của bước k, có 1 khối các số 0 trong $k-1$ cột đầu tiên của các dòng này. Áp dụng Q_k hình thành các tổ hợp tuyến tính của các dòng này, và các tổ hợp tuyến tính của các phần tử 0 còn lại là 0. Sau n bước, tất cả các phần tử nằm bên dưới đường chéo đã được khử và $Q_n \dots Q_2 Q_1 A$ là ma trận tam giác trên.

2.5.3 Phản xạ Householder

Mỗi Q_k được chọn để là một ma trận Unita dạng

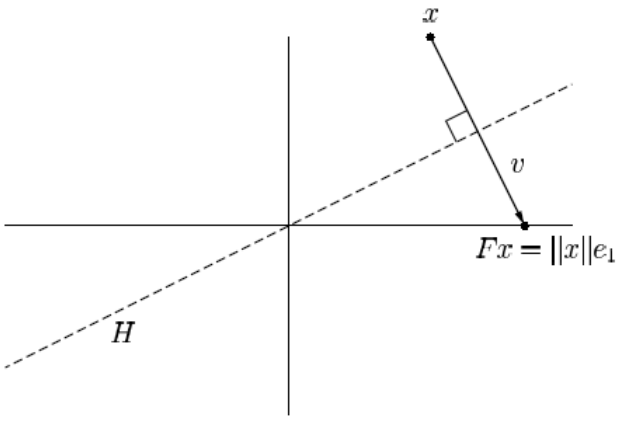
$$Q_k = \begin{bmatrix} I & 0 \\ 0 & F \end{bmatrix}, \tag{2.5.2}$$

với I là ma trận đơn vị $(k-1) \times (k-1)$ và F là ma trận Unita $(m-k+1) \times (m-k+1)$. Phép nhân với F phải đưa vào các số 0 vào cột thứ k. Thuật toán Householder chọn F là một trận đặc biệt được gọi là *phản xạ Householder*.

Giả sử, bắt đầu bước k, các phần tử $k, \dots, m$ của cột thứ k được cho bởi vector $x \in \mathbb{C}^{m-k+1}$. Để đưa ra chính xác các số 0 vào cột thứ k, phản xạ Householder F nên tác động ánh xạ

$$x = \begin{bmatrix} \times \\ \times \\ \times \\ \vdots \\ \times \end{bmatrix} \xrightarrow{F} Fx = \begin{bmatrix} \|x\| \\ 0 \\ 0 \\ \vdots \\ 0 \end{bmatrix} = \|x\| e_1. \tag{2.5.3}$$

Ý tưởng cho việc thực hiện này được cho biết trong Hình 2.7. Phản xạ F sẽ phản xạ không gian $\mathbb{C}^{m-k+1}$ qua một siêu phẳng H trực giao với $v = \|x\| e_1 - x$. Một *siêu phẳng* là sự tổng quát hóa số chiều cao hơn của mặt phẳng 2 chiều trong không gian 3 chiều - một không gian con 3 chiều của một không gian 4 chiều, một không gian con 4 chiều của một không gian 5



Hình 2.7: Phản xạ Householder

chiều, ... Tổng quát, một siêu phẳng có thể được đặc trưng như tập hợp các điểm trực giao với một vector khác 0 được cố định. Trong Hình 2.7, vector đó là $v = \|x\|e_1 - x$, và đường nét gạch như là một miêu tả của H được xem là "bờ".

Khi phản xạ được áp dụng, mọi điểm trong bờ của siêu phẳng H được ánh xạ thành ảnh phản xạ của nó trong bờ khác. Đặc biệt, x được ánh xạ thành $\|x\|e_1$. Trong (2.1.11), với $y \in \mathbb{C}^m$ bất kỳ, vector

$$Py = \left(I - \frac{vv^*}{v^*v}\right) y = y - v\left(\frac{v^*y}{v^*v}\right) \tag{2.5.4}$$

là một phép chiếu trực giao của y vào không gian H. Để lấy phản xạ y qua H, ta phải lấy 2 lần hơn là trong cùng một phương. Do đó phép chiếu Fy sẽ là

$$Fy = \left(I - 2\frac{vv^*}{v^*v}\right) y = y - 2v\left(\frac{v^*y}{v^*v}\right).$$

Do đó ma trận F là

$$F = I - 2\frac{vv^*}{v^*v}. \tag{2.5.5}$$

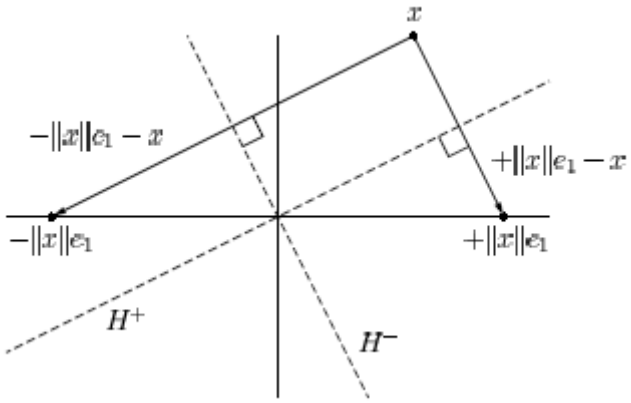
Chú ý rằng phép chiếu P (hạng $m-1$)và phản xạ F (hạng đầy đủ, Unita) chỉ khác nhau trong biểu biễn một thừa số của 2.

2.5.4 Tốt hơn của 2 phản xạ

Trong (2.5.3) và trong Hình 2.7 ta có các vấn đề, có nhiều đối xứng Householder mà chúng sẽ đưa ra các số 0 cần thiết. Vector x có thể được lấy phản xạ thành $z\|x\|e_1$, với z là vô hướng bất kỳ thỏa $|z| = 1$. Trong trường hợp số phức, có một vòng tròn của các phản xạ có thể thực hiện được, và ngay trong trường hợp số thực, có 2 lựa chọn, được miêu tả bởi các phản xạ qua hai siêu phẳng khác nhau, H^+ và H^-, như được miêu tả trong Hình 2.8.

Theo toán học, một trong hai sự lựa chọn dấu là thỏa mãn. Tuy nhiên, đây là trường hợp mà mục tiêu của tính ổn định số đưa ra một lựa chọn sẽ được lấy hơn một cái khác. Cho tính ổn định số, lấy đối xứng x thành vector $z\|x\|e_1$ mà không quá gần x. Để đạt được điều này, ta có thể chọn $z = -sign(x_1)$, với x_1 là thành phần đầu tiên của x, để mà vector đối xứng trở thành $v = -sign(x_1)\|x\|e_1 - x$, hay theo các thừa số -1

$$v = sign(x_1)\|x\|e_1 + x. \tag{2.5.6}$$



Hình 2.8: Hai phản xạ có thể thực hiện được

Để làm điều này đầy đủ, ta có thể đặt vào tùy ý quy ước $sign(x_1) = 1$ nếu $x_1 = 0$.

Không quá khó để thấy vì sao sự lựa chọn dấu làm một sự khác nhau cho tính ổn định. Giả sử trong Hình 2.8, góc giữa H^+ và trục e_1 là rất nhỏ. Khi đó vector $v = \|x\|e_1 - x$ là nhỏ hơn x hoặc $\|x\|e_1$ nhiều. Do đó tính toán v biểu diễn một phép trừ các lượng gần và sẽ hướng tới cho phép các sai số triệt tiêu nhau. Nếu ta lựa chọn dấu như trong 2.5.6, ta đảm bảo rằng $\|v\|$ là không bao giờ nhỏ hơn $\|x\|$.

2.5.5 Thuật toán

Bây giờ ta đưa ra thuật toán Householder đầy đủ. Để làm điều này, nó sẽ hữu ích để khởi tạo một ký hiệu mới (kiểu Matlab). Nếu A là một ma trận, ta xác định $a_{i:i',j:j'}$ là ma trận con $(i' - i + 1) \times (j' - j + 1)$ của A với góc trái trên là a_{ij} và góc phải dưới là $a_{i',j'}$. Trong trường hợp đặc biệt mà ma trận con giảm xuống thành một vector con của một dòng hay cột, ta viết tương ứng là $A_{i,j:j'}$ hoặc $A_{i:i',j}$.

Thuật toán theo sau tính thừa số R của phân tích QR của một ma trận A có $m \times n$ với $m \geqslant n$. Theo cách này, n vector phản xạ $v_1, \ldots, v_n$ được lưu trữ sử dụng sau.

Thuật toán 2.3 Phân tích QR Householder

1: **for** $k = 1$ to n **do**
2: $\quad x = A_{k:m,k}$
3: $\quad v_k = sign(x_1)\|x\|_2 e_1 + x$
4: $\quad v_k = v_k / \|v_k\|_2$
5: $\quad A_{k:m,k:n} = A_{k:m,k:n} - 2v_k(v_k^* A_{k:m,k:n})$
6: **end for**

2.5.6 Việc áp dụng hoặc tạo thành Q

Theo Thuật toán 2.3, A đã được giảm xuống thành dạng tam giác trên. Đó là ma trận R trong phân tích QR $A = QR$. Tuy nhiên, ma trận Unita Q đã không được xây dựng, không có ma trận con n cột của nó $\widehat{Q}$ tương ứng tới phân tích QR được giảm. Việc xây dựng Q hoặc $\widehat{Q}$ đưa thêm vào, và trong nhiều ứng dụng, ta có thể tránh điều này bằng cách làm một cách trực tiếp với công thức

$$Q^* = Q_n \ldots Q_2 Q_1 \tag{2.5.7}$$

hoặc liên hợp của nó

$$Q = Q_1 Q_2 \dots Q_n. \tag{2.5.8}$$

(Không có dấu * nào đã được quên ở đây; nhắc lại rằng mỗi Q_j là Hermit.)

Ví dụ, trong Mục 2.2 ta thấy rằng một hệ thống vuông của các phương trình $Ax = b$ có thể được giải thông qua phân tích QR của A. Q được sử dụng trong tính toán của tích Q^*b. Theo (2.5.7) ta có thể tính Q^*b bằng một chuỗi n phép toán được áp dụng cho b, các phép toán tương tự nhau được áp dụng cho A để làm nó thành ma trận tam giác. Thuật toán như sau.

Tương tự, tính toán tích Qx có thể được thực hiện bằng quá trình tương tự được thực thi

Thuật toán 2.4 Tính toán ngầm của tích Q^*b

1: **for** $k = 1$ to n **do**
2: $\quad b_{k:m} = b_{k:m} - 2v_k(v_k^* b_{k:m})$
3: **end for**

trong thứ tự ngược lại.

Ta có thể xây dựng QI thông qua Thuật toán 2.5 bằng việc tính các cột $Qe_1, Qe_2, \dots, Qe_m$

Thuật toán 2.5 Tính toán ngầm của tích Q^*b

1: **for** $k = n$ down 1 **do**
2: $\quad x_{k:m} = x_{k:m} - 2v_k(v_k^* x_{k:m})$
3: **end for**

của nó. Ngoài ra, ta có thể xây dựng Q^*I thông qua Thuật toán 2.4 và do đó kết quả là liên hợp. Một biến thể của ý tưởng này là xây dựng IQ bằng việc tính các dòng $e_1^*Q, e_2^*Q, \dots, e_m^*Q$ như được đề nghị bởi (2.5.8). Ý tưởng tốt nhất là ý tưởng đầu tiên, dựa vào Thuật toán 2.5. Vì nó bắt đầu với các phép toán bao gồm Q_n, Q_{n-1}, và chỉ sửa đổi một phần nhỏ vector được áp dụng.

Nếu $\widehat{Q}$ cần thiết hơn là Q thì nó đủ để tính toán các cột $Qe_1, Qe_2, \dots, Qe_n$.

2.5.7 Đếm số phép toán

Thuật toán 2.3 được chi phối bởi vòng lặp ở trong cùng,

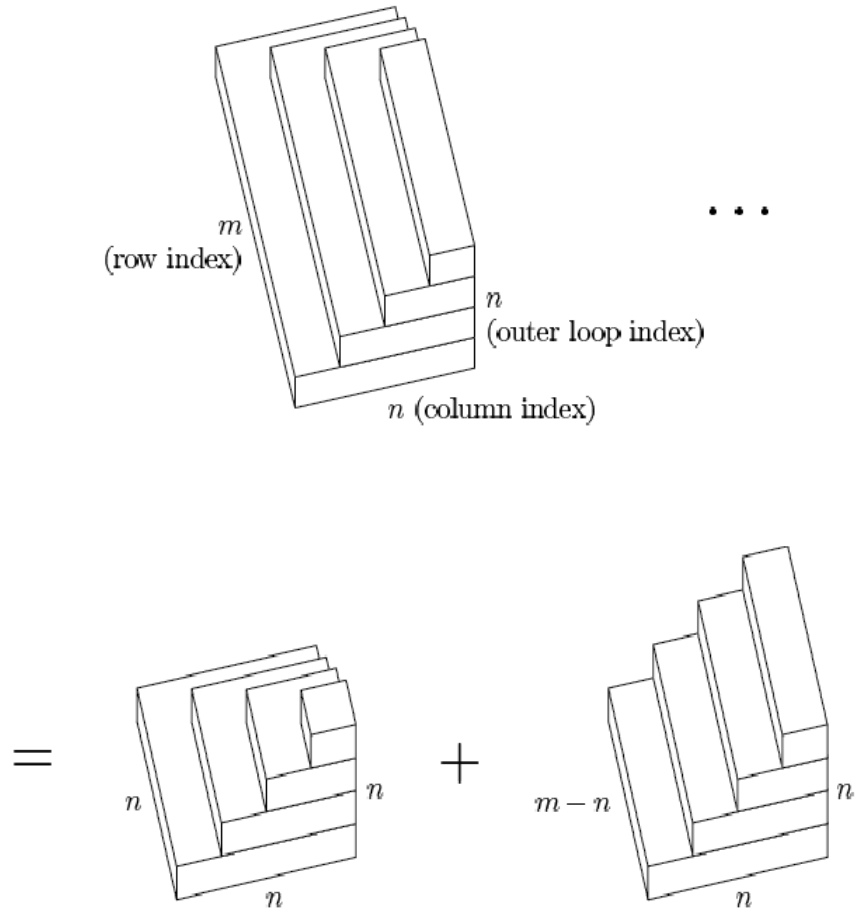
$$A_{k:m,j} - 2v_k(v_k^* A_{k:m,j}). \tag{2.5.9}$$

Nếu chiều dài vector là $l = m - k + 1$ thì tính toán này cần $4l - 1 \sim 4l$ phép toán vô hướng: l cho phép trừ, l cho phép nhân vô hướng, và $2l - 1$ cho tích vô hướng. Có ~ 4 phép toán dấu chấm động cho mỗi phần tử được thực hiện.

Ta có thể lấy tổng 4 phép toán dấu chấm động này trên một phần tử bằng lý do hình học như trong Mục 2.3. Mỗi bước liên tiếp của vòng lặp bên ngoài tính toán trong vài dòng bởi vì trong bước k, các dòng $1, \dots, k-1$ không được thay đổi. Hơn nữa, mỗi bước tính toán trong một vài cột, bởi vì các cột của $1, \dots, k-1$ của các dòng được tính toán là 0 và được bỏ qua. Do đó hoàn thành việc làm bằng một bước bên ngoài có thể được biểu diễn bằng một lớp của hình ba chiều theo sau:

Tổng số phép toán tương ứng với 4 lần thể tích của hình ba chiều. Để xác định thể tích bằng hình ảnh ta có thể chia hình ba chiều thành 2 mảnh:

Hình ba chiều bên trái có hình dạng của kim tự tháp và hội tụ tới một hình chóp khi $n \to \infty$, với thể tích là $\frac{1}{3}n^3$. Hình ba chiều bên phải có hình dạng của một cầu thang và hội tụ tới hình



lăng trụ khi $m, n \to \infty$, với thể tích là $\frac{1}{2}(m-n)n^2$. Kết hợp lại, thể tích $\sim \frac{1}{2}mn^2 - \frac{1}{6}n^3$. Việc nhân 4 phép toán dấu chấm động trên đơn vị thể tích, ta thấy

$$\text{Trực giao hóa Householder: } \sim 2mn^2 - \frac{2}{3}n^3 \text{ phép toán dấu chấm động.} \tag{2.5.10}$$

2.6 Các bài toán bình phương nhỏ nhất

2.6.1 Bài toán

Xét một hệ thống tuyến tính các phương trình có $m > n$ phương trình nhưng n không được biết. Tìm một vector $x \in \mathbb{C}^n$ sao cho $Ax = b$, với $A \in \mathbb{C}^{m\times n}$ và $b \in \mathbb{C}^m$. Tổng quát, một bài toán như vậy không có lời giải. Một vector x phù hợp chỉ tồn tại nếu b nằm trong $range(A)$, và vì b là một vector m chiều, trong khi $range(A)$ có nhiều nhất n chiều, điều này chỉ đúng cho các lựa chọn ngoại lệ của b. Ta nói rằng một hệ thống hình chữ nhật của các phương trình với $m > n$ là *được xác định hầu hết*. Vector được biết như là *thặng dư*,

$$r = b - Ax \in \mathbb{C}^m, \tag{2.6.1}$$

có thể được làm khá nhỏ bằng một lựa chọn phù hợp của x, nhưng tổng quát nó không thể được làm bằng 0.

Vì thặng dư r không thể bằng 0, nên thay vì làm nó nhỏ như có thể thực hiện được. Việc đo sự nhỏ nhất của r dẫn đến việc chọn một chuẩn. Nếu ta chọn chuẩn 2, bài toán đưa về dạng

như sau:

$$\begin{aligned} &\text{Cho } A \in \mathbb{C}^{m\times n}, m \geqslant n, b \in \mathbb{C}^m, \\ &\text{tìm } x \in \mathbb{C}^m \text{ sao cho } \|b - Ax\|_2 \text{ được cực tiểu hóa.} \end{aligned} \tag{2.6.2}$$

Đó là công thức của *bài toán bình phương nhỏ nhất* tổng quát (tuyến tính). Việc chọn chuẩn 2 có thể được bảo vệ bởi các đối số hình học và thống kê khác nhau để đưa ra các thuật toán đơn giản- cuối cùng bởi vì đạo hàm của một hàm bậc hai là tuyến tính, mà nó được đặt là 0 cho sự cực tiểu hóa.

Chuẩn 2 tương ứng với khoảng cách Euclidean, nên có một sự giải thích hình học đơn giản của (2.6.2). Ta tìm một vector $x \in \mathbb{C}^n$ sao cho vector $Ax \in \mathbb{C}^m$ là một điểm gần nhất tới b trong $range(A)$.

2.6.2 Ví dụ: việc điều chỉnh dữ liệu đa thức

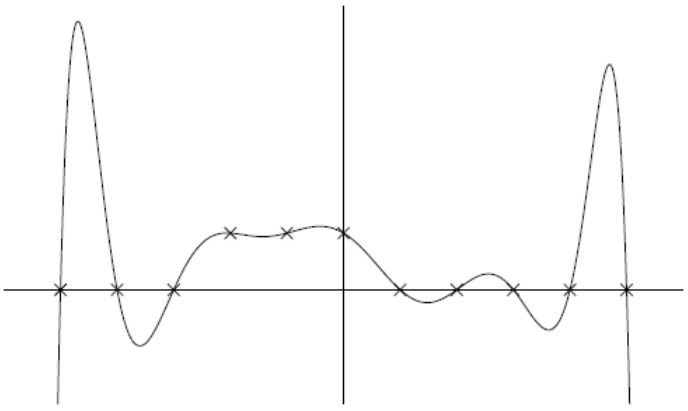
Ví dụ 2.6.1. Nội suy đa thức. Giả sử ta được cho m điểm phân biệt $x_1, \ldots, x_m \in \mathbb{C}$ và dữ liệu $y_1, \ldots, y_m \in \mathbb{C}$ tại các điểm này. Khi đó tồn tại duy nhất một *nội suy đa thức* bậc lớn nhất là $m-1$ tới các dữ liệu này trong các điểm này

$$p(x) = c_0 + c_1 x + \ldots + c_{m-1}x^{m-1}, \tag{2.6.3}$$

với tính chất là tại mỗi điểm $x_i, p(x_i) = y_i$. Quan hệ giữa $\{x_i\}, \{y_i\}$ với các hệ số $\{c_i\}$ có thể được biểu diễn bởi hệ thống Vandermonde vuông được thấy trong Ví dụ 1.2.3:

$$\begin{bmatrix} 1 & x_1 & x_1^2 & \ldots & x_1^{m-1} \\ 1 & x_2 & x_2^2 & \ldots & x_2^{m-1} \\ 1 & x_3 & x_3^2 & \ldots & x_3^{m-1} \\ \vdots & \vdots & \vdots & & \vdots \\ 1 & x_m & x_m^2 & \ldots & x_m^{m-1} \end{bmatrix} \begin{bmatrix} c_0 \\ c_1 \\ c_2 \\ \vdots \\ c_{m-1} \end{bmatrix} = \begin{bmatrix} y_1 \\ y_2 \\ y_3 \\ \vdots \\ y_m \end{bmatrix} \tag{2.6.4}$$

Để xác định các hệ số $\{c_i\}$ cho một tập dữ liệu được cho, ta có thể giải hệ thống các phương trình này, mà nó được đảm bảo là không suy biến miễn là các điểm $\{x_i\}$ là phân biệt.



Hình 2.9: Nội suy đa thức bậc 10 của 11 điểm dữ liệu

Hình 2.6.2 đưa ra một ví dụ của quá trình nội suy đa thức. Ta có 11 điểm dữ liệu trong dạng sóng vuông rời rạc được biểu diễn bởi các dấu chữ thập, và đường cong $p(x)$ đi qua chúng.

Tuy nhiên, sự điều chỉnh cho vừa là không chút nào dễ chịu. Gần cuối khoảng, $p(x)$ đưa ra sự giao động lớn mà chúng rõ ràng là một thành phần lạ của quá trình nội suy, không phản ánh dữ liệu hợp lý.

Xử lý không thỏa mãn này là đặc trưng của nội suy đa thức. Các điều chỉnh cho vừa kết quả của nó thường là xấu, và chúng đề cập đến trường hợp xấu nhất hơn là trường hợp tốt hơn nếu dữ liệu nhiều hơn được sử dụng. Ngay cả khi sự điều chỉnh cho vừa là tốt, quá trình nội suy có thể là điều kiện xấu, nghĩa là, bị ảnh hưởng bởi các nhiễu của dữ liệu. Để tránh các bài toán này, ta có thể sử dụng một tập không đồng nhất các điểm nội suy như là các điểm Chebyshev trong khoảng [-1,1]. Tuy nhiên, trong nhiều ứng dụng, nó sẽ không thường xuyên có thể thực hiện được để chọn các điểm nội suy.

Ví dụ 2.6.2. Điều chỉnh bình phương nhỏ nhất đa thức Cho $x_1, \ldots, x_m$ và $y_1, \ldots, y_m$, xét một đa thức bậc $n-1$

$$p(x) = c_0 + c_1 x + \ldots + c_{n-1} x^{n-1} \tag{2.6.5}$$

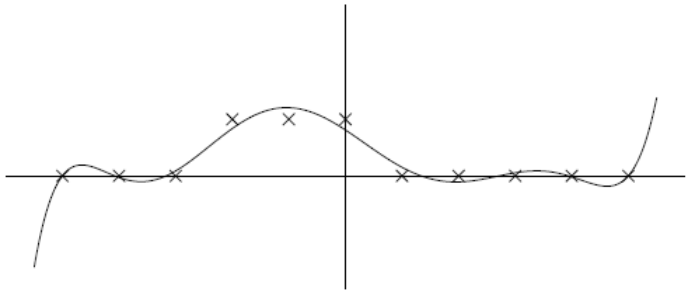
với $n < m$. Một đa thức như vậy là điều chỉnh bình phương nhỏ nhất tới dữ liệu nếu nó cực tiểu hóa tổng bình phương của độ lệch từ dữ liệu,

$$\sum_{i=1}^{m} |p(x_i) - y_i|^2. \tag{2.6.6}$$

Tổng bình phương này là tương đương với bình phương của chuẩn thặng dư, $\|r\|_2^2$ cho hệ thống Vandermonde hình chữ nhật

$$\begin{bmatrix} 1 & x_1 & & x_1^{n-1} \\ 1 & x_2 & \ldots & x_2^{n-1} \\ 1 & x_3 & \ldots & x_3^{n-1} \\ \vdots & & & \vdots \\ 1 & x_m & \ldots & x_m^{n-1} \end{bmatrix} \begin{bmatrix} c_0 \\ c_1 \\ \vdots \\ c_{n-1} \end{bmatrix} \approx \begin{bmatrix} y_1 \\ y_2 \\ y_3 \\ \vdots \\ y_m \end{bmatrix} \tag{2.6.7}$$

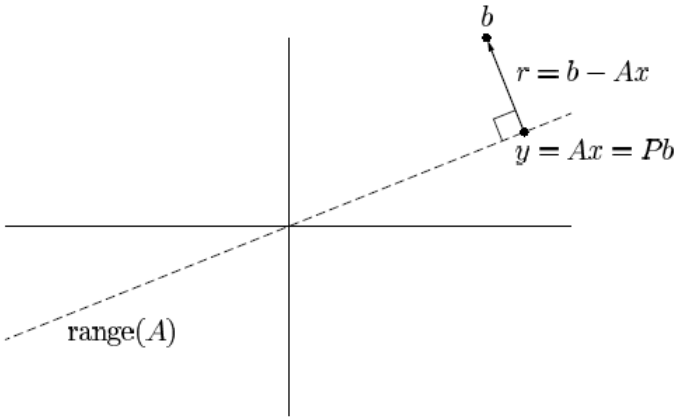
Hình minh họa nếu ta điều chỉnh cho vừa 11 điểm dữ liệu giống nhau từ ví dụ cuối cùng với đa thức bậc 7. Một đa thức mới không nội suy dữ liệu, nhưng nó thu nạp tất cả tập tính của chúng nhiều hơn là đa thức trong Ví dụ 2.6.2. Mặc dù ta không thể thấy điều này trong hình, nó cũng chỉ bị ảnh hưởng ít hơn tới các nhiễu loạn.



Hình 2.10: Đa thức bậc 7 điều chỉnh bình phương nhỏ nhất của 11 điểm dữ liệu giống nhau

2.6.3 Phép chiếu trực giao và các phương trình trực chuẩn tắc

Ý tưởng được minh họa trong Hình 2.11. Mục tiêu của chúng ta là tìm một điểm Ax gần với b nhất trong $range(A)$, để chuẩn của thặng dư $r = b - Ax$ được cực tiểu hóa. Rõ ràng điều



Hình 2.11: Công thức của bài toán bình phương nhỏ nhất liên quan tới phép chiếu trực giao

này sẽ đưa ra $Ax = Pb$, trong đó $P \in \mathbb{C}^{m\times m}$ là một phép chiếu trực giao mà nó ánh xạ $\mathbb{C}^m$ vào $range(A)$. Mặt khác, *thặng dư* $r = b - Ax$ *phải trực giao với* $range(A)$.

Định lý 2.6.1 *Cho* $A \in \mathbb{C}^{m\times n}(m \geqslant n)$ *và* $b \in \mathbb{C}^m$ *được cho. Một vector* $x \in \mathbb{C}^m$ *cực tiểu hóa chuẩn thặng dư* $\|r\|_2 = \|b - Ax\|_2$*, do đó việc giải bài toán bình phương nhỏ nhất (2.6.2) nếu và chỉ nếu* $r \perp range(A)$*, nghĩa là,*

$$A^*r = 0, \tag{2.6.8}$$

tương đương

$$A^*Ax = A^*b, \tag{2.6.9}$$

hoặc tương đương

$$Pb = Ax, \tag{2.6.10}$$

với $P \in \mathbb{C}^{m\times m}$ *là phép chiếu trực giao vào* $range(A)$*. Hệ thống* $n \times n$ *phương trình (2.6.9) (được biết như là các phương trình chuẩn tắc) là không suy biến nếu và chỉ nếu A có hạng đầy đủ. Do đó lời giải x là duy nhất nếu và chỉ nếu A có hạng đầy đủ.*

Chứng minh Sự tương đương của (2.6.8) và (2.6.10) theo sau từ các tính chất của các phép chiếu trực giao được thảo luận trong Mục 2.1, và sự tương đương của (2.6.8) và (2.6.9) theo sau từ định nghĩa của r. Để chứng minh $y = Pb$ là điểm duy nhất trong $range(A)$ mà nó cực tiểu hóa $\|b - y\|_2$, giả sử $z \neq y$ là một điểm khác trong $range(A)$. Vì $z - y$ trực giao với $b - y$, theo định lý Pythagorean $\|b-z\|_2^2 = \|b-y\|_2^2 + \|y-z\|_2^2 > \|b-y\|_2^2$. Cuối cùng, ta chú ý rằng nếu A^*A là suy biến, thì $A^*Ax = 0$ với x khác 0 bất kì, kéo theo $x^*A^*Ax = 0$. Do đó $Ax = 0$, mà nó kéo theo A có hạng không đầy đủ. Ngược lại, nếu A có hạng không đầy đủ, thì $Ax = 0$ với x khác 0, cũng kéo theo $A^*Ax = 0$, cũng kéo theo $A^*Ax = 0$, nên A^*A là suy biến. Do (2.6.9), đặc trưng của các ma trận không suy biến A^*A này đưa ra phát biểu về tính duy nhất của x.

2.6.4 Giả nghịch đảo

Nếu A có hạng đầy đủ thì lời giải x cho bài toán bình phương nhỏ nhất (2.6.2) là duy nhất và được cho bởi $x = (A^*A)^{-1}A^*b$. Ma trận $(A^*A)^{-1}A^*$ được biết như là *giả nghịch đảo* của A, được ký hiệu là A^+:

$$A^+ = (A^*A)^{-1}A^* \in \mathbb{C}^{n,m}. \tag{2.6.11}$$

Ma trận này ánh xạ các vector $b \in \mathbb{C}^m$ thành các vector $x \in \mathbb{C}^n$.

Bài toán bình phương nhỏ nhất tuyến tính hạng đầy đủ (2.6.2) tính một hoặc cả hai vector

$$x = A^+ b, \quad y = Pb, \tag{2.6.12}$$

với A^+ là giả nghịch đảo của A và P là phép chiếu trực giao vào $range(A)$.

2.6.5 Các phương trình chính tắc

Cách cổ điển để giải các bài toán bình phương nhỏ nhất là để giải các phương trình chuẩn tắc (2.6.9). Nếu A có hạng đầy đủ thì đây là hệ thống xác định dương Hermit và vuông của các phương trình n chiều. Phương pháp tiêu chuẩn của việc giải một hệ thống này là *phân tích Cholesky*, được thảo luận sau. Phương pháp này xây dựng một phân tích $A^*A = R^*R$, với R là ma trận tam giác trên, giảm (2.6.9) thành các phương trình

$$R^*Rx = A^*b. \tag{2.6.13}$$

Dưới đây là thuật toán.

Thuật toán 2.6 Bình phương tối tiểu qua các phương trình chính tắc

1: Thiết lặp ma trận A^*A và vector A^*b.
2: Tính phân tích Cholesky $A^*A = R^*R$.
3: Giải hệ thống tam giác dưới $R^*w = A^*b$ cho biến w.
4: Giải hệ thống tam giác trên $Rx = w$ cho biến x.

Các bước mà nó chi phối việc cho tính toán này là 2 bước đầu tiên. Bởi vì tính đối xứng, A^*A chỉ cần mn^2 phép toán dấu chấm động, phân nửa chi phí nếu A và A^* là các ma trận tùy ý có cùng số chiều. Phân tích Cholesky yêu cầu $n^3/3$ phép toán dấu chấm động. Kết hợp lại, việc giải bài toán bình phương nhỏ nhất bằng các phương trình chuẩn tắc bao gồm tổng số phép toán theo sau:

$$\text{Thuật toán 2.6: } \sim mn^2 + \frac{1}{3}n^3 \text{ phép toán dấu chấm động.} \tag{2.6.14}$$

2.6.6 Phân tích QR

Phương pháp "mô hình cổ điển" cho việc giải các bài toán bình phương nhỏ nhất phổ biến từ những năm 1960. Nó được dựa vào phân tích QR được sửa đổi. Theo trực giao hóa Gram - Schmidt , hoặc thường xuyên hơn là tam giác hóa Householder, xây dựng phân tích $A = \widehat{Q}\widehat{R}$. Khi đó phép chiếu trực giao P có thể được viết $P = \widehat{Q}\widehat{Q}^*$ (2.1.6), nên ta có

$$y = Pb = \widehat{Q}\widehat{Q}^*b. \tag{2.6.15}$$

Vì $y \in range(A)$ nên hệ thống $Ax = y$ có một lời giải chính xác. Kết hợp với phân tích QR và (2.6.15) cho

$$\widehat{Q}\widehat{R}x = \widehat{Q}\widehat{Q}^*b, \tag{2.6.16}$$

và nhân trái với $\widehat{Q}^*$

$$\widehat{R}x = \widehat{Q}^*b. \tag{2.6.17}$$

(Nhân với $\widehat{R}^{-1}$ cho công thức $A^+ = \widehat{R}^{-1}\widehat{Q}^*$ cho giả nghịch đảo.) Phương trình (2.6.17) là hệ thống tam giác trên, không suy biến nếu A có hạng đầy đủ, và nó sẵn sàng được giải bằng phép thế ngược.

Chú ý (2.6.17) cũng có thể được suy ra từ các phương trình chuẩn tắc. Nếu $A^*Ax = A^*b$, thì

Thuật toán 2.7 Bình phương tối tiểu thông qua phân tích QR

1: Tính phân tích QR được sửa đổi $A = \widehat{Q}\widehat{R}$.
2: Tính vector $\widehat{Q}^*b$.
3: Giải hệ thống tam giác trên $\widehat{R}x = \widehat{Q}^*b$ cho biến x.

$\widehat{R}^*\widehat{Q}^*\widehat{Q}\widehat{R}x = \widehat{R}^*\widehat{Q}^*b$, kéo theo $\widehat{R}x = \widehat{Q}^*b$.

Thuật toán 2.6 bị ảnh hưởng bởi chi phí của phân tích QR. Nếu các phản xạ Householder được sử dụng ở bước này, từ (2.5.10) ta có

$$\text{Thuật toán 3.1: } \sim 2mn^2 - \frac{2}{3}n^3 \text{ phép toán dấu chấm động.} \tag{2.6.18}$$

2.6.7 SVD

Phân tích giá trị suy biến được sửa đổi $A = \widehat{U}\widehat{\sum}V^*$ là một phương pháp khác cho việc giải bài toán bình phương nhỏ nhất. P được biểu diễn trong dạng $P = \widehat{U}\widehat{U}^*$, cho

$$y = Pb = \widehat{U}\widehat{U}^*b, \tag{2.6.19}$$

và tương tự (2.6.16) và (2.6.17) là

$$\widehat{U}\widehat{\sum}V^*x = \widehat{U}\widehat{U}^*b \tag{2.6.20}$$

và

$$\widehat{\sum}V^*x = \widehat{U}^*b. \tag{2.6.21}$$

(Nhân với $V\widehat{\sum}^{-1}$ cho $A^+ = V\widehat{\sum}^{-1}\widehat{U}^*$.)

Chú ý trong khi phân tích QR giảm bài toán bình phương nhỏ nhất thành một hệ thống tam

Thuật toán 2.8 Bình phương tối tiểu qua SVD

1: Tính SVD được sửa đổi $A = \widehat{U}\widehat{\sum}V^*$.
2: Tính vector $\widehat{U}^*b$.
3: Giải hệ thống đường chéo $\widehat{\sum}w = \widehat{U}^*b$ cho biến w.
4: Đặt $x = Vw$.

giác của các phương trình, SVD giảm nó thành một hệ thống đường chéo của các phương trình. Nếu A có hạng đầy đủ thì hệ thống đường chéo là không suy biến.

Như trước đó, (2.6.21) có thể được suy ra từ các phương trình chuẩn tắc. Nếu $A^*Ax = A^*b$ thì $V\widehat{\sum}^*\widehat{U}^*\widehat{U}\widehat{\sum}V^*x = V\widehat{\sum}^*\widehat{U}^*b$, kéo theo $\widehat{\sum}V^*x = \widehat{U}^*b$.

Đếm số phép toán cho Thuật toán 2.8 bị ảnh hưởng bởi sự tính toán của SVD. Cho $m \gg n$ chi phí này được xấp xỉ giống như phân tích QR, nhưng cho $m \approx n$ SVD là tốn kém hơn nhiều. Một ước lượng tiêu chuẩn là

$$\text{Thuật toán 2.8: } \sim 2mn^2 + 11n^3 \text{ phép toán dấu chấm động.} \tag{2.6.22}$$

Mỗi phương pháp ta đã miêu tả là thuận lợi trong các tình huống nào đó. Khi xét tốc độ, Thuật toán 2.6 là tốt nhất. Tuy nhiên, việc giải các phương trình chuẩn tắc không thường xuyên là ổn định trong sự có mặt của các sai số làm tròn, và do đó với nhiều năm, các nhà phân tích số đã giới thiệu Thuật toán 3.1 thay thế như là phương pháp tiêu chuẩn cho các bài toán bình phương nhỏ nhất. Nếu A gần với hạng không đầy đủ thì Thuật toán 3.1 có tính chất ổn định thấp hơn ý tưởng, và trong trường hợp như vậy có một lý do tốt để đưa ra Thuật toán 2.8, được dựa vào SVD.

Bài tập

1. Cho $A \in \mathbb{R}^{m\times n}$ với $m > n$. Chứng minh rằng nếu AA^* không suy biến nếu và chỉ nếu A có hạng đầy đủ.

2. Giả sử $P \in \mathbb{R}^{m\times m}$ thỏa $\|P^TP - I_m\|_2 = \epsilon < 1$. Chứng minh rằng tất cả các giá trị suy biến của P nằm trong khoảng $[1-\epsilon, 1+\epsilon]$ và $\|P - UV^T\|_2 \leqslant \epsilon$, với $P = U\sum V^T$ là SVD của P.

3. Giả sử ma trận A có dạng
$$A = \begin{bmatrix} A_1 \\ A_2 \end{bmatrix}$$
với A_1 là ma trận không suy biến có $n \times n$ chiều và A_2 là ma trận bất kỳ có $(m-n)\times n$ chiều. Chứng minh rằng $\|A^+\|_2 \leqslant \|A_1^{-1}\|_2$.

4. Cho ma trận
$$A = \begin{bmatrix} 1 & 0 \\ 0 & 1 \\ 1 & 0 \end{bmatrix}$$
Xác định phép chiếu trực giao P lên $range(A)$.

5. Cài đặt thuật toán trực giao hóa Gram – Schmidt của họ $\{u_1, u_2, \ldots, u_n\}$ độc lập tuyến tính.

6. Cho n vector $a_1, a_2, \ldots, a_n$ trong không gian vector $\mathbb{R}^m$.

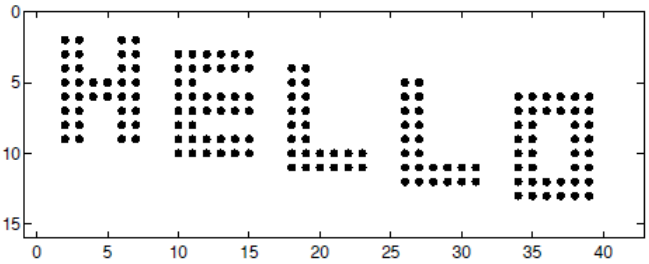
   a) Hãy viết thuật toán Gram-schmidt cổ điển với n vector $a_1, a_2, \ldots, a_n$.

   b) Viết chương trình đếm số phép gán và số phép so sánh của thuật toán vừa viết theo m và n.

   c) Ước lượng độ phức tạp thuật toán theo tổng số phép toán gán và so sánh.

7. Cho A là ma trận $m \times n$ $(m \geqslant n)$ và cho $A = \widehat{Q}\widehat{R}$ là SVD được sửa đổi của A. Chứng minh rằng A có hạng đầy đủ nếu và chỉ nếu các phần tử trên đường chéo của $\widehat{R}$ đều khác 0.

8. a) Viết một chương trình Matlab cài đặt một ma trận 15×40 với các phần tử 0 khắp nơi ngoại trừ 1 ở các vị trí được cho biết trong hình bên dưới. Số 1 ở vị trí nhất ở trên là nằm ở vị trí (2,2), và số 1 ở vị trí phải nhất ở phía dưới là nằm ở vị trí (13,29). Hình này được đưa ra với lệnh $spy(A)$.



b) Gọi SVD để tính các giá trị suy biến của A, và in các kết quả. Vẽ các số này sử dụng cả *plot* và *semily*. Hạng chính xác của A? Điều này cho thấy các giá trị suy biến được tính toán như thế nào?

c) Với mỗi i từ 1 tới $rank(A)$, xây dựng các ma trận hạng i B mà nó là xấp xỉ tốt nhất cho A trong chuẩn 2. Sử dụng các lệnh $pcolor(B)$ với $colormap(gray)$ để khởi tạo các ảnh của các xấp xỉ khác nhau này.

9. Cho x và y là các vector khác không của $\mathbb{R}^m$. Cho một thuật toán xác định ma trận Householder P sao cho Px là bội của y.

10. a) Viết một hàm $[W, R] = house(A)$ trong Matlab tính biểu diễn ẩn của phân tích QR đầy đủ $A = QR$ của một ma trận A $m \times n$ với $m \geqslant n$ sử dụng các phản xạ Householder. Các biến đầu ra là một ma trận tam giác dưới $W \in \mathbb{C}^{m\times n}$ mà các cột của nó là các vector v_k xác định các phản xạ Householder liên tiếp, và một ma trận tam giác $R \in \mathbb{C}^{m\times n}$.

b) Viết một hàm $Q = formQ(W)$ mà nó lấy ma trận W đưa ra bởi *house* như đầu vào và sinh ra một ma trận Q trực giao $m \times m$ tương ứng.

Chương 3

Điều kiện và tính ổn định

3.1 Điều kiện của một bài toán

3.1.1 Điều kiện của một bài toán

Về mặt lý thuyết, ta có thể xem một *bài toán* như là một hàm $f : X \to Y$ từ một không gian vector định chuẩn X của dữ liệu vào một không gian vector định chuẩn Y của các lời giải. Hàm f này thường là không tuyến tính (ngay trong đại số tuyến tính), nhưng ít nhất nó là hàm liên tục.

Một bài toán *điều kiện tốt* là một bài toán với tính chất mà tất cả các nhiễu nhỏ của x chỉ dẫn đến các thay đổi nhỏ trong $f(x)$. Một bài toán *điều kiện xấu* là một bài toán với tính chất mà một nhiễu nhỏ bất kì của x dẫn tới một thay đổi lớn trong $f(x)$.

3.1.2 Số điều kiện tuyệt đối

Cho δx là một nhiễu nhỏ của x, và viết $\delta f = f(x + \delta x) - f(x)$. *Số điều kiện tuyệt đối* $\widehat{\kappa} = \widehat{\kappa}(x)$ của bài toán f tại x được xác định như sau

$$\widehat{\kappa} = \lim_{\delta \to 0} \sup_{\|\delta x\| \leqslant \delta} \frac{\|\delta f\|}{\|\delta x\|}. \tag{3.1.1}$$

Cho hầu hết các bài toán, giới hạn của cân trên đúng trong công thức này có thể được làm sáng tỏ như một cân trên đúng trên tất cả các nhiễu nhỏ vô cùng δx. Thông thường ta sẽ viết công thức đơn giản như sau

$$\widehat{\kappa} = \sup_{\delta x} \frac{\|\delta f\|}{\|\delta x\|}, \tag{3.1.2}$$

với δx và δf là nhỏ vô cùng.

Nếu f là khả vi thì ta có thể ước lượng số điều kiện bằng các trung bình đạo hàm của f. Cho $J(x)$ là một ma trận mà phần tử i, j của nó là đạo hàm riêng phần $\partial f_i / \partial x_j$ được ước lượng tại x, được biết như là *Jacobian* của f tại x. Định nghĩa của đạo hàm cho $\delta f \approx J(x)\delta x$ với đẳng thức trong giới hạn $\|\delta x\| \to 0$. Số điều kiện tuyệt đối trở thành

$$\widehat{\kappa} = \|J(x)\|, \tag{3.1.3}$$

với $\|J(x)\|$ là chuẩn của $J(x)$ sinh ra bởi các chuẩn trong X và Y.

3.1.3 Số điều kiện tương đối

Số điều kiện tương đối $\kappa = \kappa(x)$ được xác định bởi

$$\kappa = \lim_{\delta \to 0} \sup_{\|\delta x\| \leqslant \delta} \left(\frac{\|\delta f\|}{\|f(x)\|} \Big/ \frac{\|\delta x\|}{\|x\|} \right), \tag{3.1.4}$$

hoặc giả sử δx và δf là nhỏ vô cùng

$$\kappa = \sup_{\delta x} \left(\frac{\|\delta f\|}{\|f(x)\|} \Big/ \frac{\|\delta x\|}{\|x\|} \right). \tag{3.1.5}$$

Nếu f là khả vi thì ta có thể biểu diễn con số này trong các số hạng của Jacobian:

$$\kappa = \frac{\|J(x)\|}{\|f(x)\|/\|x\|}. \tag{3.1.6}$$

Một bài toán *điều kiện tốt* nếu κ là nhỏ (ví dụ, $1, 10, 10^2$), và *điều kiện xấu* nếu κ là lớn (ví dụ, $10^6, 10^{16}$).

3.1.4 Ví dụ

Ví dụ 3.1.1. Xét bài toán $f : x \mapsto x/2$, với $x \in \mathbb{C}$. Jacobian của hàm f phải là đạo hàm $J = f' = 1/2$, nên theo (3.1.6),

$$\kappa = \frac{\|J\|}{\|f(x)\|/\|x\|} = \frac{1/2}{(x/2)/x} = 1.$$

Bài toán này là có điều kiện tốt với chuẩn bất kỳ.

Ví dụ 3.1.2. Xét bài toán $f : x \mapsto \sqrt{x}$ với $x > 0$. Jacobian của f là đạo hàm $J = f' = 1/(2\sqrt{x})$, nên ta có

$$\kappa = \frac{\|J\|}{\|f(x)\|/\|x\|} = \frac{1/(2\sqrt{x})}{\sqrt{x}/x} = \frac{1}{2}.$$

Đây cũng là bài toán có điều kiện tốt.

Ví dụ 3.1.3. Xét bài toán $f(x) = x_1 - x_2$ từ vector $x = (x_1, x_2)^* \in \mathbb{C}^2$. Cho đơn giản, ta sử dụng chuẩn ∞ trong không gian dữ liệu $\mathbb{C}^2$. Jacobian của f là

$$J = \begin{bmatrix} \dfrac{\partial f}{\partial x_1} & \dfrac{\partial f}{\partial x_2} \end{bmatrix} = [1 \ -1],$$

với $\|J\|_\infty = 2$. Khi đó số điều kiện là

$$\kappa = \frac{\|J\|_\infty}{\|f(x)\|/\|x\|} = \frac{2}{|x_1 - x_2|/\max\{|x_1|, |x_2|\}}.$$

Con số này là lớn nếu $|x_1 - x_2| \approx 0$ nên bài toán này là điều kiện xấu khi $x_1 \approx x_2$.

Ví dụ 3.1.4. Xét bài toán tính $f(x) = \tan x$ cho x gần 10^{100}. Trong bài toán này, các nhiễu tương đối rất nhỏ trong x có thể đưa đến các thay đổi lớn tùy ý trong $\tan x$. Kết quả: $\tan(10^{100})$ là không thể tính được trong hầu hết các máy tính.

Ví dụ 3.1.5. Xác định các nghiệm của một đa thức với các hệ số được cho trước là một ví dụ cổ điển của bài toán điều kiện xấu. Xét $x^2 - 2x + 1 = (x-1)^2$, với nghiệm bội $x = 1$. Một nhiễu nhỏ trong các hệ số có thể dẫn đến một thay đổi lớn hơn trong các nghiệm. Ví dụ, $x^2 - 2x + 0.9999 = (x - 0.99)(x - 1.01)$. Thật vậy, các nghiệm có thể thay đổi tương ứng với

căn bậc hai của thay đổi trong các hệ số nên trong trường hợp này Jacobian là vô hạn (bài toán là không khả vi), và $\kappa = \infty$.

Việc tìm nghiệm đa thức là đặc thù của bài toán điều kiện xấu trong các trường hợp mà chúng không bao gồm các nghiệm bội. Nếu hệ số a_i của đa thức $p(x)$ được làm nhiễu bằng một con số nhỏ vô cùng δa_i thì nhiễu của nghiệm thứ j, x_j là $\delta x_j = -\dfrac{(\delta a_i)x_j^i}{p'(x_j)}$, với p' là đạo hàm của p. Do đó, số điều kiện của x_j tương ứng với các nhiễu của hệ số đơn a_i là

$$\kappa = \frac{|\delta x_j|}{|x_j|} \Big/ \frac{|\delta a_i|}{|a_i|} = \frac{|a_i x_j^{i-1}|}{|p'(x_j)|}. \tag{3.1.7}$$

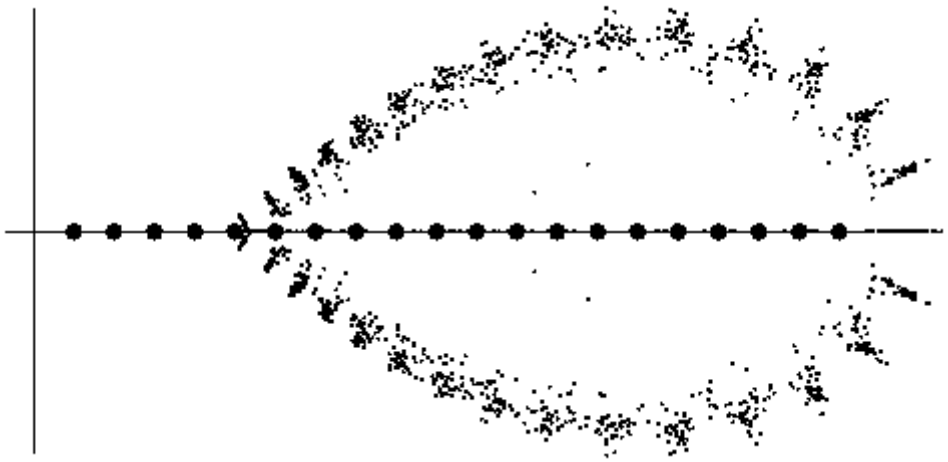
Số này thường là rất lớn. Xét "đa thức Wilkinson"

$$p(x) = \prod_{i=1}^{20}(x-i) = a_0 + a_1 x + \ldots + a_{19}x^{19} + x^{20}. \tag{3.1.8}$$

Nghiệm dễ bị ảnh hưởng nhất của đa thức này là $x = 15$, và nó thay đổi hệ số $a_{15} \approx 1.67 \times 10^9$. Số điều kiện là

$$\kappa \approx \frac{1.67 \times 10^9 \cdot 15^{14}}{5!14!} \approx 5.1 \times 10^{13}.$$

Hình 3.1 miêu tả bài toán điều kiện xấu.



Hình 3.1: Ví dụ điều kiện xấu kinh điển của Wilkinson. Các dấu chấm lớn là các nghiệm của đa thức (3.1.8) chưa bị nhiễu. Các dấu chấm nhỏ là các nghiệm được thêm vào trong mặt phẳng phức của 100 đa thức bị nhiễu ngẫu nhiên với các hệ số được xác định bởi $\widetilde{a}_k = a_k(1 + 10^{-10}r_k)$, với r_k là một số từ phân phối chuẩn của trung bình 0 và phương sai 1

Ví dụ 3.1.6. Bài toán tính trị riêng của một ma trận không đối xứng cũng thường là bài toán điều kiện xấu. So sánh hai ma trận

$$\begin{bmatrix} 1 & 1000 \\ 0 & 1 \end{bmatrix}, \quad \begin{bmatrix} 1 & 1000 \\ 0.001 & 1 \end{bmatrix}, \tag{3.1.9}$$

mà các trị riêng của chúng tương ứng là $\{1, 1\}$ và $\{0, 2\}$. Mặt khác, nếu một ma trận A là đối xứng (tổng quát hơn, nếu nó là chuẩn tắc), thì các trị riêng của nó là điều kiện tốt. Nếu λ và $\lambda + \delta\lambda$ tương ứng với các trị riêng của A và $A + \delta A$ thì $|\delta\lambda| \leqslant \|\delta A\|_2$, dấu bằng xảy ra nếu δA là một bội của ma trận đơn vị. Do đó số điều kiện tuyệt đối của bài toán trị riêng đối xứng là $\widehat{\kappa} = 1$, nếu các nhiễu được đo trong chuẩn 2 thì số điều kiện tương đối là $\kappa = \dfrac{\|A\|_2}{|\lambda|}$.

3.1.5 Điều kiện của phép nhân ma trận với vector

Cố định $A \in \mathbb{C}^{m\times n}$ và xét bài toán tính Ax từ số liệu đầu vào x. Từ định nghĩa của κ, với $\|\cdot\|$ là chuẩn vector tùy ý và ma trận được bao gồm tương ứng, ta thấy

$$\kappa = \sup_{\delta x}\left(\frac{\|A(x+\delta x)-Ax\|}{\|Ax\|}\Big/\frac{\|\delta x\|}{\|x\|}\right) = \sup_{\delta x}\frac{\|A\delta x\|}{\|\delta x\|}\Big/\frac{\|\delta x\|}{\|x\|}$$

Do đó

$$\kappa = \|A\|\frac{\|x\|}{\|Ax\|} \tag{3.1.10}$$

(trường hợp đặc biệt của (3.1.6)). Đây là một công thức chính xác cho κ, phụ thuộc vào cả A và x.

Giả sử A là ma trận vuông và không suy biến. Khi đó ta có thể sử dụng $\dfrac{\|x\|}{\|Ax\|} \leqslant \|A^{-1}\|$ để làm rời (3.1.10) thành một chặn độc lập với x:

$$\kappa \leqslant \|A\|\|A^{-1}\|. \tag{3.1.11}$$

Hoặc

$$\kappa = \alpha\|A\|\|A^{-1}\| \tag{3.1.12}$$

với

$$\alpha = \frac{\|x\|}{\|Ax\|}\Big/\|A^{-1}\|. \tag{3.1.13}$$

Cho các sự lựa chọn nào đó của x, ta có $\alpha = 1$, và do đó $\kappa = \|A\|\|A^{-1}\|$. Nếu $\|\cdot\| = \|\cdot\|_2$ thì điều này sẽ xuất hiện bất kỳ lúc nào x là một bội của vector suy biến phải cực tiểu của A.

Thật vậy, A không cần phải là ma trận vuông. Nếu $A \in \mathbb{C}^{m\times n}$ với $m \geqslant n$ có hạng đầy đủ, các phương trình (3.1.11) - (3.1.13) đúng với A^{-1} được thay thế bằng giả nghịch đảo A^{+} được xác định trong (2.6.11).

Định lý 3.1.1 *Cho $A \in \mathbb{C}^{m\times m}$ là không suy biến và xét phương trình $Ax = b$. Bài toán tính b, x được cho, có số điều kiện*

$$\kappa = \|A\|\frac{\|x\|}{\|b\|} \leqslant \|A\|\|A^{-1}\| \tag{3.1.14}$$

tương ứng với các nhiễu của x. Bài toán tính x, b được cho, có số điều kiện

$$\kappa = \|A^{-1}\|\frac{\|b\|}{\|x\|} \leqslant \|A\|\|A^{-1}\| \tag{3.1.15}$$

tương ứng với các nhiễu của b. Nếu $\|\cdot\| = \|\cdot\|_2$ thì dấu bằng trong (3.1.14) xảy ra nếu x là một bội của vector suy biến phải của A tương ứng với giá trị suy biến nhỏ nhất σ_m, và dấu bằng trong (3.1.15) xảy ra nếu b là bội của một vector suy biến trái của A tương ứng với giá trị suy biến lớn nhất σ_1.

3.1.6 Số điều kiện của một ma trận

Tích $\|A\|\|A^{-1}\|$ là *số điều kiện* của A (liên quan tới chuẩn $\|\cdot\|$), được ký hiệu bởi $\kappa(A)$:

$$\kappa(A) = \|A\|\|A^{-1}\|. \tag{3.1.16}$$

Nếu $\kappa(A)$ nhỏ thì A được gọi là *điều kiện tốt*. Nếu $\kappa(A)$ lớn thì A là *điều kiện xấu*. Nếu A suy biến thì $\kappa(A) = \infty$.

Nếu $\|\cdot\| = \|\cdot\|_2$ thì $\|A\| = \sigma_1$ và $\|A^{-1}\| = \dfrac{1}{\sigma_m}$. Do đó

$$\kappa(A) = \frac{\sigma_1}{\sigma_m} \tag{3.1.17}$$

trong chuẩn 2, và công thức này được sử dụng cho việc tính số điều kiện chuẩn 2 của các ma trận. Tỉ số $\dfrac{\sigma_1}{\sigma_m}$ được giải thích như là độ lệch tâm của siêu ellip mà nó là ảnh của quả cầu đơn vị của $\mathbb{C}^m$ dưới A (Hình 1.2).

Cho một ma trận hình chữ nhật $A \in \mathbb{C}^{m\times n}$ hạng đầy đủ, $m \geqslant n$, số điều kiện được xác định: $\kappa(A) = \|A\|\|A^+\|$. Vì A^+ được thúc đẩy bởi các bài toán bình phương nhỏ nhất nên định nghĩa này là hữu ích trong trong trường hợp $\|\cdot\| = \|\cdot\|_2$. Ta có

$$\kappa(A) = \frac{\sigma_1}{\sigma_m} \tag{3.1.18}$$

3.1.7 Điều kiện của một hệ thống các phương trình

Trong Định lý 3.1.1, ta cho A được cố định và x hoặc b được làm nhiễu. Đặc biệt, ta cho b cố định và xét bài toán $A \mapsto x = A^{-1}b$ khi A được làm nhiễu bằng δA nhỏ vô cùng. Khi đó, x phải thay đổi bằng δx nhỏ vô cùng, với

$$(A + \delta A)(x + \delta x) = b.$$

Sử dụng phương trình $Ax = b$ và việc giảm số hạng nhỏ vô cùng $(\delta A)(\delta x)$ xuống, ta được $(\delta A)x + A(\delta x) = 0$. Do đó, $\delta x = -A^{-1}(\delta A)x$. Phương trình này kéo theo $\|\delta x\| \leqslant \|A^{-1}\|\|\delta A\|\|x\|$, hoặc tương đương,

$$\frac{\|\delta x\|}{\|x\|} \Big/ \frac{\|\delta A\|}{\|A\|} \leqslant \|A\|\|A^{-1}\| = \kappa(A).$$

Dẳng thức xảy ra khi δA thỏa

$$\|A^{-1}(\delta A)x\| = \|A^{-1}\|\|\delta A\|\|x\|,$$

và nó có thể được chứng minh bằng sử dụng các chuẩn đối ngẫu mà A, b, và chuẩn $\|\cdot\|$ bất kỳ, các nhiễu δA như vậy tồn tại.

Định lý 3.1.2 *Cho b cố định và xét bài toán $x = A^{-1}b$, với A là ma trận vuông và không suy biến. Số điều kiện của bài toán này với các nhiễu tương ứng trong A là*

$$\kappa = \|A\|\|A^{-1}\| = \kappa(A). \tag{3.1.19}$$

Nếu một bài toán $Ax = b$ chứa một ma trận điều kiện xấu A thì thường ta phải hy vọng "mất $\log_{10}\kappa(A)$ chữ số" trong việc tính toán tìm lời giải, ngoại trừ dưới các trường hợp rất đặc biệt.

3.2 Số học dấu chấm động

3.2.1 Hạn chế của biểu diễn bằng số

Vì các máy tính bằng số sử dụng một số hữu hạn các số nhị phân để biểu diễn một số thực nên chúng chỉ có thể biểu diễn một tập con hữu hạn của các số thực (hoặc số phức). Hạn chế này đưa ra hai khó khăn. Đầu tiên, các số được biểu diễn không thể là lớn hoặc nhỏ tùy ý. Thứ hai, ở đây phải là các khoảng trống giữa chúng.

Các máy tính hiện đại biểu diễn các số đủ lớn và nhỏ mà hạn chế đầu tiên hiếm khi đưa ra các khó khăn. Ví dụ, số học chính xác gấp đôi IEEE được sử dụng một cách thưa thớt cho phép các số lớn như 1.79×10^{308} và nhỏ như 2.23×10^{-308}. Mặt khác, *tràn số* và *tràn dưới* thông thường không là nguy cơ quan trọng.

Bằng phản chứng, bài toán các khoảng trống giữa các số được biểu diễn liên quan với tính toán khoa học. Ví dụ, trong số học độ chính xác bội IEEE, khoảng [1,2] được biểu diễn bằng một tập con rời rạc

$$1, 1 + 2^{-52}, 1 + 2 \times 2^{-52}, 1 + 3 \times 2^{-52}, \ldots, 2. \tag{3.2.1}$$

Khoảng [2,4] được biểu diễn bằng các số bội của 2 ,

$$2, 2 + 2^{-51}, 2 + 2 \times 2^{-51}, 2 + 3 \times 2^{-51}, \ldots, 4,$$

Tổng quát, khoảng $[2^j, 2^{j+1}]$ được biểu diễn bằng (3.2.1) nhân với 2^j. Do đó trong số học độ chính xác bội IEEE, các khoảng trống giữa các số gần kề không bao giờ lớn hơn $2^{-52} \approx 2.22 \times 10^{-16}$.

3.2.2 Số chấm động

Số học IEEE là một ví dụ của một hệ thống số học được dựa vào sự biểu diễn *dấu chấm động* của các số thực. Trong một hệ thống số dấu chấm động, vị trí của dấu chấm thập phân (hoặc nhị phân) được lưu trữ một cách tách biệt từ các chữ số, và các khoảng trống giữa các số vô hướng được biểu diễn gần kề trong kích thước của các số. Điều này phân biệt với biểu diễn *dấu cố định* nơi mà các khoảng trống có cùng kích thước.

Hệ thống dấu chấm động của tập con rời rạc $\mathbf{F}$ của các số thực $\mathbb{R}$ xác định bởi một số nguyên $\beta \geqslant 2$ được biết như là *cơ số* hay *radix*(tiêu biểu là 2) và một số nguyên $t \geqslant 1$ được biết như là *độ chính xác* (24 hoặc 53 cho IEEE độ chính xác đơn và bội tương ứng). Các phần tử của $\mathbf{F}$ là số 0 cùng với tất cả các số có dạng

$$x = \pm(m/\beta^t)\beta^e, \tag{3.2.2}$$

với m là số nguyên nằm trong $1 \leqslant m \leqslant \beta^t$ và e là một số nguyên tùy ý. Ta có thể hạn chế thành $\beta^{t-1} \leqslant m \leqslant \beta^t - 1$ và do đó việc chọn m là duy nhất. Khi đó, con số $\pm(m/\beta^t)$ được biết như *phân số* hoặc *phần định trị* của x, và e là *số mũ*.

3.2.3 Machine Epsilon

Lời giải của $\mathbf{F}$ được tóm tắt bởi một số được biết như *machine epsilon*. Tạm thời, ta hãy xác định số này bằng

$$\epsilon_{machine} = \frac{1}{2}\beta^{1-t}. \tag{3.2.3}$$

(Ta sẽ bổ sung định nghĩa sau (3.2.7).) Số này là phân nửa khoảng cách giữa 1 và số dấu chấm động lớn hơn tiếp theo. $\epsilon_{machine}$ có tính chất theo sau:

$$\text{Với mọi } x \in \mathbb{R}, \text{ tồn tại } x' \in \boldsymbol{F} \text{ sao cho } |x - x'| \leqslant \epsilon_{machine}|x|. \tag{3.2.4}$$

Cho các giá trị của β và t thông thường trong các máy tính khác nhau, $\epsilon_{machine}$ thường nằm giữa 10^{-6} và 10^{-35}. Trong số học độ chính xác đơn và bội IEEE, $\epsilon_{machine}$ được thiết lập tương ứng là $2^{-24} \approx 5.96 \times 10^{-8}$ và $2^{-53} \approx 1.11 \times 10^{-16}$.

Cho $fl : \mathbb{R} \to \boldsymbol{F}$ là hàm cho xấp xỉ dấu chấm động gần với một số thực nhất. Bất đẳng thức 3.2.4 có thể được phát biểu liên quan tới fl

$$\text{Với mọi } x \in \mathbb{R}, \text{ tồn tại } \epsilon \text{ với } |\epsilon| \leqslant \epsilon_{machine} \text{ sao cho } fl(x) = x(1 + \epsilon). \tag{3.2.5}$$

Sự khác nhau giữa một số thực và xấp xỉ dấu chấm động gần nó nhất thường là lớn hơn $\epsilon_{machine}$ trong các số hạng liên quan.

3.2.4 Số học dấu chấm động

Trong một máy tính, tất cả các phép toán toán học được giảm xuống thành các phép toán số học chủ yếu như $+, -, \times$ và $\div$. Theo toán học, các ký hiệu này biểu diễn các phép toán trong $\mathbb{R}$. Trong một máy tính, chúng là các phép toán trong $\mathbf{F}$. Các phép toán dấu chấm động này được ký hiệu bởi $\oplus, \ominus, \otimes$ và $\oslash$.

Cho x và y là các số dấu chấm động tùy ý, nghĩa là $x, y \in \boldsymbol{F}$. Cho $*$ là một trong số những phép toán $+, -, \times$, hoặc $\div$, và cho $\circledast$ là một trong những phép toán dấu chấm động $\oplus, \ominus, \otimes$ hoặc $\oslash$. Khi đó,

$$x \circledast y = fl(x * y). \tag{3.2.6}$$

Nếu tính chất này đúng thì từ (3.2.5) và (3.2.6), máy tính có tính chất đơn giản và mạnh như sau

Tiên đề cơ sở của số học dấu chấm động
Với mọi $x, y \in \boldsymbol{F}$, tồn tại ϵ với $|\epsilon| \leqslant \epsilon_{machine}$ sao cho

$$x \circledast y = (x * y)(1 + \epsilon). \tag{3.2.7}$$

Mặt khác, *mọi phép toán số học dấu chấm động là chính xác lên tới một sai số tương đối của kích thước nhiều nhất* $\epsilon_{machine}$.

3.2.5 Số học dấu chấm động phức

Các số phức dấu chấm động được biểu diễn như là các cặp của các số thực dấu chấm động, và các phép toán cơ bản trên chúng được tính bằng sự giảm bớt thành các phần thực và phần ảo. Tiên đề (3.2.7) là hợp lý cho số phức như là các số học dấu chấm động thực, ngoại trừ cho $\otimes$ và $\oslash$, $\epsilon_{machine}$ phải được tăng lên từ (3.2.3) bằng các thừa số trong bậc $2^{3/2}$ và $2^{5/2}$ tương ứng. Một $\epsilon_{machine}$ được điều chỉnh trong kiểu này, phân tích sai số làm tròn cho các số phức có thể tiến hành như cho các số thực.

3.3 Tính ổn định

3.3.1 Các thuật toán

Trong mục 3.1, ta xác định *bài toán* toán học như là một hàm $f : X \to Y$ từ không gian vector X của dữ liệu vào một không gian vector Y của các lời giải.

Một *thuật toán* có thể được xem như một ánh xạ khác $\tilde{f} : X \to Y$ giữa hai không gian giống nhau. Cho một bài toán f, một máy tính mà hệ thống dấu chấm động của nó thỏa mãn (3.2.7) (nhưng không cần thiết thỏa (3.2.6)), một thuật toán cho f và một sự thực thi thuật toán này trong dạng của một chương trình máy tính được cố định. Dữ liệu $x \in X$ được cho, dữ liệu này được làm tròn tới dấu chấm động thỏa mãn (3.2.5) và khi đó được áp dụng như đầu vào của chương trình máy tính. Kết quả là sự tập hợp các số dấu chấm động mà chúng thuộc không gian vector Y (vì thuật toán được thiết kế để giải f). Cho kết quả tính được này được gọi là $\tilde{f}(x)$.

$\tilde{f}(x)$ sẽ bị ảnh hưởng bởi các sai số làm tròn. Phụ thuộc vào các trường hợp, nó cũng có thể bị ảnh hưởng tất cả các loại phức tạp khác như sự hội tụ cho phép hoặc cũng như các công việc khác chạy trong máy tính, trong các trường hợp mà phép gán của các tính toán thành các bộ xử lý không được xác định cho tới khi chạy. Do đó "hàm" $\tilde{f}(x)$ cũng có thể lấy các giá trị khác nhau từ 1 tới số tiếp theo nên nó có thể là đa trị. (Thật vậy, bài toán f nên được cho phép là đa trị; điều này cho phép xử lý bằng tay các trường hợp mà một lời giải không duy nhất là có thể chấp nhận được, ví dụ, hai căn bậc hai của một số phức.).

Ký hiệu dấu ($\tilde{}$) là rất thích hợp. $\tilde{f}$ được tính tương tự như f.

3.3.2 Sự đúng đắn

Ngoài các trường hợp tầm thường, $\tilde{f}$ không thể là hàm liên tục. Tuy nhiên, một thuật toán tốt nên xấp xỉ bài toán được kết hợp f. Ta xét *sai số tuyệt đối* của một phép tính, $\|\tilde{f}(x)-f(x)\|$, hoặc *sai số tương đối*,

$$\frac{\|\tilde{f}(x)-f(x)\|}{\|f(x)\|}. \tag{3.3.1}$$

Trong sách này ta chủ yếu sử dụng các con số tương đối, và do đó (3.3.1) sẽ là độ đo sai số tiêu chuẩn của chúng ta.

Nếu $\tilde{f}$ là một thuật toán tốt thì nó có thể loại ra sai số tương đối là nhỏ của bậc $\epsilon_{machine}$. Thuật toán $\tilde{f}$ cho một bài toán f là *đúng đắn* nếu với mỗi $x \in X$,

$$\frac{\|\tilde{f}(x)-f(x)\|}{\|f(\tilde{x})\|} = O(\epsilon_{machine}). \tag{3.3.2}$$

3.3.3 Tính ổn định

Ta nói rằng một thuật toán cho bài toán f là *ổn định* nếu với mọi $x \in X$,

$$\frac{\|\tilde{f}(x)-f(\tilde{x})\|}{\|f(\tilde{x})\|} = O(\epsilon_{machine}) \tag{3.3.3}$$

với $\tilde{x}$ nào đó thỏa

$$\frac{\|\tilde{x}-x\|}{\|x\|} = O(\epsilon_{machine}). \tag{3.3.4}$$

3.3.4 Tính ổn định ngược

Ta nói rằng một thuật toán $\tilde{f}$ cho một bài toán f là *ổn định ngược* nếu với mọi $x \in X$,

$$\tilde{f}(x) = f(\tilde{x}) \text{ với } \tilde{x} \text{ nào đó thỏa } \frac{\|\tilde{x}-x\|}{\|x\|} = O(\epsilon_{machine}). \tag{3.3.5}$$

Đây là một việc xiết chặt xác định tính ổn định trong đó $O(\epsilon_{machine})$ trong (3.3.3) đã được thay thế bởi 0.

3.3.5 Ý nghĩa của $O(\epsilon_{machine})$

Bây giờ ta giải thích rõ ràng ý nghĩa của "$O(\epsilon_{machine})$" trong 3.3.2 - 3.3.5.

Ký hiệu

$$\varphi(t) = O(\psi(t)) \tag{3.3.6}$$

là một tiêu chuẩn trong toán học, với một định nghĩa rõ ràng. Phương trình này khẳng định rằng tồn tại hằng số dương nào đó C sao cho, với mọi t đủ gần một giới hạn đã biết (ví dụ, $t \to 0$ hoặc $t \to \infty$),

$$|\varphi(t)| \leqslant C\psi(t). \tag{3.3.7}$$

ví dụ, $\sin^2 t = O(t^2)$ khi $t \to 0$ khẳng định rằng tồn tại một hằng số C sao cho, với mọi t đủ nhỏ, $\sin^2 t| \leqslant Ct^2$.

Hơn nữa tiêu chuẩn trong toán học là các phát biểu có dạng

$$\varphi(s,t) = O(\psi(t)) \text{ cùng kiểu trong } s, \tag{3.3.8}$$

với φ là hàm phụ thuộc vào biến t và s. Từ "cùng kiểu" cho biết tồn tại một hằng số C như trong (3.3.7) mà nó là đúng cho mọi sự lựa chọn của s. Do đó, ví dụ

$$(\sin^2 t)(\sin^2 x) = O(t^2)$$

cùng kiểu khi $t \to 0$, nhưng sự cùng kiểu là tổn thất nếu ta thay thế $\sin^2 s$ bằng s^2.

Trong sách này, sử dụng ký hiệu "O"

$$\|\text{con số được tính}\| = O(\epsilon_{machine}). \tag{3.3.9}$$

Đầu tiên, "$\|$con số được tính$\|$" biểu diễn chuẩn của một số nào đó hoặc sự lựa chọn các số được xác định bởi một thuật toán $\tilde{f}$ cho một bài toán f, phụ thuộc cả hai dữ liệu $x \in X$ cho f và $\epsilon_{machine}$. Ví dụ sai số tương đối trong (3.3.1). Thứ hai, quá trình giới hạn ẩn là $\epsilon_{machine} \to 0$ (nghĩa là, $\epsilon_{machine}$ là biến tương ứng với t trong (3.3.8)). Thứ ba, "O" áp dụng cùng kiểu cho mọi dữ liệu $x \in X$ (nghĩa là, x là biến tương ứng với s). Ta sẽ ít khi đề cập sự cùng kiểu tương ứng với $x \in X$ mà nó thường là ẩn.

Phương trình (3.3.9) nói rằng nếu ta chạy thuật toán câu hỏi trong các máy tính thỏa (3.2.5 và (3.2.7) cho một chuỗi các giá trị của $\epsilon_{machine}$ mà nó giảm xuống thành 0, khi đó $\|$con số được tính$\|$ sẽ được đảm bảo để giảm xuống trong tỷ lệ thức với $\epsilon_{machine}$ hoặc nhanh hơn. Các máy tính lý tưởng này được yêu cầu để thỏa mãn (3.2.5) và (3.2.7) nhưng không thỏa yêu cầu khác.

3.3.6 Phụ thuộc vào m và n, không phụ thuộc A và b

Giả sử ta đang xét một thuật toán cho việc giải một hệ thống $m \times m$ không suy biến của các phương trình $Ax = b$ cho biến x, và ta khẳng định rằng kết quả tính được $\tilde{x}$ cho thuật toán này thỏa mãn

$$\frac{\|\tilde{x} - x\|}{\|x\|} = O(\kappa(A)\epsilon_{machine}). \tag{3.3.10}$$

Khẳng định này nghĩa là chặn

$$\frac{\|\tilde{x} - x\|}{\|x\|} \leqslant C\kappa(A)\epsilon_{machine}. \tag{3.3.11}$$

đúng cho một hằng số C, không phụ thuộc ma trận A hoặc vế bên phải b, với mọi $\epsilon_{machine}$ đủ nhỏ.

Nếu mẫu số trong một công thức giống (3.3.11) là 0, ý nghĩa của nó được xác định bởi quy ước theo sau. Khi ta viết (3.3.11)

$$\|\tilde{x} - x\| \leqslant C\kappa(A)\epsilon_{machine}\|x\|. \tag{3.3.12}$$

Ở đây là không khác nhau nếu $\|x\| \neq 0$, nhưng nếu $\|x\| = 0$, (3.3.12) làm rõ ý nghĩa của (3.3.10) là $\|\tilde{x} - x\| = 0$ với mọi $\epsilon_{machine}$ đủ nhỏ.

Mặc dù hằng số C của (3.3.11) hoặc (3.3.12) không phụ thuộc vào A hoặc b mà phụ thuộc vào số chiều m. Đây là một chuỗi định nghĩa của một bài toán trong mục (3.1). Nếu các chiều m hoặc n xác định một bài toán f thay đổi, khi đó các không gian X và Y cũng phải thay đổi, và do đó ta có một bài toán mới, f'. Như một vấn đề thực tiễn, các hiệu quả của các sai số làm tròn trong các thuật toán của phương pháp số trong đại số tuyến tính thường phát triển với m và n. Tuy nhiên, sự phát triển này thường là đủ chậm mà nó không đúng đắn. Phụ thuộc vào m hoặc n tiêu biểu là tuyến tính, bậc hai hoặc bậc ba trong trường hợp xấu nhất (số mũ phụ thuộc vào sự lựa chọn chuẩn tốt như chọn thuật toán), và các sai số cho hầu hết dữ liệu là nhỏ hơn nhiều trong trường hợp xấu nhất.

3.3.7 Sự độc lập của chuẩn

Định lý 3.3.1 *Cho các bài toán f và các thuật toán $\tilde{f}$ xác định trong không gian hữu hạn chiều X và Y, các tính chất của sự đúng đắn, tính ổn định và ổn định ngược độc lập với sự lựa chọn các chuẩn trong X và Y.*

Chứng minh Trong một không gian vector hữu hạn chiều X và Y, nếu $\|\cdot\|$ và $\|\cdot\|'$ là hai chuẩn trong cùng không gian thì chúng tương đương nhau. Khi đó tồn tại các hằng số dương C_1 và C_2 sao cho $C_1\|x\| \leqslant \|x\|' \leqslant C_2\|x\|$ với mọi x trong không gian đó. Sự thay đổi của chuẩn có thể làm ảnh hưởng đến kích thước của hằng số C ẩn trong một phát biểu bao gồm $O(\epsilon_{machine})$, nhưng không tồn tại một hằng số như vậy.

3.3.8 Tính ổn định của số học dấu chấm động

Bốn bài toán tính toán đơn giản nhất là $+, -, \times$, và $\div$. Ta sẽ sử dụng các phép toán dấu chấm động $\oplus, \ominus, \otimes$, và $\oslash$ được cung cấp với máy tính. Các tiên đề (3.2.5) và (3.2.7) cho thấy bốn ví dụ phù hợp với tiêu chuẩn này của các thuật toán đều là ổn định ngược.

Ta hãy chứng minh điều này cho phép trừ, vì đó là một phép toán cơ bản có thể mà ta có thể mong đợi là rủi ro lớn nhất của tính không ổn định. Như trong Ví dụ 3.1.4, không gian dữ liệu X là tập hợp các vector 2 chiều, $\mathbb{C}^2$, và không gian lời giải Y là tập hợp các vô hướng, $\mathbb{C}$.

Với dữ liệu $x = (x_1, x_2)^* \in X$, bài toán phép trừ tương ứng với hàm $f(x_1, x_2) = x_1 - x_2$, và thuật toán ta đang xét có thể được viết

$$\tilde{f}(x_1, x_2) = fl(x_1) \ominus fl(x_2).$$

Phương trình này có nghĩa rằng đầu tiên ta làm tròn x_1 và x_2 tới các giá trị dấu chấm động, khi đó áp dụng phép toán $\ominus$. Do (3.2.5), ta có

$$fl(x_1) = x_1(1 + \epsilon_1),\ fl(x_2) = x_2(1 + \epsilon_2)$$

với $|\epsilon_1|, |\epsilon_2| \leqslant \epsilon_{machine}$. Do (3.2.7), ta có

$$fl(x_1) \ominus fl(x_2) = (fl(x_1) - fl(x_2))(1 + \epsilon_3)$$

với $|\epsilon_3| \leqslant \epsilon_{machine}$. Kết hợp các phương trình này ta được

$$\begin{aligned} fl(x_1) \ominus fl(x_2) &= [x_1(1 + \epsilon_1) - x_2(1 + \epsilon_2)](1 + \epsilon_3) \\ &= x_1(1 + \epsilon_1)(1 + \epsilon_3) - x_2(1 + \epsilon_2)(1 + \epsilon_3) \\ &= x_1(1 + \epsilon_4) - x_2(1 + \epsilon_5) \end{aligned}$$

với $|\epsilon_4|, |\epsilon_5| \leqslant 2\epsilon_{machine} + O(\epsilon^2_{machine})$. Mặt khác, kết quả tính được $\tilde{f}(x) = fl(x_1) \ominus fl(x_2)$ chính xác bằng với hiệu $\tilde{x_1} - \tilde{x_2}$, với $\tilde{x_1}$, và $\tilde{x_2}$ thỏa mãn

$$\frac{|\tilde{x_1} - x_1|}{|x_1|} = O(\epsilon_{machine}),\ \frac{|\tilde{x_2} - x_2|}{|x_2|} = O(\epsilon_{machine}),$$

và $C > 2$ bất kỳ sẽ đủ cho các hằng số ẩn trong các ký hiệu "O". Với sự lựa chọn một chuẩn bất kỳ $\|\cdot\|$ trong không gian $\mathbb{C}^2$, điều này kéo theo (3.3.5).

3.3.9 Các ví dụ

Ví dụ 3.3.1. Tích trong. Giả sử ta được cho các vector $x, y \in \mathbb{C}^m$ và mong muốn tính tích trong $\alpha = x^*y$. Thuật toán rõ ràng là để tính từng cặp tích $\overline{x_i}y_i$ với $\otimes$ và cộng với $\oplus$ để thu được một kết quả tính được $\tilde{\alpha}$. Nó có thể được chứng tỏ rằng thuật toán này là ổn định ngược.
Ví dụ 3.3.2. Tích ngoài. Mặt khác, giả sử ta mong muốn tính tích ngoài hạng 1 $A = xy^*$ với các vector $x \in \mathbb{C}^m, y \in \mathbb{C}^n$. Thuật toán rõ ràng là để tính mn tích $x_i\overline{y_i}$ với $\otimes$ và tập hợp chúng thành một ma trận $\tilde{A}$. Thuật toán này là ổn định, nhưng nó không ổn định ngược. Vì ma trận $\tilde{A}$ không có hạn chính xác là 1 nên nó không thể được viết tổng quát dưới dạng $(x+\delta x)(y+\delta y)^*$. Các bài toán mà số chiều của không gian nghiệm Y là lớn hơn số chiều của không gian bài toán X, tính ổn định ngược là hiếm.
Ví dụ 3.3.3. Giả sử ta sử dụng $\oplus$ để tính $x + 1$, $x \in \mathbb{C}$: $\tilde{f}(x) = fl(x) \oplus 1$. Thuật toán này là ổn định nhưng không ổn định ngược. Lý do là cho $x \approx 0$, phép cộng $\oplus$ sẽ đưa ra các sai số tuyệt đối của kích thước $O(\epsilon_{machine})$. So với kích thước x, các sai số này là không bị chặn, nên chúng không thể được hiểu như lý do bởi các nhiễu tương đối nhỏ trong dữ liệu.Nếu bài toán đã được tính $x + y$ cho dữ liệu x và y, khi đó thuật toán sẽ là ổn định ngược.
Ví dụ 3.3.4. Cho $\cos x$, $\cos 0 \neq 0$, như trong ví dụ trước. Cho cả $\sin x$ và $\cos x$, tính ổn định ngược cũng được thể hiện bên ngoài hàm có đạo hàm bằng 0 tại các điểm nào đó. Ví dụ, giả sử ta tính $f(x) = \sin x$ trong một máy tính với $x = \pi/2 - \delta, 0 < \delta \ll 1$. Giả sử ta đủ may mắn để có được một kết quả được tính như là câu trả lời chính xác, được làm tròn tới hệ thống dấu chấm động: $\tilde{f}(x) = fl(\sin x)$. Vì $f'(x) = \cos x \approx \delta$ nên ta có $\tilde{f}(x) = f(\tilde{x})$ với $\tilde{x}$ nào đó thỏa mãn $\tilde{x} - x \approx (\tilde{f}(x) - f(x))/\delta = O(\epsilon_{machine}/\delta)$. Vì δ có thể là nhỏ tùy ý nên sai số ngược này không phải là của kích thước $O(\epsilon_{machine})$.

3.3.10 Thuật toán không ổn định

Sử dụng đa thức đặc trưng để tìm các trị riêng của một ma trận.

Vì z là một trị riêng của A nếu và chỉ nếu $p(z) = 0$, với $p(z)$ là đa thức đặc trưng của $det(zI - A)$, các nghiệm của p là các trị riêng của A. Một phương pháp được đưa ra cho việc tính toán các trị riêng:

1. Tìm các hệ số của đa thức đặc trưng,
2. Tìm các nghiệm của nó.

Thuật toán này không chỉ là không ổn định ngược mà còn là không ổn định, và nó không nên được sử dụng. Mặc dù trong các trường hợp mà việc trích các trị riêng là bài toán điều kiện tốt, nó có thể đưa ra các sai số tương đối lớn hơn $\epsilon_{machine}$.

Tính không ổn định được phát hiện trong việc tìm nghiệm của bước thứ hai. Như ta thấy trong Ví dụ 3.1.4, bài toán tìm các nghiệm của một đa thức, các hệ số được cho, nói chung là điều kiện xấu. Các sai số nhỏ trong các hệ số của đa thức đặc trưng có xu hướng được khuếch đại khi tìm các nghiệm, mặc dù việc tìm nghiệm đã hoàn thành.

Ví dụ, giả sử $A = I$, ma trận đơn vị 2×2. Các trị riêng của A không nhạy với nhiễu của các phần tử, và một thuật toán ổn định có thể tính chúng với các sai số $O(\epsilon_{machine})$. Tuy nhiên, thuật toán miêu tả ở trên đưa ra các sai số trong bậc của $\sqrt{\epsilon_{machine}}$. Đa thức đặc trưng $x^2 - 2x + 1$, ngay trong Ví dụ 3.1.4. Khi các hệ số trong đa thức này được tính, chúng có thể được mong đợi có các sai số trong bậc của $\epsilon_{machine}$, và điều này có thể là nguyên nhân làm thay đổi các nghiệm bằng bậc $\sqrt{\epsilon_{machine}}$. Ví dụ, nếu $\epsilon_{machine} = 10^{-16}$, các nghiệm của đa thức đặc trưng tính được có thể được làm nhiễu từ các trị riêng hiện tại bởi xấp xỉ 10^{-8}, sự hao hụt 8 số nhị phân của sự đúng đắn.

Sử dụng thuật toán được miêu tả để tính các trị riêng của ma trận đơn vị 2×2. Bởi vì các hệ số và các nghiệm của $x^2 - 2x + 1$ là các số nguyên nhỏ sẽ được biểu diễn một cách chính

xác trong một máy tính. Tuy nhiên, nếu ma trận A được làm nhiễu không đáng kể

$$A = \begin{bmatrix} 1 + 10^{-14} & 0 \\ 0 & 1 \end{bmatrix},$$

thì các trị riêng tính được sẽ phân biệt bằng bậc được mong đợi $\sqrt{\epsilon_{machine}}$.

3.3.11 Sự đúng đắn của thuật toán ổn định ngược

Giả sử ta có một thuật toán ổn định ngược $\tilde{f}$ cho một bài toán $f : X \to Y$. Sự đúng đắn phụ thuộc vào số điều kiện $\kappa = \kappa(x)$ của f. Nếu $\kappa(x)$ là nhỏ thì các kết quả sẽ là đúng đắn trong nghĩa tương đối, nhưng nếu nó là lớn, sự đúng đắn sẽ cho phép tương đối.

Định lý 3.3.2 *Giả sử một thuật toán ổn định ngược được áp dụng để giải một bài toán $f : X \to Y$ với số điều kiện κ trong một máy tính thỏa các tiên đề (3.2.5) và (3.2.7). Khi đó các sai số tương đối thỏa*

$$\frac{\|\tilde{f}(x) - f(x)\|}{\|f(x)\|} = O(\kappa(x)\epsilon_{machine}). \tag{3.3.13}$$

Chứng minh Theo Định nghĩa (3.3.5) của ổn định ngược, ta có $\tilde{f}(x) = f(\tilde{x})$ với $\tilde{x}$ nào đó thỏa

$$\frac{\|\tilde{x} - x\|}{\|x\|} = O(\epsilon_{machine}).$$

Theo Định nghĩa (3.1.5) của $\kappa(x)$, điều này kéo theo

$$\frac{\|\tilde{f}(x) - f(x)\|}{\|f(x)\|} \leqslant (\kappa(x) + o(1))\frac{\|\tilde{x} - x\|}{\|x\|}, \tag{3.3.14}$$

với $o(1)$ ký hiệu con số hội tụ tới 0 khi $\epsilon_{machine} \to 0$. Kết hợp các các chặn này thu được (3.3.13).

3.3.12 Phân tích sai số ngược

Quá trình mà ta đã thực hiện trong chứng minh Định lý 3.3.2 được biết như là *phân tích sai số ngược*. Ta thu được một ước lượng đúng đắn bằng hai bước. Bước đầu tiên là nghiên cứu điều kiện của bài toán. Bước còn lại là nghiên cứu tính ổn định của thuật toán. Kết luận của chúng ta là thuật toán ổn định, khi đó sự đúng đắn cuối cùng phản ánh số điều kiện đó.

Theo toán học, điều này là không phức tạp, nhưng nó chắc chắn không phải là ý tưởng đầu tiên cho phân tích một thuật toán số. Ý tưởng đầu tiên sẽ là *phân tích sai số tiến*. Các sai số làm tròn đưa ra tại mỗi bước của tính toán được ước lượng, và vì một lý do chưa xác định, một tổng được duy trì như thế nào khi chúng có thể kết hợp từng bước một.

Thực nghiệm đã cho thấy rằng hầu hết các thuật toán của phương pháp số trong đại số tuyến tính, phân tích sai số tiến là khó thực hiện hơn phân tích sai số ngược. Giả sử thuật toán sử dụng nhiều lần trong quá khứ và được chứng minh là đúng đắn được sử dụng, cụ thể, để giải $Ax = b$ trong một máy tính. Nó là một thiết lập mà các kết quả thu được sẽ được nhất quán nhỏ hơn sự đúng đắn khi A là bài toán điều kiện xấu. Bây giờ, phân tích sai số tiến có thể nắm bắt hiện tượng này như thế nào? Số điều kiện của A thấy được nhiều hơn hoặc ít hơn tại mức của các phép toán dấu chấm động không thấy được bao gồm trong việc giải $Ax = b$. Phân tích tiến sẽ phải tìm ra số điều kiện đó nếu nó kết thúc với một kết quả chính xác.

3.4 Tính ổn định của tam giác hóa Householder

3.4.1 Thực thi

Phân tích Householder là một thuật toán ngược cho việc tính toán các phân tích QR. Ta có thể miêu tả điều này bằng môt thực thi trong Matlab được thực hiện trong số học độ chính xác bội IEEE, $\epsilon_{machine} \approx 1.11 \times 10^{-16}$.

`R = triu(randn(50));`	Đặt R là ma trận tam giác trên 50×50 với các phần tử ngẫu nhiên thông thường.
`[Q, X] = qr(randn(50));`	Đặt Q là một ma trận trực giao ngẫu nhiên bằng việc trực giao hóa một ma trận ngẫu nhiên.
`A = Q*R;`	Đặt A là tích QR, tăng lên tới các sai số làm tròn.
`[Q2, R2] = qr(A);`	Tính phân tích QR $A \approx Q_2R_2$ bằng tam giác hóa Householder.

Mục đích của 4 dòng này của Matlab là để xây dựng một ma trận với phân tích QR đã biết, $A = QR$. Khi đó nó có thể được so sánh với phân tích QR $A = Q_2R_2$ được tính bằng tam giác hóa Householder. Trên thực tế, bởi vì các sai số làm tròn, các thừa số QR của ma trận tính được A không chính xác là Q và R .

Cho Q_2 và R_2,

```
norm(Q2-Q)
ans = 0.00889
norm(R2-R)/norm(R)
ans = 0.00071
```

Các tính toán của chúng ta đã được hoàn thành với 6 chữ số chính xác, các kết quả cuối cùng chưa là chính xác chỉ với 2 hoặc 3 chữ số. Các sai số làm tròn riêng lẻ đã được khuếch đại bằng các thừa số trong bậc 10^{-3}.

Ta dường như đã hao hụt 12 chữ số của sự đúng đắn. Nhân các ma trận không chính xác Q_2 và R_2 này:

```
norm(A - Q2*R2)/norm(A)
ans = 1.432e-15
```

Tích Q_2R_2 là chính xác tới 15 chữ số. Các sai số trong Q_2 và R_2 phải được "tương quan ranh mảnh", như Wilkinson thường nói. Để một phần trong 10^{12}, chúng có thể rút gọn trong tích Q_2R_2.

Để làm nổi bật sự đúng đắn của Q_2R_2 này là đặc biệt như thế nào, ta hãy xây dựng một cặp khác các ma trận Q_3 và R_3 mà chúng là các xấp xỉ chính xác của Q và R, và nhân chúng.

`Q3 = Q + 1e-4*randn(50);`	Đặt Q_3 là một nhiễu ngẫu nhiên của Q gần Q hơn Q_2.
`R3 = R + 1e-4*randn(50);`	Đặt R_3 là một nhiễu ngẫu nhiên của R gần R hơn R_2.
`norm(A - Q3*R3)/norm(A)`	Q_3R_3 chính xác như thế nào?
`ans = 0.00088`	

Lần này, sai số trong tích là lớn. Q_2 không tốt hơn Q_3, và R_2 không tốt hơn R_3, nhưng Q_2R_2 là bậc 12 tốt hơn Q_3R_3. Trong thực nghiệm này, ta không lấy phiền muộn để làm R_3 tam giác trên hoặc Q_3 trực giao, nhưng ta đã làm khác nhau một ít.

Các sai số trong Q_2 và R_2 là *các sai số tiến*. Tổng quát, một sai số tiến lớn có thể là kết quả của bài toán điều kiện xấu hoặc một thuật toán không ổn định (Định lý 3.3.2). Chuỗi của

các không gian cột của một ma trận tam giác ngẫu nhiên là điều kiện xấu cực kỳ như là một hàm các phần tử của ma trận.

Sai số trong Q_2R_2 là *sai số ngược* hoặc *thặng dư*. Tam giác hóa Householder là ổn định ngược.

3.4.2 Định lý

Tam giác hóa Householder là ổn định ngược cho mọi ma trận A và mọi máy tính thỏa (3.2.5) và (3.2.7).

Kết quả sẽ có dạng

$$\tilde{Q}\tilde{R} = A + \delta A, \tag{3.4.1}$$

với δA nhỏ. Mặt khác, tích của Q được tính với R được tính bằng với một nhiễu nhỏ của ma trận được cho A. Theo $\tilde{R}$, ma trận tam giác trên được xây dựng bằng tam giác hóa Gram - Schmidt trong số học dấu chấm động. Nhắc lại, $Q = Q_1Q_2 \dots Q_n$ (2.5.8), với Q_k là phản xạ Householder được xác định bởi vector v_k (2.5.5) được xác định tại bước thứ k của Thuật toán 2.3. Trong tính toán dấu chấm động, ta thu được một chuỗi các vector $\tilde{v_k}$. Cho $\tilde{Q_k}$ ký hiệu phản xạ *unita một cách chính xác* được xác định bởi vector dấu chấm động $\tilde{v_k}$ (theo toán học, không phải trong máy tính). Xác định

$$\tilde{Q} = \tilde{Q_1}\tilde{Q_2} \dots \tilde{Q_n}. \tag{3.4.2}$$

Ma trận unita một cách chính xác $\tilde{Q}$ này sẽ có vai trò đối với "Q tính được" của chúng ta. Trong các ứng dụng, như được thảo luận trong mục 2.5, ma trận Q nói chung không được tạo thành rõ ràng bằng bất kì cách nào, nên nó sẽ không hữu ích để xác định một "Q tính được" của nhiều dạng trước. Các vector $\tilde{v_k}$ được hình thành rõ ràng, và thiết lập như trong (3.4.2).

Dưới đây là định lý giải thích thực thi trong Matlab của chúng.

Định lý 3.4.1 *Cho phân tích QR $A = QR$ của một ma trận $A \in \mathbb{C}^{m\times n}$ được tính bởi tam giác hóa Householder (Thuật toán 2.3) trong một máy tính thỏa các tiên đề (3.2.5) và (3.2.7), và cho các thừa số tính được $\tilde{Q}$ và $\tilde{R}$ được xác định như được cho biết ở trên. Khi đó ta có*

$$\tilde{Q}\tilde{R} = A + \delta A, \ \frac{\|\delta A\|}{\|A\|} = O(\epsilon_{machine}) \tag{3.4.3}$$

với $\delta A \in \mathbb{C}^{m\times n}$ bất kì.

Biểu thức $O(\epsilon_{machine})$ trong (3.4.3) đã được thảo luận trong các mục 3.3. Chặn đúng khi $\epsilon_{machine} \to 0$, đồng nhất cho mọi ma trận A có số chiều m và n được cố định bất kì, nhưng không đồng nhất tương ứng với m và n. Bởi vì tất cả các chuẩn trong không gian hữu hạn chiều là tương đương nhau, ta không cần một chuẩn đặc biệt (Định lý 3.3.1).

3.4.3 Phân tích một thuật toán giải phương trình $Ax = b$

Ta đã thấy rằng tam giác hóa Householder là không ổn định ngược nhưng thường không đúng đắn trong hướng ngược. Sự đúng đắn của phân tích QR đủ cho hầu hết các mục tiêu. Ta có thể chứng minh điều này bằng các đối số đơn giản.

Ví dụ mà ta sẽ xét là sử dụng tam giác hóa Householder để giải hệ thống tuyến tính $m \times m$ không suy biến $Ax = b$. Ý tưởng này được thảo luận tại phần cuối của mục 2.2. Dưới đây là một phát biểu đầy đủ hơn của thuật toán đó.

Thuật toán này là ổn định ngượcỞ đây, ta sẽ phát biểu các kết quả ổn định ngược cho 3 bước (không chứng minh).

Thuật toán 3.1 Giải $Ax = b$ bằng phân tích QR

1: $QR = A$ Phân tích A thành QR bằng Thuật toán 2.3, với Q biểu diễn tích của các đối xứng.
2: $y = Q^*b$ Xây dựng Q^*b bằng Thuật toán 2.4
3: $x = R^{-1}y$ Giải hệ thống tam giác $Rx = y$ bằng phép thế ngược (Thuật toán 3.2).

Bước đầu tiên của Thuật toán 3.1 là phân tích QR của A, dẫn đến các ma trận $\tilde{R}$ và $\tilde{Q}$ được tính. Tính ổn định ngược của quá trình này đã được biểu diễn bởi (3.4.3).

Bước thứ hai là tính Q^*b bằng Thuật toán 2.4. Khi Q^*b được tính bằng Thuật toán 2.4 với các sai số làm tròn thì kết quả sẽ không là $\tilde{Q}^*b$. Nó sẽ là một vector $\tilde{y}$ nào đó và thỏa ước lượng tính ổn định ngược sau:

$$(\tilde{Q} + \delta Q)\tilde{y} = b,\ \|\delta Q\| = O(\epsilon_{machine}). \tag{3.4.4}$$

Giống như (3.4.3), đẳng thức này là chính xác. Mặt khác, kết quả việc áp dụng các phản xạ Householder trong số học dấu chấm động chính xác là tương đương với việc nhân b với một ma trận bị nhiễu nhỏ, $(\tilde{Q} + \delta Q)^{-1}$.

Bước cuối cùng của Thuật toán 3.1 là phép thế ngược để tính $\tilde{R}^{-1}\tilde{y}$. Trong bước này, các sai số làm tròn mới sẽ được đưa ra nhưng nhiều hơn 1 nên tính toán là ổn định ngược. Ước lượng này có dạng

$$(\tilde{R} + \delta R)\tilde{x} = \tilde{y},\ \frac{\|\delta R\|}{\|\tilde{R}\|} = O(\epsilon_{machine}). \tag{3.4.5}$$

Đẳng thức trong vế trái khẳng định rằng kết quả dấu chấm động $\tilde{x}$ là lời giải chính xác của một nhiễu nhỏ của hệ thống $\tilde{R}x = \tilde{y}$.

Định lý 3.4.2 *Thuật toán 3.1 là ổn định ngược, thỏa mãn*

$$(A + \Delta A)\tilde{x} = b,\ \frac{\|\Delta A\|}{\|A\|} = O(\epsilon_{machine}) \tag{3.4.6}$$

với $\Delta A \in \mathbb{C}^{m\times n}$ bất kì.

Chứng minh Kết hợp (3.4.4) và (3.4.3), ta có

$$b = (\tilde{Q} + \delta Q)(\tilde{R} + \delta R)\tilde{x} = [\tilde{Q}\tilde{R} + (\delta Q)\tilde{R} + \tilde{Q}(\delta R) + (\delta Q)(\delta R)]\tilde{x}.$$

Do đó, theo (3.4.3),

$$b = [A + \delta A + (\delta Q)\tilde{R} + \tilde{Q}(\delta R) + (\delta Q)(\delta R)]\tilde{x}.$$

Phương trình này có dạng

$$b = (A + \Delta A)\tilde{x},$$

với ΔA là tổng của 4 số hạng. Để ước lượng (3.4.6), ta phải cho thấy rằng mỗi số hạng này là tương đối nhỏ với A.

Vì $\tilde{Q}\tilde{R} = A + \delta A$ và $\tilde{Q}$ là unita nên ta có

$$\frac{\|\tilde{R}\|}{\|A\|} \leqslant \|\tilde{Q}^*\|\frac{\|A + \delta A\|}{\|A\|} = O(1)$$

khi $\epsilon_{machine} \to 0$, do (3.4.3). Do (3.4.4)nên

$$\frac{\|(\delta Q)\tilde{R}\|}{\|A\|} \leqslant \|\delta Q\|\frac{\|\tilde{R}\|}{\|A\|} = O(\epsilon_{machine})$$

Tương tự,

$$\frac{\|\tilde{Q}(\delta R)\|}{\|A\|} \leqslant \|\tilde{Q}\| \frac{\|\delta R\|}{\|\tilde{R}\|} \frac{\|\tilde{R}\|}{\|A\|}$$

do (3.4.5). Cuối cùng,

$$\frac{\|(\delta Q)(\delta R)\|}{\|A\|} \leqslant \|\delta Q\| \frac{\|\delta R\|}{\|A\|} = O(\epsilon_{machine}^2).$$

Khi đó tổng nhiễu ΔA thỏa mãn

$$\frac{\|\Delta A\|}{\|A\|} \leqslant \frac{\|\delta A\|}{\|A\|} + \frac{\|(\delta Q)\tilde{R}\|}{\|A\|} + \frac{\|\tilde{Q}(\delta R)\|}{\|A\|} + \frac{\|(\delta Q)(\delta R)\|}{\|A\|} = O(\epsilon_{machine}),$$

như được yêu cầu.

Kết hợp Định lý 3.1.2, 3.3.2, 3.4.2 và (3.4.2) cho kết quả theo sau về sự đúng đắn của các lời giải $Ax = b$.

Định lý 3.4.3 *Lời giải $\tilde{x}$ được tính bằng Thuật toán 3.1 thỏa mãn*

$$\frac{\|\tilde{x} - x\|}{\|x\|} = O(\kappa(A)\epsilon_{machine}). \tag{3.4.7}$$

3.5 Tính ổn định của phép thế ngược

3.5.1 Hệ thống tam giác

Một hệ thống tổng quát của các phương trình $Ax = b$ có thể được giảm xuống thành một hệ thống tam giác trên $Rx = b$ bằng phân tích QR. Các hệ thống tam giác dưới và tam giác trên cũng xuất hiện trong khử Gauss, trong phân tích Cholesky, và trong các tính toán số khác . Các hệ thống này dễ dàng được giải bằng một qúa trình của phép thế liên tiếp, gọi là *phép thế tiến* nếu hệ thống là tam giác dưới và *phép thế ngược* nếu hệ thống là tam giác trên. Mặc dù hai trường hợp là đồng nhất, cho tính xác định ta xét phép thế ngược trong mục này.

Giả sử ta mong muốn giải $Rx = b$,

$$\begin{bmatrix} r_{11} & r_{12} & \dots & r_{1m} \\ & r_{22} & & \\ & & \ddots & \vdots \\ & & & r_{mm} \end{bmatrix} \begin{bmatrix} x_1 \\ x_2 \\ \vdots \\ x_m \end{bmatrix} = \begin{bmatrix} b_1 \\ b_2 \\ \vdots \\ b_m \end{bmatrix}, \tag{3.5.1}$$

với $b \in \mathbb{C}^m$ và $R \in \mathbb{C}^{m \times m}$ là các ma trận không suy biến và tam giác trên được cho, và $x \in \mathbb{C}^m$ không được biết. Ta có thể làm điều này bằng việc giải các thành phần của x, bắt đầu với x_m và hoàn thành với x_1. Cho thuận lợi ta viết thuật toán như một chuỗi các công thức hơn là vòng lặp.

Cấu trúc là tam giác, với một phép trừ và một phép nhân tại mỗi vị trí. Do đó đếm số phép toán là hai lần diện tích tam giác $m \times m$:

$$\text{Phép thế ngược: } \sim m^2 \text{ phép toán dấu chấm động.} \tag{3.5.2}$$

Thuật toán 3.2 Phép thế ngược

1: $x_m = \frac{b_m}{r_{mm}}$

2: $x_{m-1} = \dfrac{(b_{m-1} - x_m r_{m-1,m})}{r_{m-1,m-1}}$

3: $x_{m-2} = \dfrac{(b_{m-2} - x_{m-1} r_{m-2,m-1} - x_m r_{m-2,m})}{r_{m-2,m-2}}$

4: $\vdots$

5: $x_j = \left(b_j - \sum_{k=j+1}^{m} x_k r_{jk}\right) \Big/ r_{jj}$

3.5.2 Định lý ổn định ngược

Phép thế ngược xuất hiện như một trong ba bước trong lời giải của $Ax = b$ bằng phân tích QR. Trong (3.4.3) - (3.4.5) ta khẳng định rằng mỗi bước này là ổn định ngược nhưng ta không chứng minh yêu cầu này.

Định lý 3.5.1 *Cho Thuật toán 3.2 được áp dụng cho bài toán (3.5.4) bao gồm các số dấu chấm động trong một máy tính thỏa (3.2.7). Thuật toán này là ổn định ngược mà lời giải tính được* $\tilde{x} \in \mathbb{C}^m$ *thỏa mãn*

$$(R + \delta R)\tilde{x} = b \tag{3.5.3}$$

cho tam giác trên $\delta R \in \mathbb{C}^{m\times m}$ *bất kì mà*

$$\frac{\|\delta R\|}{\|R\|} = O(\epsilon_{machine}). \tag{3.5.4}$$

Đặc biệt, với mỗi i, j,

$$\frac{|\delta r_{ij}|}{|r_{ij}|} \leqslant m\epsilon_{machine} + O(\epsilon^2_{machine}). \tag{3.5.5}$$

Trong (3.5.5) và thông qua mục này, ta tiếp tục sử dụng quy ước của (3.3.12) mà nếu mẫu số bằng 0, thì tử số cũng được khẳng định bằng 0 (với mọi $\epsilon_{machine}$ đủ nhỏ).

3.5.3 $m = 1$

Theo (3.5.3), công việc của chúng ta là biểu diễn mọi sai số dấu chấm động như là một nhiễu của dữ liệu đầu vào. Ta bắt đầu với trường hợp đơn giản nhất mà R có số chiều là 1×1. Phép thế ngược trong trường hợp này bao gồm một bước,

$$\tilde{x_1} = b_1 \oslash r_{11}.$$

Tiên đề (3.2.7) cho $\oslash$ đảm bảo rằng lời giải tính được là gần đúng:

$$\tilde{x_1} = \frac{b_1}{r_{11}}(1 + \epsilon_1),\ |\epsilon_1 \leqslant \epsilon_{machine}. \tag{3.5.6}$$

Tuy nhiên, ta muốn biểu diễn sai số nếu như nó đưa ra kết quả từ một nhiễu trong R. Để kết thúc điều này, ta đặt $\epsilon_1^{'} = -\dfrac{\epsilon_1}{1 + \epsilon_1}$, công thức ở trên trở thành

$$\tilde{x_1} = \frac{b_1}{r_{11}(1 + \epsilon_1^{'})},\ |\epsilon_1^{'}| \leqslant \epsilon_{machine} + O(\epsilon^2_{machine}). \tag{3.5.7}$$

Chú ý rằng $\epsilon_1^{'}$ bằng $-\epsilon_1$ cộng với một số hạng bậc ϵ_1^2. Ta có thể tự do di chuyển các nhiễu tương đối nhỏ từ các tử số tới các mẫu số hoặc ngược lại, và kết quả thay đổi bằng các số hạng bậc $\epsilon_{machine}^2$.

Trong (3.5.7), đẳng thức là chính xác; phép chia thuộc về toán học chứ không phải là dấu chấm động. Công thức phát biểu rằng phép thế ngược 1×1 là ổn định ngược, với $\tilde{x_1}$ là lời giải chính xác tới một bài toán bị nhiễu, cụ thể

$$(r_{11} + \delta r_{11})\tilde{x_1} = b_1,$$

với $\delta r_{11} = \epsilon_1^{'} r_{11}$. Do đó

$$\frac{|\delta r_{11}|}{|r_{11}|} \leqslant \epsilon_{machine} + O(\epsilon_{machine}^2).$$

3.5.4 $m = 2$

Giả sử ta có một ma trận tam giác trên $R \in \mathbb{C}^{2\times 2}$ và một vector $b \in \mathbb{C}^2$. Tính $\tilde{x} \in \mathbb{C}^2$ tiến hành trong hai bước. Đầu tiên là giống như trong trường hợp 1×1:

$$\tilde{x_2} = b_2 \oslash r_{22} = \frac{b_2}{r_{22}(1+\epsilon_1)},\ |\epsilon_1| \leqslant \epsilon_{machine} + O(\epsilon_{machine}^2). \tag{3.5.8}$$

Bước thứ hai được xác định bởi công thức

$$\tilde{x_1} = (b_1 \ominus (\tilde{x_2} \otimes r_{12})) \oslash r_{11}.$$

Để thiết lặp ổn định ngược, ta phải biểu diễn các sai số trong 3 phép toán dấu chấm động này như các nhiễu trong các phần tử r_{ij}.

Ta sử dụng tiên đề (3.2.7) để giải thích phép nhân dấu chấm động như là một nhiễu trong r_{12}:

$$\tilde{x_1} = (b_1 \ominus \tilde{x_2} r_{12}(1+\epsilon_2)) \oslash r_{11},\ |\epsilon_2| \leqslant \epsilon_{machine}.$$

Đầu tiên, ta viết công thức với toán học chính xác theo (3.2.7):

$$\tilde{x_1} = (b_1 - \tilde{x_2} r_{12}(1+\epsilon_2))(1+\epsilon_3) \oslash r_{11} \tag{3.5.9}$$

$$= \frac{(b_1 - \tilde{x_2} r_{12}(1+\epsilon_2))(1+\epsilon_3)}{r_{11}}(1+\epsilon_4). \tag{3.5.10}$$

Ở đây (3.5.8) đảm bảo $|\epsilon_3||\epsilon_4| \leqslant \epsilon_{machine}$. Ta đổi chỗ các số hạng ϵ_3 và ϵ_4 từ tử số thành mẫu số. Điều này đưa ra

$$\tilde{x_1} = \frac{b_1 - \tilde{x_2} r_{12}(1+\epsilon_2)}{r_{11}(1+\epsilon_3^{'})(1+\epsilon_4^{'})},$$

với $|\epsilon_3^{'}|, |\epsilon_4^{'}| \leqslant \epsilon_{machine} + O(\epsilon_{machine}^2)$, hoặc tương đương

$$\tilde{x_1} = \frac{b_1 - \tilde{x_2} r_{12}(1+\epsilon_2)}{r_{11}(1+2\epsilon_5)}, \tag{3.5.11}$$

với $|\epsilon_5| \leqslant \epsilon_{machine} + O(\epsilon_{machine}^2)$. Công thức này phát biểu rằng $\tilde{x_1}$ sẽ là chính xác nếu r_{22}, r_{12} và r_{11} được làm nhiễu bởi các thừa số tương ứng $(1+\epsilon_1), (1+\epsilon_2)$ và $(1+2\epsilon_5)$. Các nhiễu này có thể được tóm tắt bằng phương trình

$$(R + \delta R)\tilde{x} = b,$$

với các phần tử δr_{ij} của δR thỏa mãn

$$\begin{bmatrix} |\delta r_{11}|/|r_{11}| & |\delta r_{12}|/|r_{12}| \\ & |\delta r_{22}|/|r_{22}| \end{bmatrix} = \begin{bmatrix} 2|\epsilon_5| & |\epsilon_2| \\ & |\epsilon_1| \end{bmatrix} \leqslant \begin{bmatrix} 2 & 1 \\ & 1 \end{bmatrix} \epsilon_{machine} + O(\epsilon_{machine}^2).$$

Công thức này đảm bảo $\|\delta R\|/\|R\| = O(\epsilon_{machine})$ trong chuẩn ma trận bất kỳ và do đó phép thế ngược 2×2 đó là ổn định ngược.

3.5.5 $m = 3$

Phân tích cho một ma trận 3×3 bao gồm tất cả lý do cần thiết cho trường hợp tổng quát. Đầu tiên, 2 bước là giống như trước:

$$\tilde{x_3} = b_3 \oslash r_{33} = \frac{b_3}{r_{33}(1+\epsilon_1)}, \tag{3.5.12}$$

$$\tilde{x_2} = (b_2 \ominus (\tilde{x_3} \otimes r_{23})) \oslash r_{22} = \frac{b_2 - \tilde{x_3} r_{23}(1+\epsilon_2)}{r_{22}(1+2\epsilon_3)}, \tag{3.5.13}$$

với

$$\begin{bmatrix} 2|\epsilon_3| & |\epsilon_2| \\ & |\epsilon_1| \end{bmatrix} \leqslant \begin{bmatrix} 2 & 1 \\ & 1 \end{bmatrix} \epsilon_{machine} + O(\epsilon_{machine}^2).$$

Bước thứ ba bao gồm tính toán

$$\tilde{x_1} = [(b_1 \ominus (\tilde{x_2} \otimes r_{12})) \ominus (\tilde{x_3} \otimes r_{13})] \oslash r_{11}. \tag{3.5.14}$$

Ta biến đổi hai phép toán $\otimes$ trong (3.5.14) thành phép nhân toán học bằng việc đưa ra các nhiễu ϵ_4 và ϵ_5

$$\tilde{x_1} = [(b_1 \ominus \tilde{x_2} r_{12}(1+\epsilon_4)) \ominus \tilde{x_3} r_{13}(1+\epsilon_5)] \oslash r_{11}.$$

Ta biến đổi các phép toán $\ominus$ thành các phép trừ toán học thông qua các nhiễu ϵ_6 và ϵ_7:

$$\tilde{x_1} = [(b_1 - \tilde{x_2} r_{12}(1+\epsilon_4))(1+\epsilon_6) - \tilde{x_3} r_{13}(1+\epsilon_5)](1+\epsilon_7) \oslash r_{11}.$$

Cuối cùng, $\oslash$ được ước lượng bằng việc sử dụng ϵ_8. Ta hãy thay thế điều này bằng $\epsilon_8^{'}$ với $|\epsilon_8| \leqslant \epsilon_{machine} + O(\epsilon_{machine})$ và đặt kết quả trong mẫu số:

$$\tilde{x_1} = \frac{[(b_1 - \tilde{x_2} r_{12}(1+\epsilon_4))(1+\epsilon_6) - \tilde{x_3} r_{13}(1+\epsilon_5)](1+\epsilon_7)}{r_{11}(1+\epsilon_8^{'})}.$$

Ta đổi ϵ_7 thành $\epsilon_7^{'}$ và di chuyển nó tới mẫu số như thường dùng. Số hạng bao gồm ϵ_6 yêu cầu một thủ thuật mới. Ta di chuyển nó thành mẫu số như vậy, nhưng để giữ đẳng thức hợp lệ, ta làm cân bằng bằng việc đặt một thừa số mới $(1+\epsilon_6^{'})$ vào số hạng r_{13}. Khi đó

$$\tilde{x_1} = \frac{b_1 - \tilde{x_2} r_{12}(1+\epsilon_4) - \tilde{x_3} r_{13}(1+\epsilon_5)(1+\epsilon_6^{'})}{r_{11}(1+\epsilon_6^{'})(1+\epsilon_7^{'})(1+\epsilon_8^{'})}.$$

Bây giờ r_{13} có 2 nhiễu của kích thước nhiều nhất là $\epsilon_{machine}$, và r_{11} có ba nhiễu. Trong công thức này, tất cả các sai số trong phép tính đã được biểu diễn như các nhiễu trong các phần tử của R.

Kết quả có thể được tóm tắt như

$$(R + \delta R)\tilde{x} = b,$$

với các phần tử δr_{ij} thỏa mãn

$$\begin{bmatrix} |\delta r_{11}|/|r_{11}| & |\delta r_{12}|/|r_{12}| & |\delta r_{13}|/|r_{13}| \\ & |\delta r_{22}|/|r_{22}| & |\delta r_{23}|/|r_{23}| \\ & & |\delta r_{33}|/|r_{33}| \end{bmatrix} \leqslant \begin{bmatrix} 3 & 1 & 2 \\ & 2 & 1 \\ & & 1 \end{bmatrix} \epsilon_{machine} + O(\epsilon_{machine}^2).$$

3.5.6 m tổng quát

Phân tích trong các trường hợp số chiều cao hơn là tương tự. Ví dụ, trong trường hợp 5×5 ta thu được chặn

$$\frac{|\delta R|}{|R|} \leqslant \begin{bmatrix} 5 & 1 & 2 & 3 & 4 \\ & 4 & 1 & 2 & 3 \\ & & 3 & 1 & 2 \\ & & & & 1 \end{bmatrix} \epsilon_{machine} + O(\epsilon_{machine}^2). \tag{3.5.15}$$

Các phần tử của ma trận trong công thức này thu được từ 3 thành phần. Các phép nhân $\tilde{x_k} r_{jk}$ đưa ra các nhiễu $\epsilon_{machine}$ trong kiểu

$$\otimes : \begin{bmatrix} 0 & 1 & 1 & 1 & 1 \\ & 0 & 1 & 1 & 1 \\ & & 0 & 1 & 1 \\ & & & 0 & 1 \\ & & & & 0 \end{bmatrix}. \tag{3.5.16}$$

Các phép chia cho r_{kk} đưa ra các nhiễu trong kiểu

$$\oslash : \begin{bmatrix} 1 & & & & \\ & 1 & & & \\ & & 1 & & \\ & & & 1 & \\ & & & & 1 \end{bmatrix}. \tag{3.5.17}$$

Cuối cùng, phép trừ cũng xuất hiện trong kiểu (3.5.16). Do quyết định tính toán từ trái sang phải nên mỗi phép trừ đưa ra một nhiễu trên đường chéo và tại mỗi vị trí tới bên phải. Điều này tăng thêm lên thành kiểu

$$\ominus : \begin{bmatrix} 4 & 0 & 1 & 2 & 3 \\ & 3 & 0 & 1 & 2 \\ & & 2 & 0 & 1 \\ & & & 1 & 0 \\ & & & & 0 \end{bmatrix}. \tag{3.5.18}$$

Thêm (3.5.16), (3.5.17) và (3.5.18) đưa ra kết quả trong (3.5.15). Điều này hoàn thành chứng minh của Định lý 3.5.1.

3.6 Quy định của các bài toán bình phương nhỏ nhất

3.6.1 Bốn bài toán quy định

Trong mục này ta quay lại bài toán bình phương nhỏ nhất tuyến tính (2.6.2), được minh họa lại như trong Hình 3.2. Giả sử ma trận xác định bài toán có hạng đầy đủ, và viết $\|\cdot\| = \|\cdot\|_2$:

Cho $A \in \mathbb{C}^{m \times n}$có hạng đầy đủ, $m \geqslant n, b \in \mathbb{C}^m$, tìm $x \in \mathbb{C}^n$ sao cho $\|b - Ax\|$ được cực tiểu hóa. (3.6.1)

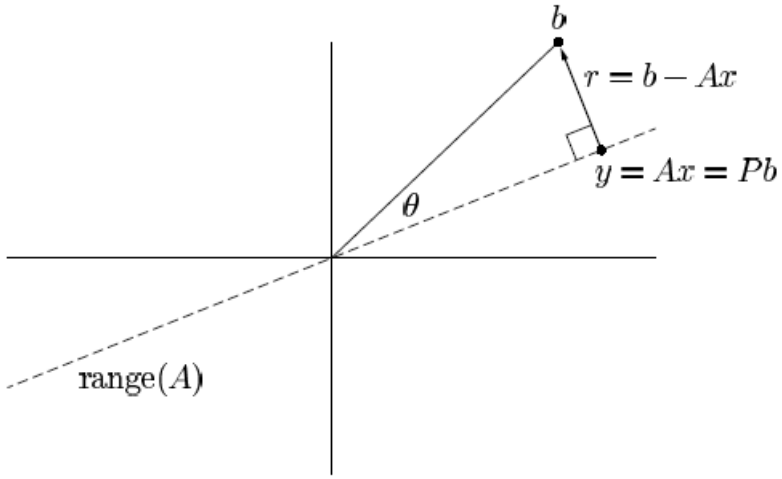
Lời giải x và tương ứng $y = Ax$ là gần b nhất trong $range(A)$ được cho bởi

$$x = A^+ b, \; y = Pb, \tag{3.6.2}$$

với $A^+ \in \mathbb{C}^{n \times m}$ là giả nghịch đảo của A (2.6.11) và $P = AA^+ \in \mathbb{C}^{m \times m}$ là phép chiếu trực giao lên trên $range(A)$.

Quy định liên quan với độ nhạy của các lời giải tới các nhiễu trong dữ liệu. Cho (3.6.1), ta sẽ khảo sát hai lựa chọn của mỗi dữ liệu. Dữ liệu cho bài toán là ma trận A có $m \times n$ chiều và vector b có m chiều. Lời giải là vector hệ số x hoặc điểm $y = Ax$ tương ứng. Do đó

$$\text{Dữ liệu:} A, b, \quad \text{Lời giải:} x, y.$$



Hình 3.2: Bài toán bình phương nhỏ nhất

3.6.2 Định lý

Đầu tiên là số điều kiện của A. Cho một ma trận vuông, đó là là $\kappa(A) = \|A\|\|A^{-1}\|$, và trong trường hợp hình chữ nhật, định nghĩa tổng quát hóa (3.1.18),

$$\kappa(A) = \|A\|\|A^{+}\| = \frac{\sigma_1}{\sigma_n}. \tag{3.6.3}$$

Thứ hai là góc θ như trong Hình 3.2, một độ đo của tính chính xác của sự điều chỉnh cho vừa:

$$\theta = \cos^{-1}\frac{\|y\|}{\|b\|}. \tag{3.6.4}$$

Thứ ba là độ đo của $\|y\|$ bao nhiêu để hạ xuống giá trị có thể lớn nhất của nó, $\|A\|$ và $\|x\|$ được cho:

$$\eta = \frac{\|A\|\|x\|}{\|y\|} = \frac{\|A\|\|x\|}{\|Ax\|}. \tag{3.6.5}$$

Các tham số này nằm trong

$$1 \leqslant \kappa(A) < \infty,\ 0 \leqslant \theta \leqslant \pi/2,\ 1 \leqslant \eta \leqslant \kappa(A). \tag{3.6.6}$$

Định lý 3.6.1 *Cho $b \in \mathbb{C}^m$ và $A \in \mathbb{C}^{m\times n}$ có hạng đầy đủ được cố định. Bài toán bình phương nhỏ nhất (3.6.1) có các số điều kiện tương đối trong chuẩn 2 theo sau (3.1.5) miêu tả các độ nhạy của y và x tới các nhiễu trong b và A:*

Các kết quả trong dòng đầu tiên là chính xác cho các nhiễu δb nào đó và các kết quả trong dòng thứ hai là các chặn trên.

	y	x
b	$\frac{1}{\cos\theta}$	$\frac{\kappa(A)}{\eta\cos\theta}$
A	$\frac{\kappa(A)}{\cos\theta}$	$\kappa(A)+\frac{\kappa(A)^2\tan\theta}{\eta}$

Trong trường hợp đặc biệt $m = n$, (3.6.1) giảm xuống thành một hệ thống các phương trình không suy biến, vuông với $\theta = 0$. Trong trường hợp này, các số trong cột thứ 2 của định lý giảm xuống thành $\kappa(A)/\eta$ và $\kappa(A)$ mà chúng là các kết quả trong (3.1.14) và (3.1.19) suy ra dễ dàng hơn, và số trong vị trí bên trái thấp hơn có thể được thay thế bằng 0.

3.6.3 Biến đổi thành một ma trận đường chéo

Cho A có một SVD dạng $A = U\sum V^*$, với $\sum$ là ma trận đường chéo $m \times n$ với các phần tử trên đường chéo dương. Vì các nhiễu là đo được trong chuẩn 2, các kích thước của chúng không bị tác động bởi một thay đổi cơ sở Unita, nên cách xử lý nhiễu của A là giống như của $\sum$. Do đó, không mất tính tổng quát, ta có thể giải quyết $\sum$ một cách trực tiếp. Cho phần còn lại của thảo luận, ta giả sử $A = \sum$ và viết

$$A = \begin{bmatrix} \sigma_1 & & & \\ & \sigma_2 & & \\ & & \ddots & \\ & & & \sigma_n \\ & & & \end{bmatrix} = \begin{bmatrix} A_1 \\ 0 \end{bmatrix}. \tag{3.6.7}$$

Ở đây A_1 là ma trận đường chéo có $n \times n$ chiều, các phần tử còn lại của A là 0.

Phép chiếu trực giao của b lên trên $range(A)$ bây giờ là không tầm thường. Viết

$$b = \begin{bmatrix} b_1 \\ b_2 \end{bmatrix},$$

với b_1 chứa n phần tử đầu tiên của b. Khi đó phép chiếu $y = Pb$ là

$$y = \begin{bmatrix} b_1 \\ 0 \end{bmatrix}.$$

Để tìm x tương ứng ta có thể viết $Ax = y$ như

$$\begin{bmatrix} A_1 \\ 0 \end{bmatrix} x = \begin{bmatrix} b_1 \\ 0 \end{bmatrix},$$

mà nó suy ra

$$x = A_1^{-1} b_1. \tag{3.6.8}$$

Từ các công thức này, phép chiếu trực giao và giả nghịch đảo là các ma trận khối 2×2 và 1×2.

$$P = \begin{bmatrix} I & 0 \\ 0 & 0 \end{bmatrix}, \; A^+ = [A_1^{-1} \quad 0]. \tag{3.6.9}$$

3.6.4 Độ nhạy của y tới các nhiễu trong b

Ta bắt đầu với 4 kết quả quy định đơn giản nhất. Do (3.6.2), các quan hệ giữa b và y chỉ là phương trình tuyến tính $y = Pb$. Ánh xạ Jacobi này là P vào chính nó, với $\|P\| = 1$ theo (3.6.9). Do (3.1.6) và (3.6.4), số điều kiện của y tương ứng với các nhiễu trong b là

$$\kappa_{b\mapsto y} = \frac{\|P\|}{\|y\|/\|b\|} = \frac{1}{\cos\theta}.$$

Điều này thiết lặp kết quả ở bên trái phía trên của Định lý 3.6.1. Số điều kiện được thực hiện cho các nhiễu δb với $\|P(\delta b)\| = \|\delta b\|$, mà nó xuất hiện khi δb bằng 0 ngoại trừ n phần tử đầu tiên.

3.6.5 Độ nhạy của x tới các nhiễu trong b

Quan hệ giữa b và x cũng là tuyến tính, $x = A^+b$, với Jacobian A^+. Do (3.1.6), (3.6.4) và (3.6.5), số điều kiện của x tương ứng với các nhiễu trong b là

$$\kappa_{b\mapsto x} = \frac{\|A^+\|}{\|x\|/\|b\|} = \|A^+\|\frac{\|b\|}{\|y\|}\frac{\|y\|}{\|x\|} = \|A^+\|\frac{1}{\cos\theta}\frac{\|A\|}{\eta} = \frac{\kappa(A)}{\eta\cos\theta}.$$

Điều này thiết lặp kết quả ở bên phải phía trên của Định lý 3.6.1. Số điều kiện của x được thực hiện bởi các nhiễu δb thỏa mãn $\|A^+(\delta b)\| = \|A^+\|\|\delta b\| = \|\delta b\|/\sigma_n$, mà nó xuất hiện khi δb bằng 0 ngoại trừ trong phần tử thứ n (hoặc cũng có thể trong các phần tử khác, nếu A có nhiều một giá trị suy biến bằng σ_n).

3.6.6 Độ dốc range của A

Phân tích các nhiễu trong A là bài toán không tuyến tính và tinh vi hơn. Ta sẽ tiếp tục bằng việc tính các Jacobi theo phương pháp đại số. Đầu tiên ta quan sát các nhiễu trong A làm ảnh hưởng tới bài toán bình phương nhỏ nhất trong 2 cách: chúng làm biến dạng ánh xạ của $\mathbb{C}^n$ lên trên $range(A)$, và chúng làm thay đổi $range(A)$.

Ta có thể hình dung các thay đổi nhỏ trong $range(A)$ như "các độ dốc" nhỏ của không gian này. Góc lớn nhất của độ dốc $\delta\alpha$ mà nó có thể tác động bằng một nhiễu nhỏ δA được xác định như sau. Ảnh dưới A của quả cầu đơn vị n chiều là một siêu ellip mà nó nằm trong $range(A)$. Để thay đổi $range(A)$ hiệu quả như có thể thực hiện được, ta nắm được một điểm $p = Av$ trong siêu ellip (do đó $\|v\| = 1$) và nhích nó trong một phương δp trực giao với $range(A)$. Một ma trận nhiễu mà nó đạt được hầu hết một cách hiệu là $\delta A = (\delta p)v^*$, mà nó cho $(\delta A)v = \delta p$ với $\|\delta A\| = \|\delta p\|$. Bây giờ rõ ràng rằng để thu được độ dốc lớn nhất với một $\|\delta p\|$ được cho, ta sẽ lấy p là gần với gốc như có thể thực hiện được. Đó là, ta muốn $p = \sigma_n u_n$, với σ_n là giá trị suy biến nhỏ nhất của A và u_n là vector suy biến trái tương ứng. Với A trong dạng đường chéo (3.6.7), p là bằng với cột cuối cùng của A, v^* là vector n chiều $(0, 0, \ldots, 0, 1)$, và δA là một nhiễu của các phần tử của A bên dưới đường chéo trong cột này. Một nhiễu như vậy làm dốc $range(A)$ bằng góc $\delta\alpha$ được cho bởi $\tan(\delta\alpha) = \|\delta p\|/\sigma_n$. Vì $\|\delta p\| = \|\delta A\|$ và $\delta\alpha \leqslant \tan(\delta\alpha)$, ta có

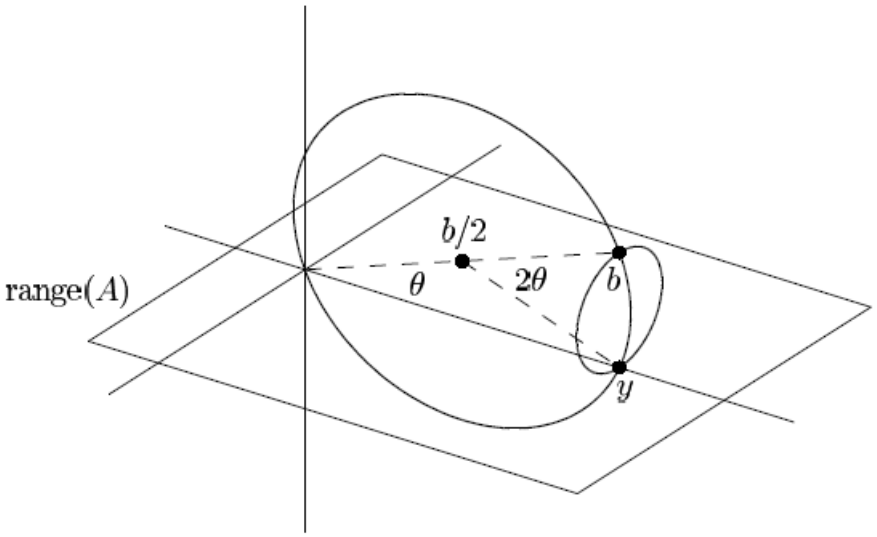
$$\delta\alpha \leqslant \frac{\|\delta A\|}{\sigma_n} = \frac{\|\delta A\|}{\|A\|}\kappa(A), \tag{3.6.10}$$

với đẳng thức thu được cho các lựa chọn δA của loại vừa được miêu tả, chứng minh chúng là nhỏ vô cùng (để $\delta\alpha = \tan(\delta\alpha)$).

3.6.7 Độ nhạy của y tới các nhiễu trong A

Bây giờ ta được chuẩn bị để suy ra dòng thứ hai của bảng trong Định lý 3.6.1. Ta bắt đầu với phần tử bên trái của nó. Vì y là phép chiếu trực giao của b lên trên $range(A)$, nó được xác định bởi b và một mình $range(A)$. Do đó, để phân tích độ nhạy của y tới các nhiễu trong A, ta có thể nghiên cứu một cách đơn giản hiệu quả trong y của độ dốc $range(A)$ bằng góc $\delta\alpha$ nào đó.

Một tính chất hình học tao nhã xuất hiện khi ta cho rằng b cố định và xem y biến đổi



Hình 3.3: Hai đường tròn trong hình cầu dọc theo mà y di chuyển khi $range(A)$ thay đổi. Đường tròn lớn, bán kính $\|b\|/2$, tương ứng với độ dốc $range(A)$ trong mặt phẳng $0 - b - y$, và đường tròn nhỏ, bán kính $(\|b\|/2)\sin\theta$, tương ứng với độ dốc của nó trong một phương trực giao. Tuy nhiên $range(A)$ bị dốc, y còn lại trong hình cầu bán kính $\|b\|/2$ có tâm tại $b/2$

như $range(A)$ bị dốc (Hình 3.3). Cho dù như thế nào $range(A)$ bị dốc như thế nào, vector $y \in range(A)$ thường phải là trực giao với $y - b$. Đó là, đường $y - b$ phải nằm tại góc bên phải với đường $0 - y$. Mặt khác, khi $range(A)$ được điều chỉnh, y di chuyển dọc theo quả cầu bán kính $\|b\|/2$ đặt tâm tại điểm $b/2$.

Độ dốc $range(A)$ trong mặt phẳng $0 - b - y$ bởi góc $\delta\alpha$ thay đổi thành góc 2θ tại tâm $b/2$ bằng $2\delta\alpha$. Do đó nhiễu tương ứng δy là cơ sở của một tam giác cân với góc ở giữa $2\delta\alpha$ và độ dài cạnh $\|b\|/2$. Điều này kéo theo $\|\delta y\| = \|b\|\sin(\delta\alpha)$. Độ dốc $range(A)$ trong phương khác bất kỳ đưa ra kết quả trong hình học tương tự trong một mặt phẳng khác và các nhiễu nhỏ hơn bằng một thừa số nhỏ như $\sin\theta$. Do đó cho các nhiễu bất kỳ bằng một góc $\delta\alpha$ ta có

$$\|\delta y\| \leqslant \|b\|\sin(\delta\alpha) \leqslant \|b\|\delta\alpha. \tag{3.6.11}$$

Do (3.6.4) và (3.6.10), điều này cho chúng ta $\|\delta y\| \leqslant \|\delta A\|\kappa(A)\|y\|/\|A\|\cos\theta$, đó là,

$$\frac{\|\delta y\|}{\|y\|} \Big/ \frac{\|\delta A\|}{\|A\|} \leqslant \frac{\kappa(A)}{\cos\theta}. \tag{3.6.12}$$

Điều này thiết lập kết quả bên trái phía dưới hơn của Định lý 3.6.1.

3.6.8 Độ nhạy của x tới các nhiễu trong A

Bây giờ ta sẵn sàng để phân tích quan hệ thú vị nhất của Định lý 3.6.1: độ nhạy của x tới các nhiễu trong A.

Một nhiễu δA làm dốc tự nhiên thành 2 phần: một phần δA_1 trong n dòng đầu tiên của A, và phần khác δA_2 trong $m-n$ dòng còn lại:

$$\delta A = \begin{bmatrix} \delta A_1 \\ \delta A_2 \end{bmatrix} = \begin{bmatrix} \delta A_1 \\ 0 \end{bmatrix} + \begin{bmatrix} 0 \\ \delta A_2 \end{bmatrix}$$

Đầu tiên, ta hãy xét hiệu quả của các nhiễu δA_1. Một nhiễu như vậy thay đổi ánh xạ của A trong phạm vi của nó, nhưng không $range(A)$ vào chính nó hoặc y. Nó làm nhiễu A_1 bằng δA_1 trong hệ thống vuông (3.6.8) không thay đổi b_1. Số điều kiện cho các nhiễu như vậy được cho bởi (3.1.19), có dạng

$$\frac{\|\delta x\|}{\|x\|} \Big/ \frac{\|\delta A_1\|}{\|A\|} \leqslant \kappa(A_1) = \kappa(A). \tag{3.6.13}$$

Tiếp theo ta xét hiệu quả của các nhiễu δA_2 (nhỏ vô cùng). Một nhiễu như vậy làm dốc $range(A)$ không thay đổi việc ánh xạ A trong không gian này. Điểm y và do đó vector b_1 được làm nhiễu, nhưng không là A_1. Số điều kiện cho các nhiễu như vậy được cho bởi (3.1.15), có dạng

$$\frac{\|\delta x\|}{\|x\|} \Big/ \frac{\|\delta b_1\|}{\|b_1\|} \leqslant \frac{\kappa(A_1)}{\eta(A_1; x)} = \frac{\kappa(A)}{\eta}. \tag{3.6.14}$$

Để hoàn thành đối số ta cần để liên kết δb_1 với δA_2. Bây giờ vector b_1 là y được biểu diễn trong các tọa độ của $range(A)$. Do đó, chỉ các thay đổi trong y được thực hiện như thay đổi trong b_1 là nằm song song với $range(A)$; các thay đổi trực giao không có hiệu quả. Đặc biệt, nếu $range(A)$ bị làm dốc bởi môt góc $\delta\alpha$ trong mặt phẳng $0-b-y$, nhiễu kết quả δy không nằm song song với $range(A)$ nhưng tại một góc $\pi/2-\theta$. Do đó, thay đổi trong b_1 thỏa mãn $\|\delta b_1\| = \sin\theta\|\delta y\|$. Theo (3.6.11), do đó ta có

$$\|\delta b_1\| \leqslant (\|b\|\delta\alpha)\sin\theta. \tag{3.6.15}$$

Lạ thật, nếu $range(A)$ được làm dốc trong một phương trực giao với mặt phẳng $0-b-y$, ta thu được chặn giống nhau, nhưng cho một lý do khác. Bây giờ δy là song song với $range(A)$, nhưng nó là một thừa số của $\sin\theta$ nhỏ hơn, như được miêu tả ở trên trong sự kết nối với Hình 3.3. Do đó ta có $\|\delta y\| \leqslant (\|b\|\delta\alpha)\sin\theta$, và vì $\|\delta b_1\| \leqslant \|\delta y\|$, ta lại được (3.6.15).

Bây giờ tất cả các phần nằm trong nơi này. Vì $\|b_1\| = \|b\|\cos\theta$, ta có thể viết lại (3.6.15) như

$$\frac{\|\delta b_1\|}{\|b_1\|} \leqslant (\delta\alpha)\tan\theta. \tag{3.6.16}$$

Liên kết $\delta\alpha$ với $\|\delta A_2\|$ theo (3.6.10) và kết hợp (3.6.14) với (3.6.16), ta được

$$\frac{\|\delta x\|}{\|x\|} \Big/ \frac{\|\delta A_2\|}{\|A\|} \leqslant \frac{\kappa(A_1)^2\tan\theta}{\eta}.$$

Thêm điều này vào (3.6.13) thiết lặp kết quả bên phải ở phía dưới hơn của Định lý 3.6.1.

3.7 Tính ổn định của các thuật toán bình phương nhỏ nhất

3.7.1 Ví dụ

Xét ví dụ số với $m = 100, n = 15$. Dưới đây là một cài đặt trong Matlab:

```
m=100; n=15;
t = (0:m-1)'/(m-1);          Đặt t là một sự rời rạc hóa của [0,1].
A = []; for i = 1:n,         Xây dựng ma trận Vandermonde.
 A =[A t.^(i-1)]; end
b=exp(sin(4*t));             Vế bên phải.
b=b/2006.787453080206;       Trực chuẩn hóa.
```

Ý tưởng ở đằng sau ví dụ này là điều chỉnh bình phương nhỏ nhất của hàm $exp(\sin(4\tau))$ trong khoảng [0,1] bằng một đa thức bậc 14. Đầu tiên ta rời rạc hóa [0,1], xác định một vector t của 100 điểm cách quãng bằng nhau từ 0 tới 1. Ma trận A là ma trận Vandermonde 100×15 mà các cột của nó là các lũy thừa $1, \tau, \ldots, \tau^{14}$ giống nhau tại các điểm của t, và vế bên phải b là hàm $exp(\sin(4\tau))$ giống nhau tại các điểm này.

Lý do cho dòng cuối cùng kỳ lạ của code là như sau. Cho đơn giản, ta sẽ chỉ so sánh các hệ số x_{15} được tính toán bằng các thuật toán khác nhau. Không có dòng cuối cùng này, giá trị chính xác của x_{15} sẽ là 2006.787453080206... (hình này thu đươc với một gói số học được mở rộng đúng đắn). Bằng việc chia cho con số này, ta thu được một bài toán mà lời giải có $x_{15} = 1$ để so sánh các thuật toán dễ hơn.

Ta sẽ cần 3.6.3 - 3.6.5 để xác định các con số này cho sự đúng đắn thỏa mãn bằng việc giải bài toán bình phương nhỏ nhất với sự hỗ trợ của Matlab:

```
x = A\b; y = A*x;            Giải bài toán bình phương nhỏ nhất.
kappa = cond(A)
    kappa = 2.2718e+10       κ(A)
theta = asin(norm(b-y)/norm(b))
    theta = 3.7461e-06       θ
eta = norm(A)*norm(x)/norm(y)
    eta = 2.1036e+05         η
```

Kết quả $\kappa(A) \approx 10^{10}$ cho thấy rằng các đơn thức $1, t, \ldots, t^{14}$ tạo thành một cơ sở có điều kiện xấu cao. Kết quả $\theta \approx 10^{-6}$ cho thấy rằng $exp(\sin(4t))$ có thể được điều chỉnh cho rất gần nhau bởi một đa thức bậc 14. Khi cho η, các giá trị của nó khoảng 10^5 là khoảng giữa 1 và $\kappa(A)$ được cho phép bởi (3.6.6).

Việc thêm vào các số này vào các công thức của Định lý 3.6.1, các số điều kiện của y và x tương ứng với các nhiễu trong b và A là xấp xỉ

	y	x
b	1.0	1.1×10^5
A	2.3×10^{10}	3.2×10^{10}

3.7.2 Tam giác hóa Householder

Thuật toán tam giác hóa Householder được thực thi trong Matlab như sau:

```
[Q,R] = qr(A,0);
x = R\(Q'*b);                Tam giác hóa Householder của A.
x(15)                        Giải x
   ans = 1.00000031528723
```

Do sự trực chuẩn hóa nên kết quả chính xác sẽ là $x_{15} = 1$. Do đó ta có một sai số tương đối khoảng 3×10^{-7}. Vì phép tính được làm trong số học chính xác bội IEEE với $\epsilon_{machine} \approx 10^{-16}$, điều này có nghĩa là các sai số làm tròn đã được khuếch đại bởi một thừa số bậc 10^9. Số điều

kiện của x tương ứng với các nhiễu trong A là bậc 10^{10}. Do đó sự không đúng trong x_{15} có thể được giải thích hoàn toàn bằng điều kiện xấu, không phải là không ổn định. Thuật toán 3.1 là ổn định ngược.

Ta tạo thành $\widehat{Q}$ rõ ràng. Nó đủ để lưu trữ các vector v_k được xác định tại bước thứ k của Thuật toán 2.3 (phương trình (2.5.6)), khi đó nó có thể được sử dụng để tính $\widehat{Q} * b$ bằng Thuật toán 2.4. Trong Matlab, ta có thể đạt được hiệu quả này bằng việc tính một phân tích QR không chỉ của A mà của ma trận $[Ab]$ có kích thước "được tăng thêm" $m \times (n+1)$. Trong quá trình của phân tích này, n phản xạ Householder làm A thành ma trận tam giác trên cũng được áp dụng với b, việc rời khỏi vector $\widehat{Q} * b$ trong n vị trí đầu tiên của cột $n+1$. Khi đó phản xạ thứ $(n+1)$ thêm vào được áp dụng để làm các phần tử $n+2, \ldots, m$ của $n+1$ cột 0, nhưng điều này không làm thay đổi n phần tử đầu tiên của cột đó. Do đó:

```
[Q,R] = qr([A b],0);
Qb = R(1:n,n+1);          Tam giác hóa Householder của [A b].
R = R(1:n, 1:n)           Trích ra Q̂*b ...
x = B\Qb;                 ... và R̂.
x(15)                     Giải tìm x.
   ans = 1.00000031529465
```

Các sai số được đưa ra trong phân tích QR của A làm đầy chúng được đưa ra trong tính toán $\widehat{Q} * b$.

Đây cũng là cách thứ 3 để giải bài toán bình phương nhỏ nhất thông qua tam giác hóa Householder trong Matlab. Chúng ta có thể sử dụng phép toán được xây dựng sẵn

```
x = A\b;
x(15)                     Giải tìm x.
   ans = 0.99999994311087
```

Kết quả này cho bậc của độ lớn chính xác hơn. Lý do cho điều này là phép toán của Matlab sử dụng *phân tích QR với việc quay cột*, dựa vào một phân tích $AP = \widehat{Q}\widehat{R}$, mà P là một ma trận hoán vị.

Ba biến thể của phân tích QR này là bằng nhau. Tất cả chúng có thể được chứng minh là ổn định ngược.

Định lý 3.7.1 *Cho bài toán bình phương nhỏ nhất hạng đầy đủ (2.6.2) được giải bằng tam giác hóa Householder (Thuật toán 3.1) trong một máy tính thỏa mãn (3.2.5) - (3.2.7). Thuật toán này là ổn định ngược mà lời giải được tính $\tilde{x}$ có tính chất*

$$\|(A + \delta A)\tilde{x} - b\| = min, \ \frac{\|\delta A\|}{\|A\|} = O(\epsilon_{machine}) \tag{3.7.1}$$

với $\delta A \in \mathbb{C}^{m \times n}$. Điều này là đúng nếu $\widehat{Q} * b$ được tính thông dạng thông thường của $\widehat{Q}$ hoặc bằng Thuật toán 2.4. Nó cũng đúng cho tam giác hóa Householder với việc quay cột bất kì.

3.7.3 Trực giao hóa Gram - Schmidt

Cách khác để giải một bài toán bình phương nhỏ nhất là trực giao hóa Gram - Schmidt được sửa đổi (Thuật toán 2.2). Cho $m \approx n$, điều này đưa ra nhiều phép toán hơn là xấp xỉ Householder, nhưng với $m \gg n$, số phép toán dấu chấm động cho cả thuật toán là xấp xỉ tiệm cận $2mn^2$.

Trình tự Matlab theo sau thực thi thuật toán này trong mô hình trước. Hàm *msg* là một thực thi của Thuật toán 2.2

```
[Q,R] = msg(A);           Trực giao hóa Gram - Sc
x = R\(Q'*b);             A.
x(15)                     Giải tìm x.
   ans = 1.02926594532672
```

Các sai số làm tròn đã được khuếch đại bằng một thừa số bậc 10^{14}, xa hơn số điều kiện của bài toán. Thuật toán này là không ổn định, và nguyên nhân được nhận biết dễ dàng. Như được đề cập tại phần cuối cùng của mục 2.4, trực giao hóa Gram - Schmidt đưa ra các ma trận $\widehat{Q}$ mà các cột của chúng không trực giao một cách chính xác.

Tính không ổn định có thể được cho phép bằng một phát biểu lại của thuật toán. Vì bước lặp Gram - Schmidt phát biểu tích $\widehat{Q}\widehat{R}$ chính xác, ngay cả khi $\widehat{Q}$ không có các cột trực giao, một xấp xỉ để cài đặt các phương trình chuẩn $Rx = (\widehat{Q}^*\widehat{Q})^{-1}\widehat{Q}^*b$ với vector Rx, thì x được tính bằng phép thế ngược. Chỉ cần $\widehat{Q}$ được tính ít nhất là điều kiện tốt, phương pháp này sẽ là không ổn định được miêu tả bên dưới cho các phương trình chính tắc được áp dụng cho các ma trận bất kì. Tuy nhiên, nó bao gồm việc làm thêm vào không cần thiết và nên không được sử dụng trong thực hành.

Một phương pháp tốt hơn của phương pháp Gram - Schmidt ổn định là sử dụng một hệ thống được làm tăng thêm của các phương trình, ngay trong thực thi Housholder thứ hai ở trên:

```
[Q R] = msg([A b]);          Trực giao hóa Gram - Schmidt của
Qb = R(1:n, n+1);            [A b].
R = R(1:n,1:n);              Trích ra Q̂ * b ...
x = R\Qb;                    ... và R̂.
x(15)                        Giải tìm x.
   ans = 1.0000005653399
```

Kết quả bây giờ trông tốt như với tam giác hóa Householder.

Định lý 3.7.2 *Lời giải của bài toán bình phương nhỏ nhất hạng đầy đủ (2.6.2) bằng trực giao hóa Gram - Schmidt cũng là ổn định ngược, thỏa mãn (3.7.1), chứng minh rằng $\widehat{Q}^*b$ được tao thành như được cho biết trong đoạn code ở trên.*

3.7.4 Các phương trình chính tắc

Một xấp xỉ khác cơ bản tới các bài toán bình phương nhỏ nhất là lời giải của các phương trình chính tắc (Thuật toán 2.6), tiêu biểu bằng phân tích Cholesky (Mục 4.4). Với $m \gg n$, phương pháp này là nhanh gấp đôi các phương pháp phụ thuộc vào sự trực giao hóa, chỉ yêu cầu mn^2 phép toán dấu chấm động (2.6.14). Trong thực thi theo sau, bài toán được giải trong 1 dòng đơn của Matlab bằng phép toán

```
x = (A'*A)\(A'*b);           Tạo thành và giải các phương trình
x(15)                        chính tắc.
   ans = 0.39339069870283
```

Nó là trường hợp xấu nhất mà ta đã thu được, với không một chữ số đơn của sự đúng đắn. Sử dụng các phương trình chính tắc rõ ràng là phương pháp không ổn định cho việc giải các bài toán bình phương nhỏ nhất.

Giả sử ta có một thuật toán ổn định ngược cho bài toán hạng đầy đủ (2.6.2) mà nó đưa ra một lời giải $\tilde{x}$ thỏa mãn $\|(A + \delta A)\tilde{x} - b\| = min$ cho δA bất kì với $\|\delta A\|/\|A\| = O(\epsilon_{machine})$. Do Thuật toán 3.3.2 và (3.6.1), ta có

$$\frac{\|\tilde{x} - x\|}{\|x\|} = O\left(\left(\kappa + \frac{\kappa^2 \tan\theta}{\eta}\right)\epsilon_{machine}\right), \tag{3.7.2}$$

với $\kappa = \kappa(A)$. Bây giờ giả sử A là điều kiện xấu, nghĩa là, $\kappa \gg 1$, và θ bị chặn từ $\pi/2$. Phụ thuộc vào các giá trị của các tham số khác nhau, hai chổ rất khác nhau có thể xuất hiện. Nếu $\tan\theta$ bậc 1 và $\eta \ll \kappa$, vế phải của (3.7.2) là $(\kappa^2\epsilon_{machine})$. Mặc khác, nếu $\tan\theta$ là gần 0 hoặc η cũng gần κ, chặn là $O(\kappa\epsilon_{machine})$. *Số điều kiện của bài toán bình phương nhỏ nhất có thể nằm bất kì nơi nào trong phạm vi từ κ tới κ^2.*

Phân tích Cholesky là một thuật toán ổn định cho hệ thống này của các phương trình mà nó đưa ra một lời giải $\tilde{x}$ thỏa mãn $(A^*A+\delta H)\tilde{x} = A^*b$ với δH bất kì mà $\|\delta H\|/\|A^*A\| = O(\epsilon_{machine})$ (Định lý 4.4.2). Tuy nhiên, ma trận A^*A có số điều kiện κ^2 mà không phải là κ. Do đó cách tốt nhất mà ta có thể mong đợi từ các phương trình chính tắc là

$$\frac{\|\tilde{x} - x\|}{\|x\|} = O(\kappa^2 \epsilon_{machine}). \tag{3.7.3}$$

Xử lý của các phương trình chính tắc được chi phối bởi κ^2, không phải κ.

Nếu $\tan\theta$ là bậc 1 và $\eta \ll \kappa$, hoặc nếu κ bậc 1, thì (3.7.2) và (3.7.3) là cùng bậc và các phương trình chính tắc là ổn định. Nếu κ lớn và hoặc $\tan\theta$ là gần với 0 hoăc η là gần với κ thì (3.7.3) lớn hơn nhiều (3.7.2) và các phương trình chính tắc là không ổn định. *Các phương trình chính tắc tiêu biểu là không ổn định cho các bài toán điều kiện xấu bao gồm các cố định gần.* Trong bài toán ví dụ của chúng ta, với $\kappa^2 \approx 10^{20}$, phân tích Cholesky không mang lại các chữ số chính xác.

Theo sau các định nghĩa, một thuật toán là ổn định chỉ nếu nó thõa mãn xử lý một cách đồng nhất qua tất cả các bài toán dưới sự cân nhắc. Do đó kết quả theo sau là một hình thức hóa của các sự quan sát đã làm.

Định lý 3.7.3 *Lời giải của bài toán bình phương nhỏ nhất hạng đầy đủ (2.6.2) thông qua các phương trình chính tắc (Thuật toán 2.6) là không ổn định. Tính ổn định có thể được đạt được, tuy nhiên, bằng sự hạn chế tới một lớp các bài toán mà trong đó $\kappa(A)$ bị chặn đều trên hoặc $(\tan\theta)/\eta$ bị chặn đều dưới.*

3.7.5 SVD

Sử dụng SVD (Thuật toán 2.8) để giải bài toán bình phương nhỏ nhất như được đề cập trong mục 2.6 là ổn định:

```
[U,S,V] = svd(A,0);
x = V*(S\(U'*b));                    SVD được sửa đổi của A.
x(15)                                Giải tìm x.
   ans = 0.99999998230471
```

Thật vậy, đây là đúng đắn nhất trong tất cả các kết quả thu được trong các thực thi của chúng ta, thất bại của tam giác hóa Householder với việc quay cột (của Matlab) bằng một thừa số khoảng 3. Một định lý có thể được chứng minh trong dạng thông thường.

Định lý 3.7.4 *Lời giải của bài toán bình phương nhỏ nhất hạng đầy đủ (2.6.2) bằng SVD (Thuật toán 2.8) là ổn định ngược, thỏa mãn ước lượng (3.7.1).*

3.7.6 Các bài toán bình phương nhỏ nhất hạng không đầy đủ

Trong mục này ta đã được định nghĩa 4 thuật toán ổn định ngược cho các bài toán bình phương nhỏ nhất tuyến tính: tam giác hóa Householder, tam giác hóa Householder với việc quay cột, Gram -Schmidt được sửa đổi với phép tính ngầm của $\widehat{Q} * b$, và SVD. Từ điểm chính của phân tích ổn định cổ điển của bài toán hạng đầy đủ (2.6.2), sự khác nhau giữa các thuật toán này là không quan trọng, nên ta có thể sử dụng đơn giản nhất và chi phí ít nhất, tam giác hóa Householder không quay.

Tuy nhiên, các loại khác của các bài toán bình phương nhỏ nhất mà việc quay cột và SVD đưa ra một sự quan trọng đặc biệt. Đây là các bài toán mà A có hạng nhỏ hơn n, có thể với $m < n$, nên hệ thống các phương trình là *được xác định dưới.* Các bài toán như vậy không có lời giải duy nhất trừ khi ta thêm điều kiện, tiêu biểu là bản thân x phải có một chuẩn nhỏ như

có thể. Sự phức tạp xa hơn là lời giải chính xác phụ thuộc vào hạng của A, và việc xác định số lượng các hạng nằm trong biểu diễn của các sai số làm tròn là vấn đề không bao giờ là tầm thường.
Do đó các bài toán bình phương nhỏ nhất hạng không đầy đủ không là một lớp con bài toán thách thức của các bài toán bình phương nhỏ nhất, nhưng khác nhau cơ bản. Vì sự xác định của một lời giải là mới, nên không có lý do mà thuật toán là ổn định câc bài toán hạng đầy đủ cũng phải ổn định cho trường hợp hạng không đầy đủ. Thật vậy, chỉ các thuật toán ổn định cho các bài toán hạng không đầy đủ là phụ thuộc vào SVD. Một sự lựa chọn là tam giác hóa Householder với việc quay cột, mà nó là ổn định cho hầu hết các bài toán.

Bài tập

1. (a) Chứng minh rằng $(1+O(\epsilon_{machine}))(1+O(\epsilon_{machine})) = 1+O(\epsilon_{machine})$. Ý nghĩa chính xác của phát biểu này là nếu f là một hàm thỏa $f(\epsilon_{machine}) = (1+O(\epsilon_{machine}))(1+O(\epsilon_{machine}))$ khi $\epsilon_{machine} \to 0$ thì f cũng thỏa $f(\epsilon_{machine}) = (1+O(\epsilon_{machine}))$ khi $\epsilon_{machine} \to 0$.
   (b) Chứng minh rằng $(1+O(\epsilon_{machine}))^{-1} = (1+O(\epsilon_{machine}))$.

2. Xét một thuật toán cho bài toán tính SVD (đầy đủ)của một ma trận. Dữ liệu cho bài toán này là ma trận A và lời giải là ba ma trận U (Unita), Σ (đường chéo), và V (Unita) sao cho $A = U\Sigma V^*$.
   (a) Giải thích thuật toán này là ổn định ngược.
   (b) Vì một lý do đơn gian mà thuật toán này là không ổn định ngược. Giải thích.
   (c) Các thuật toán tiêu chuẩn cho việc tính toán SVD là ổn định. Giải thích tính ổn định cho 1 thuật toán như vậy.

3. (a) Cho các ma trận Unita $Q_1, \ldots, Q_k \in \mathbb{C}^{m\times m}$ được cố định và xét bài toán tính tích $B = Q_k \ldots Q_1 A$, với $A \in \mathbb{C}^{m\times n}$. Tính toán được thực hiện từ trái qua phải bằng các phép toán dấu chấm động trong máy tính thỏa (3.2.5) và (3.2.7). Chứng minh rằng thuật toán là ổn định ngược (A được làm nhiễu, Q_j cố định và không được làm nhiễu).
   (b) Cho một ví dụ để chứng minh kết quả này là không đúng nếu các ma trận Unita Q_j được thay bằng các ma trận tùy ý $X_j \in \mathbb{C}^{m\times m}$.

4. Xét
$$\begin{bmatrix} 1 & 1 \\ 1 & 1.0001 \\ 1 & 1.0001 \end{bmatrix}, \quad \begin{bmatrix} 2 \\ 0.0001 \\ 4.0001 \end{bmatrix}$$
   (a) Tính A^+, P.
   (b) Tìm các lời giải chính xác của x và $y = Ax$ cho bài toán bình phương nhỏ nhất $Ax \approx b$.
   (c) Tính $\kappa(A), \theta$ và η.

5. Cho $A \in \mathbb{C}^{m\times n}$ hạng n và $b \in \mathbb{C}^m$, xét hệ thống khối 2×2 của các phương trình
$$\begin{bmatrix} I & A \\ A^* & 0 \end{bmatrix} \begin{bmatrix} r \\ x \end{bmatrix} = \begin{bmatrix} b \\ 0 \end{bmatrix}$$
   với I là ma trận đơn vị $n \times n$. Chứng minh hệ thống này có một nghiệm duy nhất $(r, x)^T$ và r, x là thặng dư và lời giải của bài toán bình phương nhỏ nhất (3.6.1).

6. Cho đoạn code trong Matlab như sau

```
[U, V, S] = svd(A);
S = diag(A);
tol = max(size(A))*S(1)*eps;
r = sum(S > tol);
S = diag(ones(r, 1)./S(1:r));
X = V(:, 1:r)*S*U(:, 1:r)';
```

Đoạn code trên trả về kết quả gì?

Chương 4

Hệ phương trình

4.1 Khử Gauss

4.1.1 Phân tích LU

Khử Gauss biến một hệ thống tuyến tính đầy đủ thành một hệ thống tam giác trên bằng việc áp dụng các phép biến đổi tuyến tính đơn giản bên trái. Nó tương tự tam giác hóa Householder cho việc tính các phân tích QR. Khác nhau là các phép biến đổi được áp dụng trong khử Gauss không là unita.

Cho $A \in \mathbb{C}^{m\times m}$ là một ma trận vuông. (Thuật toán cũng có thể được áp dụng cho các ma trận hình chữ nhật, nhưng điều này ít được làm trong thực hành) Ý tưởng là để biến đổi A thành một ma trận tam giác trên U có $m \times m$ chiều bằng việc đưa ra các số 0 bên dưới đường chéo, đầu tiên trong cột 1, trong cột 2, ...- như trong tam giác hóa Householder. Điều này được làm bằng việc trừ các bội của mỗi dòng từ các dòng con theo sau. Quá trình "khử" này là bằng với việc nhân A cho một chuỗi các ma trận tam giác dưới L_k trong vế trái:

$$\underbrace{L_{m-1}\dots L_2L_1}_{L^{-1}}A = U. \tag{4.1.1}$$

Đặt $L = L_1^{-1}L_2^{-1}\dots L_{m-1}^{-1}$ được $A = LU$. Do đó ta thu được một *phân tích LU* của A

$$A = LU, \tag{4.1.2}$$

với U là ma trận tam giác trên và L là ma trận tam giác dưới. Nó đưa ra L là *tam giác dưới đơn vị*, nghĩa là tất cả các phần tử trên đường chéo của nó là bằng 1.

Ví dụ, giả sử ta bắt đầu với một ma trận 4×4. Thuật toán tiến hành trong 3 bước (so với (2.5.1))

$$\underset{A}{\begin{bmatrix}\times&\times&\times&\times\\ \times&\times&\times&\times\\ \times&\times&\times&\times\\ \times&\times&\times&\times\end{bmatrix}} \xrightarrow{L_1} \underset{L_1A}{\begin{bmatrix}\times&\times&\times&\times\\ \mathbf{0}&\times&\times&\times\\ \mathbf{0}&\times&\times&\times\\ \mathbf{0}&\times&\times&\times\end{bmatrix}} \xrightarrow{L_2} \underset{L_2L_1A}{\begin{bmatrix}\times&\times&\times&\times\\ &\times&\times&\times\\ &\mathbf{0}&\times&\times\\ &\mathbf{0}&\times&\times\end{bmatrix}} \xrightarrow{L_3} \underset{L_3L_2L_1A}{\begin{bmatrix}\times&\times&\times&\times\\ &\times&\times&\times\\ &&\times&\times\\ &&\mathbf{0}&\boldsymbol{x}\end{bmatrix}}.$$

Phép biến đổi thứ k L_k đưa các số 0 bên dưới đường chéo trong cột k bằng việc trừ các bội của dòng k từ các dòng $k+1,\dots,m$. Vì $k-1$ phần tử đầu tiên của dòng k là 0, phép toán này không phá hủy các số 0 bất kì được đưa ra trước đó.

Do đó khử Gauss làm tăng thêm sự phân loại các thuật toán của chúng ta cho việc phân tích một ma trận:

$$\text{Gram - Schmidt: } A = QR \text{ bằng trực giao hóa tam giác,}$$
$$\text{Householder: } A = QR \text{ bằng tam giác hóa trực giao,}$$
$$\text{Khử Gauss: } A = LU \text{ bằng tam giác hóa tam giác.}$$

4.1.2 Ví dụ

Giả sử ta bắt đầu với ma trận 4×4

$$A = \begin{bmatrix} 2 & 1 & 1 & 0 \\ 4 & 3 & 3 & 1 \\ 8 & 7 & 9 & 5 \\ 6 & 7 & 9 & 8 \end{bmatrix}. \tag{4.1.3}$$

Bước đầu tiên của khử Gauss là

$$L_1 A = \begin{bmatrix} 1 & & & \\ -2 & 1 & & \\ -4 & & 1 & \\ -3 & & & 1 \end{bmatrix} \begin{bmatrix} 2 & 1 & 1 & 0 \\ 4 & 3 & 3 & 1 \\ 8 & 7 & 9 & 5 \\ 6 & 7 & 9 & 8 \end{bmatrix} = \begin{bmatrix} 2 & 1 & 1 & 0 \\ & 1 & 1 & 1 \\ & 3 & 5 & 5 \\ & 4 & 6 & 8 \end{bmatrix}.$$

Ta đã trừ 2 lần dòng thứ nhất từ dòng thứ 2, 4 lần dòng thứ nhất từ dòng thứ 3, và 3 lần dòng thứ nhất từ dòng thứ 4. Bước thứ 2 giống như điều này:

$$L_2 L_1 A = \begin{bmatrix} 1 & & & \\ & 1 & & \\ & -3 & 1 & \\ & -4 & & 1 \end{bmatrix} \begin{bmatrix} 2 & 1 & 1 & 0 \\ & 1 & 1 & 1 \\ & 3 & 5 & 5 \\ & 4 & 6 & 8 \end{bmatrix} = \begin{bmatrix} 2 & 1 & 1 & 0 \\ & 1 & 1 & 1 \\ & & 2 & 2 \\ & & 2 & 4 \end{bmatrix}.$$

Lần này ta đã trừ 3 lần dòng 2 từ dòng 3 và 4 lần dòng 2 từ dòng 4. Cuối cùng, trong bước thứ 3 ta trừ dòng 3 từ dòng 4:

$$L_3 L_2 L_1 A = \begin{bmatrix} 1 & & & \\ & 1 & & \\ & & 1 & \\ & & -1 & 1 \end{bmatrix} \begin{bmatrix} 2 & 1 & 1 & 0 \\ & 1 & 1 & 1 \\ & & 2 & 2 \\ & & 2 & 4 \end{bmatrix} = \begin{bmatrix} 2 & 1 & 1 & 0 \\ & 1 & 1 & 1 \\ & & 2 & 2 \\ & & & 2 \end{bmatrix} = U.$$

Bây giờ, để đưa ra phân tích đầy đủ $A = LU$, ta cần tính tích $L = L_1^{-1} L_2^{-1} L_3^{-1}$. Nghịch đảo của L_1 phải là L_1, nhưng với mỗi phần tử bên dưới đường chéo được lấy phủ định:

$$\begin{bmatrix} 1 & & & \\ -2 & 1 & & \\ -4 & & 1 & \\ -3 & & & 1 \end{bmatrix}^{-1} = \begin{bmatrix} 1 & & & \\ 2 & 1 & & \\ 4 & & 1 & \\ 3 & & & 1 \end{bmatrix}. \tag{4.1.4}$$

Tương tự, các nghịch đảo của L_2 và L_3 được thu được bằng việc lấy phủ định các phần tử dưới đường chéo. Cuối cùng, tích $L_1^{-1} L_2^{-1} L_3^{-1}$ cũng là ma trận tam giác dưới đơn vị với các phần tử dưới đường chéo khác 0 của L_1^{-1}, L_2^{-1}, và L_3^{-1} đã đưa vào những nơi xấp xỉ. Cùng với tất cả, ta có

$$\underset{A}{\begin{bmatrix} 2 & 1 & 1 & 0 \\ 4 & 3 & 3 & 1 \\ 8 & 7 & 9 & 5 \\ 6 & 7 & 9 & 8 \end{bmatrix}} = \underset{L}{\begin{bmatrix} 1 & & & \\ 2 & 1 & & \\ 4 & 3 & 1 & \\ 3 & 4 & 1 & 1 \end{bmatrix}} \underset{U}{\begin{bmatrix} 2 & 1 & 1 & 0 \\ & 1 & 1 & 1 \\ & & 2 & 2 \\ & & & 2 \end{bmatrix}} \tag{4.1.5}$$

4.1.3 Công thức tổng quát

Dưới đây là các công thức tổng quát cho một ma trận $m \times m$. Giả sử x_k ký hiệu là cột thứ k của ma trận bắt đầu bước k. Khi đó phép biến đổi L_k phải được chọn sao cho

$$x_k = \begin{bmatrix} x_{1k} \\ \vdots \\ x_{kk} \\ x_{k+1,k} \\ \vdots \\ x_{mk} \end{bmatrix} \xrightarrow{L_k} L_k x_k = \begin{bmatrix} x_{1k} \\ \vdots \\ x_{kk} \\ 0 \\ \vdots \\ 0 \end{bmatrix}.$$

Để làm điều này ta mong muốn trừ l_{jk} lần dòng k từ dòng j, với l_{jk} là *số nhân*

$$l_{jk} = \frac{x_{jk}}{x_{kk}} \ (k < j \leqslant m). \tag{4.1.6}$$

Ma trận L_k có dạng

$$L_k = \begin{bmatrix} 1 & & & & \\ & \ddots & & & \\ & & 1 & & \\ & & -l_{k+1,k} & 1 & \\ & & \vdots & & \ddots \\ & & -l_{mk} & & 1 \end{bmatrix},$$

với các phần tử dưới đường chéo khác 0 được thay thế trong cột k. Điều này tương tự (2.5.2) cho tam giác hóa Householder.

Trong ví dụ ở trên, L_k đó có thể được đảo ngược bằng việc lấy phủ định các phần tử dưới đường chéo của nó (5.6.1), và L đó có thể được tạo thành bằng việc tập hợp các phần tử l_{jk} trong các nơi xấp xỉ (4.1.5). Ta hãy xác định

$$l_k = \begin{bmatrix} 0 \\ \vdots \\ 0 \\ l_{k+1,k} \\ \vdots \\ l_{m,k} \end{bmatrix}.$$

Khi đó L_k có thể được viết $L_k = I - l_k e_k^*$, với e_k là vector cột với 1 nằm ở vị trí k và 0 nằm ở những vị trí khác. Kiểu thưa thớt của l_k kéo theo $e_k^* l_k = 0$, và do đó $(I - l_k e_k^*)(I + l_k e_k^*) = I - l_k e_k^* l_k e_k^* = I$. Mặt khác, nghịch đảo của L_k là $I + l_k e_k^*$ như trong (4.1.5).

Xét ví dụ, tích $L_k^{-1} L_{k+1}^{-1}$. Từ kiểu thưa thớt của l_{k+1}, ta có $e_k^* l_{k+1} = 0$, và do đó

$$L_k^{-1} L_{k+1}^{-1} = (I + l_k e_k^*)(I + l_{k+1} e_{k+1}^*) = I + l_k e_k^* + l_{k+1} e_{k+1}^*.$$

Do đó $L_k^{-1} L_{k+1}^{-1}$ cũng là ma trân tam giác dưới đơn vị với các phần tử của cả L_k^{-1} và L_{k+1}^{-1} được thêm vào bên dưới đường chéo. Khi ta lấy tích của tất cả các ma trận này để tạo thành dạng L

$$L = L_1^{-1} L_2^{-1} \dots L_{m-1}^{-1} = \begin{bmatrix} 1 & & & & \\ l_{21} & 1 & & & \\ l_{31} & l_{32} & 1 & & \\ \vdots & \vdots & \ddots & \ddots & \\ l_{m1} & l_{m2} & \cdots & l_{m,m-1} & 1 \end{bmatrix}. \tag{4.1.7}$$

Mặc dù ta không đề cập nó trong mục 2.3, xét sự thưa thớt mà chúng dẫn đến (4.1.7) cũng xuất hiện trong giải thích (2.3.10) của Gram - Schmidt được sửa đổi xử lý các phép nhân phải liên tiếp cho các ma trận tam giác R_k.

Khử Gauss trong thực hành, các ma trận L_k không bao giờ được tạo thành và được nhân rõ ràng. Các số nhân l_{jk} được tính và lưu trực tiếp vào L, và khi đó các phép biến đổi L_k được áp dụng:

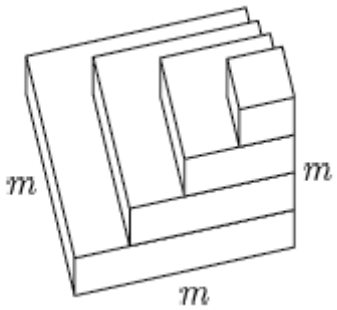
Thuật toán 4.1 Khử Gauss không quay

```
U = A, L = I
for k = 1 to m − 1 do
    for j = k + 1 to m do
        l_jk = u_jk / u_kk
        u_{j,k:m} = u_{j,k:m} − l_jk u_{k,k:m}
    end for
end for
```

4.1.4 Đếm số phép toán

Đếm phép toán tiệm cận của thuật toán này có thể được suy ra từ hình học. Việc làm được chi phối bởi phép toán vector trong vòng lặp bên trong, $u_{j,k:m} = u_{j,k:m} - l_{jk}u_{k,k:m}$, mà nó thực thi một phép nhân vector với vô hướng và một phép trừ vector. Nếu $l = m - k + 1$ ký hiệu là chiều dài của các vector dòng đang được thao tác, số phép toán dấu chấm động là $2l$: 2 phép toán dấu chấm động trên phần tử.

Với mỗi giá trị của k, vòng lặp bên trong được lặp lại cho các dòng $k + 1, \ldots, m$. Việc làm đã bao gồm sự tương ứng tới một lớp của khối theo sau:
Đây là hình giống với hình mà ta đã đưa ra trong mục 2.3 để biểu diễn tam giác hóa Householder



(giả sử $m = n$). Tuy nhiên, mỗi hình lập phương đơn vị biểu diễn 4 phép toán dấu chấm động hơn là 2. Như trước đây, khối hội tụ tới một hình chớp khi $m \to \infty$, với thể tích $\frac{1}{3}m^3$. Tại 2 phép toán dấu chấm động trên một đơn vị thể tích, điều này tăng thêm thành

$$\text{Khử Gauss: } \approx \frac{2}{3}m^3 \text{ phép toán dấu chấm động.} \tag{4.1.8}$$

4.1.5 Giải phương trình $Ax = b$ bằng phân tích LU

Nếu A được phân tích thành L và U thì một hệ thống các phương trình $Ax = b$ được giảm xuống thành dạng $LUx = b$. Khi đó nó có thể được giải bằng việc giải 2 hệ thống tam giác: đầu

tiên là $Ly = b$ với biến y (phép thế ngược), khi đó $Ux = y$ với biến x (phép thế ngược). Bước đầu tiên cần $\sim \frac{2}{3}m^3$ phép toán dấu chấm động. Bước 2 và bước 3 thì mỗi bước cần $\sim m^2$ phép toán dấu chấm động. Tổng số là $\sim \frac{2}{3}m^3$ phép toán dấu chấm động, một phần hai của hình là $\sim \frac{4}{3}m^3$ phép toán dấu chấm động (2.5.10) cho một lời giải bằng tam giác hóa Householder (Thuật toán 3.1).

Vì sao khử Gauss thường được sử dụng hơn là phân tích QR để giải các hệ thống vuông của các phương trình? Thừa số của 2 là một lý do . Tuy nhiên, nó có là cơ sở lập luận có liên quan tới lịch sử mà ý tưởng khử đã được biết nhiều thế kỷ. nơi mà phân tích QR của các ma trận không được biết cho tới khi sự phát minh của các máy tính. Để thay thế khử Gauss như một phương pháp của sự lựa chọn, phân tích QR sẽ phải có một thuận lợi thuyết phục.

4.1.6 Tính không ổn định của khử Gauss không quay

Khử Gauss như được đưa ra không dùng được cho việc giải các hệ thống tuyến tính tổng quát vì nó không ổn định ngược. Tính không ổn định có liên hệ với một cái khác, khó khăn rõ ràng hơn. Với các ma trận nào đó, khử Gauss thất bại hoàn toàn, bởi vì nó xâm phạm đến việc chia cho 0.

Ví dụ, xét

$$A = \begin{bmatrix} 0 & 1 \\ 1 & 1 \end{bmatrix}.$$

Ma trận này có hạng đầy đủ và là điều kiện tốt, với $\kappa(A) = (3+\sqrt{5})/2 \approx 2.618$ trong chuẩn 2. Tuy nhiên, khử Gauss thất bại tại bước đầu tiên.

Giả sử ta áp dụng khử Gauss tới

$$A = \begin{bmatrix} 10^{-20} & 1 \\ 1 & 1 \end{bmatrix}. \tag{4.1.9}$$

Quá trình không thất bại. Thay vì, 10^{20} lần dòng đầu tiên được trừ từ dòng thứ 2, và các thừa số theo sau được đưa ra:

$$L = \begin{bmatrix} 1 & 0 \\ 10^{20} & 1 \end{bmatrix},\ U = \begin{bmatrix} 10^{-20} & 1 \\ 0 & 1-10^{20} \end{bmatrix}.$$

Tuy nhiên, giả sử các tính toán này được thực hiện trong số học dấu chấm động với $\epsilon_{machine} \approx 10^{-16}$. Số $1-10^{20}$ sẽ không được biểu diễn chính xác. Nó sẽ được làm tròn thành số dấu chấm động gần nhất. Chơ đơn giản, nó chính xác là -10^{20}. Khi đó các ma trận dấu chấm động đưa ra bởi thuật toán sẽ là

$$\tilde{L} = \begin{bmatrix} 1 & 0 \\ 10^{20} & 1 \end{bmatrix},\ \tilde{U} = \begin{bmatrix} 100^{-20} & 1 \\ 0 & -10^{20} \end{bmatrix}.$$

Bậc của việc làm tròn này có thể cho phép được đầu tiên. Sau tất cả, ma trận $\tilde{U}$ là gần với U chính xác liên quan với $\|U\|$. Tuy nhiên, bài toán trở thành rõ ràng khi ta tính tích $\tilde{L}\tilde{U}$:

$$\tilde{L}\tilde{U} = \begin{bmatrix} 10^{-20} & 1 \\ 1 & 0 \end{bmatrix}.$$

Ma trận này không gần với A, vì 1 ở vị trí $(2,2)$ đã được thay thế bằng 0. Nếu ta giải hệ thống $\tilde{L}\tilde{U}x = b$ thì kết quả sẽ khác lời giải $Ax = b$. Ví dụ, với $b = (1,0)^*$ ta được $\tilde{x} = (0,1)^*$, trong khi lời giải chính xác là $x \approx (-1,1)^*$.

Khử Gauss được tính toán phân tích LU ổn định: $\tilde{L}$ và $\tilde{U}$ là gần với các thừa số chính xác cho một ma trận gần với A. Phân tích LU là *không ổn định ngược*. Như một quy tắc, nếu một bước của thuật toán là ổn định nhưng không phải là thuật toán ổn định ngược cho việc giải một bài toán con, tính ổn định trên tất cả phép tính có thể là nguy cơ.

Thật vậy, cho các ma trận A có $m \times m$ chiều tổng quát, khử Gauss không quay là không ổn định hoặc không ổn định như một thuật toán tổng quát cho phân tích LU. Hơn nữa, các ma trận tam giác được sinh ra có các số điều kiện mà chúng có thể là tùy ý hơn là các số điều kiện này của chính ma trận A, việc đưa thêm vào các nguồn không ổn định trong các giai đoạn phép thế tiến và ngược của lời giải $Ax = b$.

4.2 Pivoting

Trong mục trước ta thấy rằng khử Gauss trong dạng thuần túy của nó là không ổn định. Tính không ổn định có thể được điều khiển bằng việc làm nhiễu các dòng của ma trận đang được tính trong nó, một phép toán gọi là *quay (pivoting)*. Pivoting đã là một đặc trưng tiêu chuẩn của các tính toán khử Gauss từ những năm 1950.

4.2.1 Pivots

Tại bước k của khử Gauss, các bội của dòng k được trừ từ các dòng $k+1, \ldots, m$ của ma trận X để đưa các số 0 trong k phần tử của các dòng này. Trong phép toán dòng k, cột k, và đặc biệt phần tử x_{kk} đóng vai trò đặc biệt. Ta gọi x_{kk} là *pivot*. Từ mỗi phần tử trong ma trận con $X_{k+1:m,k:m}$ được trừ tích của một số trong dòng k và trong cột k, chia cho x_{kk}:

$$\begin{bmatrix} \times & \times & \times & \times & \times \\ & x_{kk} & \times & \times & \times \\ & \times & \times & \times & \times \\ & \times & \times & \times & \times \\ & \times & \times & \times & \times \end{bmatrix} \longrightarrow \begin{bmatrix} \times & \times & \times & \times & \times \\ & x_{kk} & \times & \times & \times \\ & \mathbf{0} & \times & \times & \times \\ & \mathbf{0} & \times & \times & \times \\ & \mathbf{0} & \times & \times & \times \end{bmatrix}$$

Tuy nhiên, không có lý do vì sao dòng và cột thứ k phải được chọn cho sự khử. Ví dụ, ta có thể dễ dàng đưa các số 0 vào trong cột k bằng việc cộng thêm các bội của dòng i nào đó với $k < i \leqslant m$ với các dòng khác $k, \ldots, m$. Trong trường hợp này, x_{ik} sẽ là pivot. Dưới đây là một minh họa với $k = 2$ và $i = 4$:

$$\begin{bmatrix} \times & \times & \times & \times & \times \\ & \times & \times & \times & \times \\ & \times & \times & \times & \times \\ & x_{kk} & \times & \times & \times \\ & \times & \times & \times & \times \end{bmatrix} \longrightarrow \begin{bmatrix} \times & \times & \times & \times & \times \\ & \mathbf{0} & \times & \times & \times \\ & \mathbf{0} & \times & \times & \times \\ & x_{kk} & \times & \times & \times \\ & \mathbf{0} & \times & \times & \times \end{bmatrix}.$$

Tương tự, ta có thể đưa các số 0 vào trong cột j hơn là cột k. Dưới đây là một minh họa với $k = 2, i = 4, j = 3$:

$$\begin{bmatrix} \times & \times & \times & \times & \times \\ & \times & \times & \times & \times \\ & \times & \times & \times & \times \\ & \times & x_{ij} & \times & \times \\ & \times & \times & \times & \times \end{bmatrix} \longrightarrow \begin{bmatrix} \times & \times & \times & \times & \times \\ & \times & \mathbf{0} & \times & \times \\ & \times & \mathbf{0} & \times & \times \\ & \times & x_{ij} & \times & \times \\ & \times & \mathbf{0} & \times & \times \end{bmatrix}.$$

Nói chung, ta tự do chọn phần tử bất kì của $X_{k:m,k:m}$ làm pivot, chỉ cần nó khác 0. Khả năng mà một phần tử $x_{kk} = 0$ có thể xuất hiện kéo theo một vài tính linh hoạt của việc chọn pivot

đôi khi có thể là cần thiết, ngay cả toán học thuần túy. Cho tính ổn định số, nó mong muốn để pivot ngay khi x_{kk} khác 0 nếu có một phần tử lớn hơn có thể. Đặc biệt, nó là phổ biến để lựa chọn như là pivot số lớn nhất giữa một tập các phần tử đang được xét như các ứng viên.

Cấu trúc của quá trình khử nhanh chóng trở nên khó hiểu nếu các số 0 được đưa vào trong các loại tùy ý thông qua ma trận. Để thấy điều này, ta muốn giữ lại cấu trúc tam giác được miêu tả trong mục cuối cùng, và đây là một cách dễ dàng để làm điều này. Ta sẽ không nghĩ về pivot x_{ij} như ở vế bên trái trong minh họa ở trên. Thay vì, tại bước k, ta sẽ cho rằng các dòng và cột của ma trận đang làm việc được hoán vị để mà di chuyển x_{ij} vào vị trí (k, k). Khi đó sự khử được làm, các số 0 được đưa vào thành các phần tử $k+1, \ldots, m$ của cột k, ngay cả khi trong khử Gauss không quay. Sự đổi chổ cho nhau các dòng và các cột có thể là cái thường được nghĩ như *pivoting*.

Ý tưởng mà các dòng và các cột được đổi chổ cho nhau là một khái niệm bắt buộc. Nếu như nó là một ý tưởng tốt để hoán đổi vị trí chúng trong máy tính là ít rõ ràng. Trong các thực thi bất kỳ, dữ liệu trong bộ nhớ máy tính là được hoán đổi tại mỗi bước pivot. Mặt khác, hiệu quả tương đương được đạt được bởi địa chỉ không trực tiếp với các vector chỉ số được hoán vị. Xấp xỉ là thay đổi tốt nhất từ máy tới máy và phụ thuộc vào nhiều nhân tố.

4.2.2 Quay từng phần

Nếu mỗi phần tử của $X_{k:m,k:m}$ được xem xét như là một pivot có thể tại bước k, thì có $O((m-k)^2)$ phần tử được kiểm tra để xác định là lớn nhất. Tính tổng trên m bước, tổng chi phí của việc chọn các pivot là $O(m^3)$ phép toán, cộng thêm chi phí khử Gauss đáng kể. Chiến lược tốn kém này được gọi là *quay đầy đủ*.

Đặc biệt, các pivot tốt tương đương nhau có thể được tìm thấy bằng việc xét một số lớn hơn nhiều số phần tử. Phương pháp tiêu chuẩn cho việc làm này là *quay từng phần*. Ở đây, chỉ các dòng là được hoán đổi vị trí cho nhau. Pivot tại mỗi bước được chọn như là phần tử lớn nhất của $m-k+1$ phần tử trên đường chéo phụ của cột k mà nó có tổng số chi phí là $O(m-k)$ phép toán cho việc chọn pivot tại mỗi bước. Do đó $O(m^2)$ phép toán trên tất cả. Để đưa pivot thứ k vào vị trí (k, k), không cột nào cần được hoán đổi vị trí; nó đủ để hoán vị dòng k với dòng chứa pivot.

$$\begin{bmatrix} \times & \times & \times & \times & \times \\ & \times & \times & \times & \times \\ & \times & \times & \times & \times \\ & x_{ik} & \times & \times & \times \\ & \times & \times & \times & \times \end{bmatrix} \xrightarrow{P_1} \begin{bmatrix} \times & \times & \times & \times & \times \\ & x_{ik} & \times & \times & \times \\ & \times & \times & \times & \times \\ & \times & \times & \times & \times \\ & \times & \times & \times & \times \end{bmatrix} \xrightarrow{L_1} \begin{bmatrix} \times & \times & \times & \times & \times \\ & x_{ik} & \times & \times & \times \\ & \mathbf{0} & \times & \times & \times \\ & \mathbf{0} & \times & \times & \times \\ & \mathbf{0} & \times & \times & \times \end{bmatrix}.$$

Chọn pivot Đổi dòng Khử

Thuật toán này có thể được biểu diễn như một tích ma trận. Ta đã thấy trong mục cuối cùng mà bước khử tương ứng với phép nhân trái bởi một ma trận tam giác dưới cơ bản L_k. Quay từng phần làm phức tạp các chủ đề bằng việc áp dụng một ma trận hoán vị P_k trong vế trái của ma trận làm việc trước mỗi sự khử.Sau $m-1$ bước, A trở thành một ma trận tam giác dưới U:

$$L_{m-1}P_{m-1}\ldots L_2P_2L_1P_1A = U. \tag{4.2.1}$$

4.2.3 Ví dụ

Ta sẽ sử dụng lại ví dụ 4.1.3,

$$A = \begin{bmatrix} 2 & 1 & 1 & 0 \\ 4 & 3 & 3 & 1 \\ 8 & 7 & 9 & 5 \\ 6 & 7 & 9 & 8 \end{bmatrix}. \tag{4.2.2}$$

Với quay từng phần, việc đầu tiên ta làm là đổi chổ dòng đầu tiên và dòng thứ 3 (nhân trái với P_1):

$$\begin{bmatrix} & & 1 & \\ & 1 & & \\ 1 & & & \\ & & & 1 \end{bmatrix} \begin{bmatrix} 2 & 1 & 1 & 0 \\ 4 & 3 & 3 & 1 \\ 8 & 7 & 9 & 5 \\ 6 & 7 & 9 & 8 \end{bmatrix} = \begin{bmatrix} 8 & 7 & 9 & 5 \\ 4 & 3 & 3 & 1 \\ 2 & 1 & 1 & 0 \\ 6 & 7 & 9 & 8 \end{bmatrix}.$$

Bước khử đầu tiên (nhân trái với L_1):

$$\begin{bmatrix} 1 & & & \\ -\frac{1}{2} & 1 & & \\ -\frac{1}{4} & & 1 & \\ -\frac{3}{4} & & & 1 \end{bmatrix} \begin{bmatrix} 8 & 7 & 9 & 5 \\ 4 & 3 & 3 & 1 \\ 2 & 1 & 1 & 0 \\ 6 & 7 & 9 & 8 \end{bmatrix} = \begin{bmatrix} 8 & 7 & 9 & 5 \\ & -\frac{1}{2} & -\frac{3}{2} & -\frac{3}{2} \\ & -\frac{3}{4} & -\frac{5}{4} & -\frac{5}{4} \\ & \frac{7}{4} & \frac{9}{4} & \frac{17}{4} \end{bmatrix}.$$

Dòng thứ 2 và dòng thứ 4 được đổi chổ (nhân với P_2):

$$\begin{bmatrix} 1 & & & \\ & & & 1 \\ & & 1 & \\ & 1 & & \end{bmatrix} \begin{bmatrix} 8 & 7 & 9 & 5 \\ & -\frac{1}{2} & -\frac{3}{2} & -\frac{3}{2} \\ & -\frac{3}{4} & -\frac{5}{4} & -\frac{5}{4} \\ & \frac{7}{4} & \frac{9}{4} & \frac{17}{4} \end{bmatrix} = \begin{bmatrix} 8 & 7 & 9 & 5 \\ & \frac{7}{4} & \frac{9}{4} & \frac{17}{4} \\ & -\frac{3}{4} & -\frac{5}{4} & -\frac{5}{4} \\ & -\frac{1}{2} & -\frac{3}{2} & -\frac{3}{2} \end{bmatrix}.$$

Khi đó bước khử thứ 2 (nhân với L_1):

$$\begin{bmatrix} 1 & & & \\ & 1 & & \\ & \frac{3}{7} & 1 & \\ & \frac{2}{7} & & 1 \end{bmatrix} \begin{bmatrix} 8 & 7 & 9 & 5 \\ & \frac{7}{4} & \frac{9}{4} & \frac{17}{4} \\ & -\frac{3}{4} & -\frac{5}{4} & -\frac{5}{4} \\ & -\frac{1}{2} & -\frac{3}{2} & -\frac{3}{2} \end{bmatrix} = \begin{bmatrix} 8 & 7 & 9 & 5 \\ & \frac{7}{4} & \frac{9}{4} & \frac{17}{4} \\ & & -\frac{2}{7} & \frac{4}{7} \\ & & -\frac{6}{7} & -\frac{2}{7} \end{bmatrix}.$$

Dòng thứ 3 và dòng thứ 4 được đổi chổ (nhân với P_3):

$$\begin{bmatrix} 1 & & & \\ & 1 & & \\ & & & 1 \\ & & 1 & \end{bmatrix} \begin{bmatrix} 8 & 7 & 9 & 5 \\ & \frac{7}{4} & \frac{9}{4} & \frac{17}{4} \\ & & -\frac{2}{7} & \frac{4}{7} \\ & & -\frac{6}{7} & -\frac{2}{7} \end{bmatrix} = \begin{bmatrix} 8 & 7 & 9 & 5 \\ & \frac{7}{4} & \frac{9}{4} & \frac{17}{4} \\ & & -\frac{6}{7} & -\frac{2}{7} \\ & & -\frac{2}{7} & \frac{4}{7} \end{bmatrix}.$$

Bước khử cuối cùng (nhân với L_3):

$$\begin{bmatrix} 1 & & & \\ & 1 & & \\ & & 1 & \\ & & -\frac{1}{3} & 1 \end{bmatrix} \begin{bmatrix} 8 & 7 & 9 & 5 \\ & \frac{7}{4} & \frac{9}{4} & \frac{17}{4} \\ & & -\frac{6}{7} & -\frac{2}{7} \\ & & -\frac{2}{7} & \frac{4}{7} \end{bmatrix} = \begin{bmatrix} 8 & 7 & 9 & 5 \\ & \frac{7}{4} & \frac{9}{4} & \frac{17}{4} \\ & & -\frac{6}{7} & -\frac{2}{7} \\ & & & \frac{2}{3} \end{bmatrix}.$$

4.2.4 Phân tích $PA = LU$

Ta đã tính phân tích LU của PA, với P là một ma trận hoán vị.

$$\underset{P}{\begin{bmatrix} & & 1 & \\ & & & 1 \\ & 1 & & \\ 1 & & & \end{bmatrix}}\underset{A}{\begin{bmatrix} 2 & 1 & 1 & 0 \\ 4 & 3 & 3 & 1 \\ 8 & 7 & 9 & 5 \\ 6 & 7 & 9 & 8 \end{bmatrix}}=\underset{L}{\begin{bmatrix} 1 & & & \\ \frac{3}{4} & 1 & & \\ \frac{1}{2} & -\frac{2}{7} & 1 & \\ \frac{1}{4} & -\frac{3}{7} & \frac{1}{3} & 1 \end{bmatrix}}\underset{U}{\begin{bmatrix} 8 & 7 & 9 & 5 \\ & \frac{7}{4} & \frac{9}{4} & \frac{17}{4} \\ & & -\frac{6}{7} & -\frac{2}{7} \\ & & & \frac{2}{3} \end{bmatrix}}. \tag{4.2.3}$$

Công thức này nên được so sánh với (4.1.5). Sự có mặt của các số nguyên và các phân số ở đây là không phân biệt. Sự phân biệt ở đây là tất cả các phần tử đường chéo phụ của L có độ dài nhỏ hơn 1. Nó là một hệ quả của tính chất $|x_{kk}| = \max_j |x_{jk}|$ trong (4.1.6) được đưa ra bởi pivoting.

Nó không rõ ràng từ (4.2.3). Quá trình khử đưa về dạng

$$L_3P_3L_2P_2L_1P_1A = U,$$

mà nó không giống tam giác dưới. Sáu phép toán cơ bản này có thể được sắp xếp lại thành dạng

$$L_3P_3L_2P_2L_1P_1 = L_3^{'}L_2^{'}L_1^{'}P_3P_2P_1, \tag{4.2.4}$$

với $L_k^{'}$ bằng với L_k nhưng với các phần tử trên đường chéo phụ đổi chổ cho nhau. Để chính xác, định nghĩa

$$L_3^{'} = L_3,\ L_2^{'} = P_3L_2P_3^{-1},\ L_1^{'} = P_3P_2L_1P_2^{-1}P_3^{-1}.$$

Vì mỗi định nghĩa này chỉ áp dụng các hoán vị P_j với $j > k$ tới L_k nên dễ dàng kiểm tra $L_k^{'}$ có cấu trúc giống như L_k. Việc tính tích của các ma trận $L_k^{'}$ đưa ra

$$L_3^{'}L_2^{'}L_1^{'}P_3P_2P_1 = L_3(P_3L_2P_3^{-1})(P_3P_2L_1P_2^{-1}P_3^{-1})P_3P_2P_1 = L_3P_3L_2P_2L_1P_1$$

như trong (4.2.4).

Tổng quát, cho ma trận $m \times m$, phân tích (4.2.1) được cung cấp bởi khử Gauss với quay từng phần có thể được viết dưới dạng

$$(L_{m-1}^{'} \dots L_2^{'}L_1^{'})(P_{m-1} \dots P_2P_1)A = U, \tag{4.2.5}$$

với $L_k^{'}$ được xác định bởi

$$L_k^{'} = P_{m-1} \dots P_{k+1}L_kP_{k+1}^{-1} \dots P_{m-1}^{-1}. \tag{4.2.6}$$

Tích của các ma trận $L_k^{'}$ là ma trận tam giác dưới đơn vị và dễ dàng lấy nghịch đảo bằng việc lấy phủ định các phần tử trên đường chéo phụ, như trong khử Gauss không có quay. Viết $L = (L_{m-1}^{'} \dots L_2^{'}L_1^{'})^{-1}$ và $P = P_{m-1} \dots P_2P_1$, ta có

$$PA = LU. \tag{4.2.7}$$

Tổng quát, ma trận vuông bất kỳ A, suy biến hoặc không suy biến, có một phân tích (4.2.7), với P là một ma trận hoán vị, L là ma trận tam giác dưới đơn vị với các phần tử tam giác dưới nhỏ hơn bằng 1, và U là ma trận tam giác trên.

Công thức phổ biến (4.2.7) có một giải thích đơn giản. Khử Gauss với quay từng phần là tương đương với thủ tục theo sau:

1. Hoán vị các dòng của A theo P.

Thuật toán 4.2 Khử Gauss với partial pivoting

```
U = A, L = I, P = I
for k = 1 to m − 1 do
    Chọn i ⩾ k để cực đại hóa |u_ik|
    u_{k,1:k−1} ↔ u_{i,k:m} (đổi chỗ 2 dòng với nhau)
    l_{k,1:k−1} ↔ l_{i,1:k−1}
    p_{k,:} ↔ p_{i,:}
    for j = k + 1 to m do
        l_jk = u_jk/u_kk
        u_{j,k:m} = u_{j,k:m} − l_jk u_{k,k:m}
    end for
end for
```

2. Áp dụng khử Gauss không quay cho PA.

Quay từng phần không được tiến hành trong thực hành vì P không được biết trước lúc đầu.

Dưới đây là thuật toán.

Thuật toán này yêu cầu số phép toán dấu chấm động giống (4.1.8) như khử Gauss không quay, cụ thể là $\frac{2}{3}m^3$. Như với Thuật toán 4.1, sử dụng bộ nhớ máy tính có thể được cực tiểu hóa nếu được miêu tả bằng việc viết chồng lên U và L thành mảng tương tự lưu trữ trong A.

Trong thực hành, P không được biểu diễn một cách chính xác như là một ma trận. Các dòng được đổi chỗ cho nhau tại mỗi bước, hoặc thông qua một vector hoán vị.

4.2.5 Quay đầy đủ

Trong quay đầy đủ, sự lựa chọn các pivot lấy một số lượng thời gian đáng kể. Trong thực hành, điều này hiếm khi được làm bởi vì sự thực thi trong tính ổn định là mép biên.

Trong dạng ma trận, quay đầy đủ đưa đến mỗi bước khử với một ma trận hoán vị P_k của các dòng được áp dụng trong vế trái và cũng là hoán vị Q_k của các cột được áp dụng trong vế phải:

$$L_{m-1}P_{m-1}\dots L_2P_2L_1P_1AQ_1Q_2\dots Q_{m-1} = U. \tag{4.2.8}$$

Nếu L_k' được xác định như trong (4.2.6) (các hoán vị cột không được bao gồm) thì

$$(L_{m-1}'\dots L_2'L_1')(P_{m-1}\dots P_2P_1)A(Q_1Q_2\dots Q_{m-1}) = U. \tag{4.2.9}$$

Đặt $L = (L_{m-1}'\dots L_2'L_1')^{-1}, P = P_{m-1}\dots P_2P_1)$, và $Q = Q_1Q_2\dots Q_{m-1}$, ta được

$$PAQ = LU. \tag{4.2.10}$$

4.3 Tính ổn định của khử Gauss

4.3.1 Tính ổn định và kích thước của L và U

Phân tích tính ổn định của khử Gauss với quay từng phần là phức tạp và nó đã là một điểm khó trong giải tích số từ những năm 1950.

Trong (4.1.9), ta cho một ví dụ với ma trận 2×2 mà trong đó khử Gauss không quay là không ổn định. Thừa số L có một phận tử kích thước 20^{20}. Cố gắng giải một hệ thống các

phương trình dựa vào L đưa ra các sai số làm tròn tương đối bậc $\epsilon_{machine}$. Do đó sai số tuyệt đối bậc $\epsilon_{machine} \times 10^{20}$.

Tính không ổn định trong khử Gauss- quay hoặc không quay- chỉ có thể xuất hiện nếu một hoặc cả hai thừa số L và U là tương đối lớn với kích thước của A. Do đó, mục đích của quay là để chắc chắn rằng L và U là không quá lớn. Miễn là tất cả các số lượng trung gian xuất hiện cho tới khi khử là kích thước dễ sử dụng, các sai số làm tròn mà chúng phát sinh là rất nhỏ, và thuật toán là ổn định ngược.

Định lý theo sau làm ý tưởng này rõ ràng.

Định lý 4.3.1 *Cho phân tích $A = LU$ của một ma trận không suy biến $A \in \mathbb{C}^{m\times m}$ được tính toán bởi khử Gauss không quay (Thuật toán 4.1) trong một máy tính thỏa các tiên đề (3.2.5) và (3.2.7). Nếu A có một phân tích LU thì với mọi $\epsilon_{machine}$ đủ nhỏ, phân tích hoàn thành một cách đầy đủ trong số học dấu chấm động (không có các pivot 0), và các ma trận tính được $\tilde{L}$ và $\tilde{U}$ thỏa mãn*

$$\tilde{L}\tilde{U} = A + \delta A, \ \frac{\|\delta A\|}{\|L\|\|U\|} = O(\epsilon_{machine}) \tag{4.3.1}$$

với $\delta A \in \mathbb{C}^{m\times n}$ bất kì.

Nếu $\|L\|\|U\| = O(\|A\|)$ thì (4.3.1) khẳng định rằng khử Gauss là ổn định ngược. Nếu $\|L\|\|U\| \neq O(\|A\|)$ thì ta phải mong đợi không ổn định ngược.

Cho khử Gauss không quay, cả L và U có thể là lớn không giới hạn. Thuật toán đó là không ổn định bởi chuẩn bất kì.

4.3.2 Các thừa số tăng

Xét khử Gauss với quay từng phần. Bởi vì mỗi sự lựa chọn pivot bao gồm tối đa hóa trên một cột, thuật toán này đưa ra ma trận L với các phần tử bên dưới đường chéo có giá trị tuyệt đối nhỏ hơn hoặc bằng 1. Điều này kéo theo $\|L\| = O(1)$ trong chuẩn bất kì. Do đó, cho khử Gauss với quay từng phần, (4.3.1) giảm thành điều kiện $\dfrac{\|\delta A\|}{\|U\|} = O(\epsilon_{machine})$. Ta kết luận rằng thuật toán là ổn định ngược được đưa ra $\|U\| = O(\|A\|)$.

Khử Gauss giảm một ma trận đầy đủ A thành một ma trận tam giác trên U. Đặc biệt, cho *thừa số tăng* cho A được xác định như là tỉ số

$$\rho = \frac{\max_{i,j} |u_{ij}|}{\max_{i,j} |a_{ij}|}. \tag{4.3.2}$$

Nếu ρ bậc 1thì quá trình khử là ổn định. Nếu ρ lớn hơn bậc 1, ta phải mong đợi tính không ổn định. Đặc biệt, vì $\|L\| = O(1)$, và vì (4.3.2) kéo theo $\|U\| = O(\rho\|A\|)$, kết quả theo sau là một hệ quả của Định lý 4.3.1.

Định lý 4.3.2 *Cho phân tích $PA = LU$ của một ma trận $A \in \mathbb{C}^{m\times m}$ được tính bởi khử Gauss với quay từng phần (Thuật toán 4.2) trong một máy tính thỏa mãn các tiên đề (3.2.5) và (3.2.7). Khi đó các ma trận được tính $\tilde{P}, \tilde{L}$ và $\tilde{U}$ thỏa mãn*

$$\tilde{L}\tilde{U} = \tilde{P}A + \delta A, \ \frac{\|\delta A\|}{\|A\|} = O(\rho\epsilon_{machine}) \tag{4.3.3}$$

với $\delta A \in \mathbb{C}^{m\times n}$ bất kì, ρ là thừa số tăng của A. Nếu $|l_{ij}| < 1$ với $i > j$ (không phụ thuộc vào sự lựa chọn các pivot trong số học chính xác) thì $\tilde{P} = P$ với mọi $\epsilon_{machine}$ đủ nhỏ.

Theo Định lý 4.3.2 và định nghĩa 3.3.5 của tính ổn định ngược, khử Gauss ổn định ngược nếu $\rho = O(1)$ với tất cả các ma trận có số chiều được cho là m, và trường hợp khác là không.

4.3.3 Tính không ổn định trong trường hợp xấu nhất

Cho các ma trận A nào đó, mặc dù các hiệu quả thuận lợi của quay, ρ đưa ra là lớn. Ví dụ, giả sử A là ma trận

$$A = \begin{bmatrix} 1 & & & & 1 \\ -1 & 1 & & & 1 \\ -1 & -1 & 1 & & 1 \\ -1 & -1 & -1 & 1 & 1 \\ -1 & -1 & -1 & -1 & 1 \end{bmatrix}. \tag{4.3.4}$$

Tại bước đầu tiên, không quay diễn ra, nhưng các phần tử $2, 3, \ldots, m$ trong cột cuối cùng được làm gấp đôi từ 1 tới 2. Sự xuất hiện làm gấp đôi khác tại mỗi bước khử sau đó xảy ra. Tại bước cuối ta có

$$U = \begin{bmatrix} 1 & & & & 1 \\ & 1 & & & 2 \\ & & 1 & & 4 \\ & & & 1 & 8 \\ & & & & 16 \end{bmatrix}. \tag{4.3.5}$$

Cuối cùng, phân tích $PA = LU$

$$\begin{bmatrix} 1 & & & & 1 \\ -1 & 1 & & & 1 \\ -1 & -1 & 1 & & 1 \\ -1 & -1 & -1 & 1 & 1 \\ -1 & -1 & -1 & -1 & 1 \end{bmatrix} = \begin{bmatrix} 1 & & & & \\ -1 & 1 & & & \\ -1 & -1 & 1 & & \\ -1 & -1 & -1 & 1 & \\ -1 & -1 & -1 & -1 & 1 \end{bmatrix} \begin{bmatrix} 1 & & & & 1 \\ & 1 & & & 2 \\ & & 1 & & 4 \\ & & & 1 & 8 \\ & & & & 16 \end{bmatrix}. \tag{4.3.6}$$

Cho ma trận 5×5 này, thừa số tăng là $\rho = 16$. Cho ma trận $m \times m$ có dạng tương tự, $\rho = 2^{m-1}$.

Thừa số tăng bậc 2^m tương ứng với sự hao hụt trong bậc m bit của độ chính xác, mà nó là thê thảm cho một tính toán thực hành. Vì một máy tính điển hình biểu diễn số dấu chấm động chỉ với 64 bit nên sự hao hụt m bit của độ chính xác là không chấp nhận được cho các tính toán thực tế.

Điều này đưa chúng ta tới điểm khó khăn. Ở đây, trong thảo luận khử Gauss với quay các định nghĩa của tính ổn định đưa ra trong mục 3.3 thất bại.

Theo các định nghĩa, tất cả các vấn đề đó trong việc xác định tính ổn định hoặc tính ổn định ngược là tồn tại một chặn nào đó đều có thể dùng được tới tất cả các ma trận *cho mỗi chiều được cố định m*.

Ở đây, với mỗi m, ta có một chặn đều bao gồm hằng số 2^{m-1}. Do đó, theo các định nghĩa của chúng ta, khử Gauss là ổn định ngược.

Định lý 4.3.3 *Theo các định nghĩa trong mục 3.3, khử Gauss với quay từng phần là ổn định ngược.*

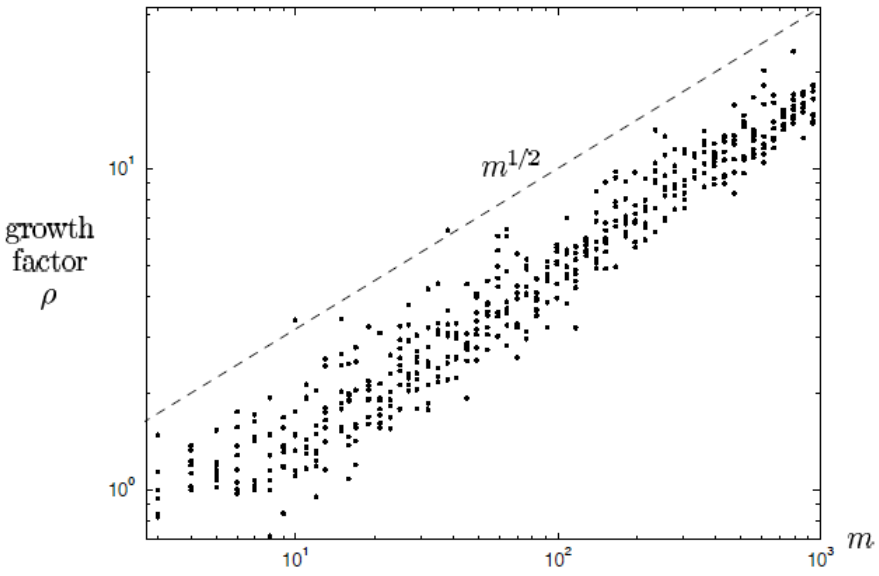
Khử Gauss cho các ma trận nào đó là không ổn định, khi ta có thể được thừa nhận bởi các thực thi số với Matlab, Linpack, Lapack, hoặc các gói phần mềm nổi tiếng hoàn hảo.

4.3.4 Tính ổn định trong thực hành

Mặc dù các ví dụ giống (4.3.4), khử Gauss với quay từng phần là hoàn toàn ổn định trong thực hành. Các thừa số lớn U giống (4.3.5) dường như không bao giờ xuất hiện trong các ứng dụng thực tế. Trong 50 năm tính toán, không bài toán ma trận nào kích thích tính không ổn định được biết đã xuất hiện dưới các trường hợp tự nhiên.

Ta có thể học nhiều hơn về hiện tượng này bằng việc xét các ma trận ngẫu nhiên. Chúng có tất cả các loại tính chất đặc biệt, và nếu ta cố gắng để miêu tả chúng như các ví dụ ngẫu nhiên từ phân phối bất kì, nó sẽ phải là một phân bố lạ lùng. Dĩ nhiên nó sẽ là vô lý để mong đợi rằng phân phối đặc biệt nào đó của các ma trận ngẫu nhiên nên phù hợp với cách xử lý của các ma trận xuất hiện trong thực hành trong cách định lượng gần.

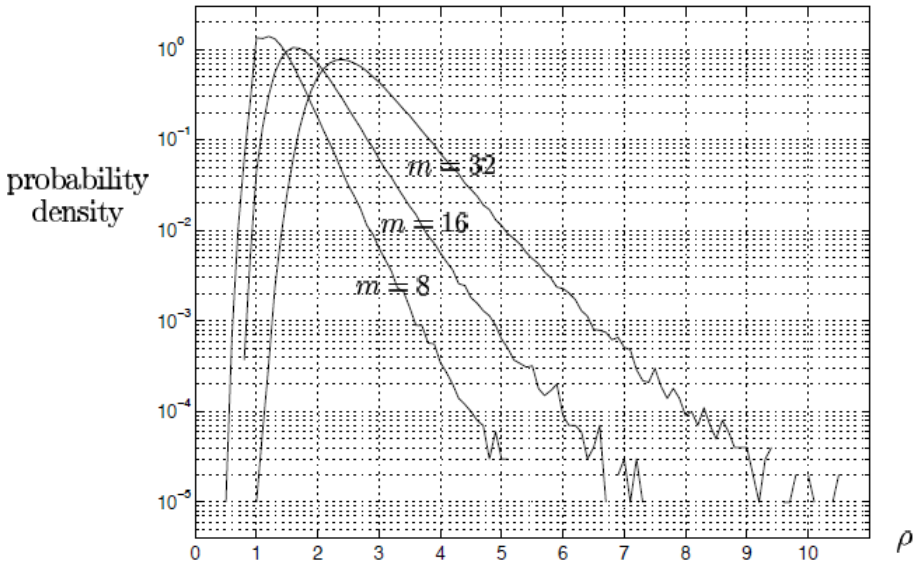
Tuy nhiên, hiện tượng được giải thích không là vấn đề của các con số rõ ràng. Các ma trận với các thừa số tăng lớn là hiếm khi loại trừ nhau trong các ứng dụng. Nếu ta có thể cho thấy rằng chúng là hiếm khi loại trừ nhau giữa các ma trận ngẫu nhiên trong lớp định nghĩa tốt nào đó, các kỹ thuật được bao gồm phải chắc chắn là giống nhau. Đối số không phụ thuộc vào một độ đo của việc thỏa thuận "loại trừ lẫn nhau" với cái khác tới thừa số đặc biệt bất kì như 2 hoặc 10 hoặc 100.



Hình 4.1: Các thừa số tăng cho khử Gauss với quay từng phần được áp dụng cho 496 ma trận ngẫu nhiên (các phần tử độc lập với nhau và được phân phối chuẩn) của các số chiều thay đổi. Kích thước của ρ là bậc $m^{1/2}$, ít hơn nhiều giá trị có thể lớn nhất 2^{m-1}.

Hình 4.1 và hình 4.2 đưa ra các thực thi với các ma trận ngẫu nhiên: mỗi phần tử là một mẫu độc lập từ phân phối chuẩn thực tế với trung vị 0 và phương sai chuẩn $m^{1/2}$. Trong Hình 4.1, một sự tập hợp của các ma trận ngẫu nhiên của các số chiều thay đổi đã được phân tích và các thừa số tăng đưa ra như một đồ thị khuếch tán. Chỉ 2 trong số các ma trận cho một thừa số tăng lớn như là $m^{1/2}$. Trong Hình 4.2, các kết quả của việc phân tích một triệu ma trận có số chiều $m = 8, 16, 32$. Ở đây, các thừa số tăng đã được tập hợp trong các ngăn bề rộng 0.2 và các dữ liệu kết quả vẽ như một phân bố trù mật xác suất. Trù mật xác suất của các thừa số tăng xuất hiện để giảm các lũy thừa với kích thước. Giữa 3 triệu ma trận này, vì thừa số tăng cao nhất có thể đã là 2,147,483,648, số lớn nhất được đếm một cách chính xác là 11.99.

Các kết quả tương tự đạt được với các ma trận ngẫu nhiên được xác định bởi các phân phối xác suất khác, như là các phần tử được phân phối đều trong [-1,1]. Nếu bạn lấy 1 tỷ ma trận ngẫu nhiên, bạn sẽ hầu hết chắc chắn không tìm thấy một ma trận mà khử Gauss là không ổn định.



Hình 4.2: Các phân phối trù mật xác suất cho các thừa số tăng của các ma trận ngẫu nhiên có số chiều $m = 8, 16, 32$, được dựa vào các kích thước mẫu của một triệu con số cho mỗi chiều. Sự trù mật xuất hiện để giảm số mũ với ρ. Hình răng cưa gần cuối mỗi đường cong là một thành phần lạ của các kích thước mẫu hữu hạn.

4.3.5 Giải thích

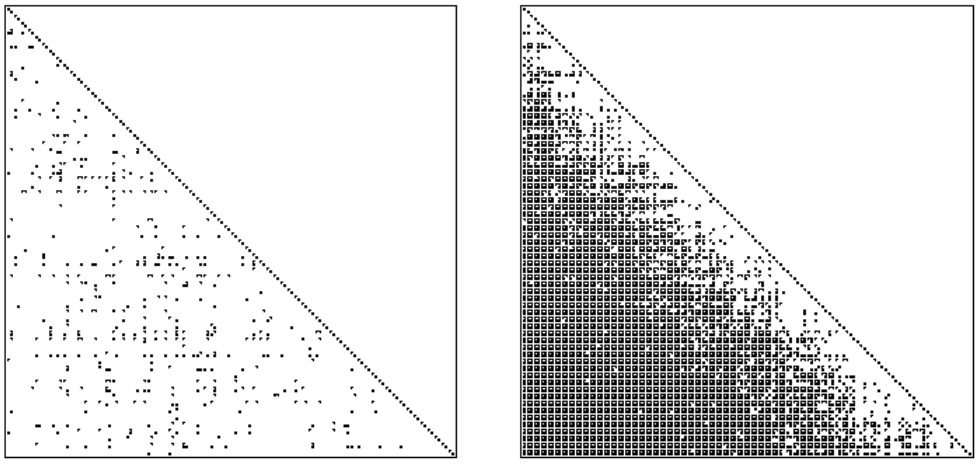
Nếu $PA = LU$ thì $U = L^{-1}PA$. Nếu khử Gauss là không ổn định khi được áp dụng tới ma trận A, kéo theo rằng ρ là lớn, khi đó L^{-1} cũng phải là quá lớn. Bây giờ, khi nó xảy ra, các ma trận tam giác ngẫu nhiên hướng tới các ma trận khả nghịch lớn, lũy thừa lớn như là một hàm của số chiều m. Đặc biệt, điều này là đúng cho các ma trận đường chéo ngẫu nhiên của dạng được phát biểu bởi khử Gauss với quay từng phần, với 1 nằm trên đường chéo và các phần tử có trị tuyệt đối nhỏ hơn bằng 1 bên dưới đường chéo.

Khi khử Gauss được áp dụng tới các ma trận ngẫu nhiên A thì các thừa số kết quả L là bất kì nhưng ngẫu nhiên. Các sự tương quan xuất hiện giữa các dấu của các phần tử của L mà chúng đưa ra các ma trận điều kiện tốt đặc biệt. Một phần tử đặc trưng của L^{-1}, xa hơn là các lũy thừa lớn, thường là nhỏ hơn 1 trong dấu giá trị tuyệt đối. Hình **??** đưa ra sự rõ ràng của hiện tượng này được dựa vào một ma trận đơn (nhưng đặc trưng) có số chiều $m = 128$.

Do đó ta đến câu hỏi: vì sao các ma trận L được đưa ra bằng khử Gauss hầu hết không bao giờ có ma trận nghịch đảo lớn?

Câu trả lời nằm trong sự xét các không gian cột. Vì U là ma trận tam giác trên và $PA = LU$, không gian cột của PA và L là giống nhau. Nghĩa là cột đầu tiên của không gian sinh PA giống cột đầu tiên của L, 2 cột đầu tiên của không gian sinh PA giống với 2 cột đầu tiên của L, ... Nếu A là ngẫu nhiên, các không gian cột của nó được định hướng một cách ngẫu nhiên, và nó theo sau rằng sự tương tự phải đúng là các không gian cột của $P^{-1}L$. Tuy nhiên, điều kiện này là không tương thích L^{-1} lớn. Nó có thể được cho thấy rằng nếu L^{-1} là lớn, khi đó các không gian cột của L, hoặc của các hoán vị bất kì $P^{-1}L$ phải được làm xuyên đi trong một mô hình mà nó xa hơn ngẫu nhiên.

Hình 4.4 cho sự rõ ràng của điều này. Hình này cho thấy "nơi năng lượng là" trong các không gian cột liên tiếp của 2 ma trận giống nhau như trong Hình 4.3. Thiết bị cho việc làm



Hình 4.3: Cho A là ma trận ngẫu nhiên 128×128 với phân tích $PA = LU$. Trong vế trái, L^{-1} cho thấy rằng: các phần tử biểu diễn các dấu chấm với độ dài lớn hơn 1. Trong vế phải hình tương tự cho $\tilde{L}^{-1}$, với $\tilde{L}$ là giống như L ngoại trừ các dấu của các phần tử trên đường chéo phụ của nó đã được ngẫu nhiên. Khử Gauss hướng đến đưa ra các ma trận L mà chúng là điều kiện tốt đặc biệt.

này là một *điển hình* Q, được xác định bởi các lệnh của Matlab

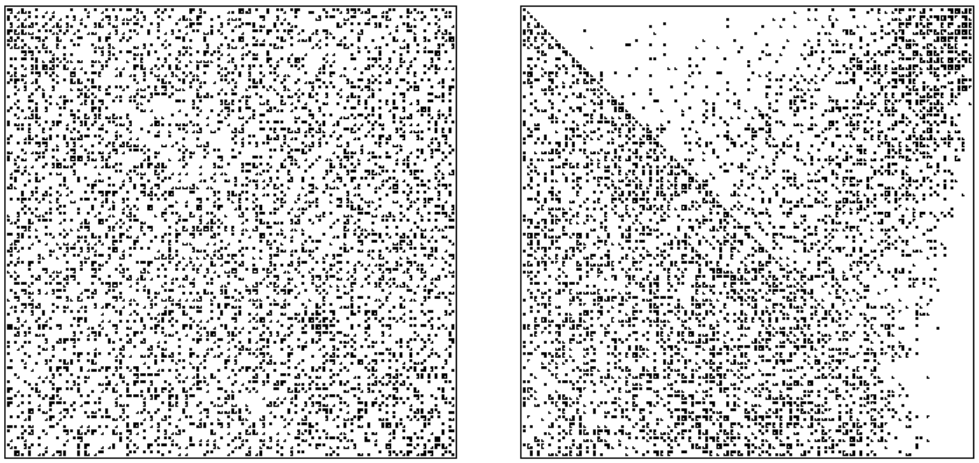
$$[Q, R] = qr(A),\ spy(abs(Q)) > 1/sqrt(m)). \tag{4.3.7}$$

Đầu tiên các lệnh này tính phân tích QR của ma trận A, khi đó đồ thị một dấu chấm tại mỗi vị trí của Q tương ứng với một phần tử lớn hơn phương sai chuẩn, $m^{-1/2}$. Hình miêu tả cho một ma trận ngẫu nhiên A, mặc dù sau đó các hoán đổi dòng với nhau thành dạng PA, các không gian cột được định hướng một cách ngẫu nhiên, trong khi với ma trận A cho một thừa số tăng lớn, các định hướng là rất xa từ ngẫu nhiên. Nó giống như là bằng việc xác định số lượng đối số này, nó có thể được chứng minh rằng các thừa số tăng lớn hơn bậc $m^{1/2}$ là hiếm giữa các ma trận ngẫu nhiên trong ý nghĩ mà cho $\alpha > 1/2$ bất kì và $M > 0$, xác suất của sự kiện $\rho > m^{\alpha}$ là nhỏ hơn m^{-M} cho mọi m đủ lớn. Tuy nhiên, như khi việc viết điều này thì một định lý như vậy không được chứng minh.

Ta hãy tóm tắt tính ổn định của khử Gauss với quay từng phần. Thuật toán này là không ổn định cao cho các ma trận A bất kì. Tuy nhiên, cho tính không ổn định xuất hiện thì các không gian cột của A phải được làm xuyên đi trong dạng rất đặc biệt, mà nó là hiếm trong ít nhất một lớp các ma trận ngẫu nhiên. Các thập kỉ của thực nghiệm tính toán đã đề nghị rằng các ma trận mà các không gian cột của chúng được làm xuyên đi trong dạng này xuất hiện rất hiếm trong các ứng dụng.

4.4 Phân tích Cholesky

Các ma trận xác định dương Hermit có thể được phân tích thành các thừa số tam giác nhanh như các ma trận tổng quát. Thuật toán tiêu chuẩn cho điều này là phân tích Cholesky, một biến thể của khử Gauss mà nó tính toán trong cả bên trái và bên phải của ma trận A trong 1 lần, lưu trữ và khai thác ma trận đối xứng.



Hình 4.4: Trong vế trái, ma trận ngẫu nhiên A sau khi hoán vị thành dạng PA, hoặc tương đương, thừa số L. Trong vế phải, ma trận $\tilde{L}$ với các dấu được ngẫu nhiên. Các không gian cột của $\tilde{L}$ được làm xuyên đi trong kiểu không giống lũy thừa để xuất hiện trong các loại đặc trưng của các ma trận ngẫu nhiên.

4.4.1 Các ma trận xác định dương Hermit

Một ma trận thực $A \in \mathbb{R}^{m\times m}$ là *đối xứng* nếu nó có các phần tử giống nhau bên dưới đường chéo cũng như bên trên đường chéo: $a_{ij} = a_{ji}$ với mọi i, j, do đó $A = A^T$. Một ma trận như vậy thỏa mãn $x^T Ay = y^T Ax$ với mọi vector $x, y \in \mathbb{R}^m$.

Với một ma trận phức $A \in \mathbb{C}^{m\times m}$, tính chất tương tự mà A là *Hermit*. Một ma trận hermit có các phần tử bên dưới đường chéo là các liên hợp phức của các phần tử ở bên trên đường chéo: $a_{ij} = \overline{a_{ij}}$, do đó $A = A^*$. (Các định nghĩa này xuất hiện trong mục trước)

Một ma trận hermit A thỏa mãn $x^*Ay = \overline{y^*Ax}$ với mọi $x, y \in \mathbb{C}^m$. Đặc biệt điều này nghĩa là cho $x \in \mathbb{C}^m$ bất kì, x^*Ax là thực. Nếu thêm $x^*Ax > 0$ với mọi $x \neq 0$, thì A được nói là *xác định dương hermit* (hoặc đôi khi là *xác định dương*). Nhiều ma trận xuất hiện trong các hệ thống vật lý là xác định dương hermit bởi vì các luật vật lý cơ bản.

Nếu A là một ma trận xác định dương hermit $m \times m$ và X là một ma trận $m \times n$ hạng đầy đủ với $m \geqslant n$ thì ma trận X^*AX cũng là xác định dương hermit. Nó là hermit bởi vì $(X^*AX)^* = X^*A^*X = X^*AX$. Nó là xác định dương bởi vì, cho vector $x \neq 0$ bất kì, ta có $Xx \neq 0$ và do đó $x^*(X^*AX)x = (Xx)^*A(Xx) > 0$. Bằng việc chọn X để làm một ma trận $m \times n$ với 1 trong mỗi cột và 0 ở những nơi khác, ta có thể viết ma trận con chính $n \times n$ bất kì của A trong dạng X^*AX. Do đó, ma trận con chính bất kì của A phải là xác định dương. Đặc biệt, mọi phần tử đường chéo của A là một số thực dương.

Các trị riêng của một ma trận xác định dương hermit cũng là các số thực dương. Nếu $Ax = \lambda x$ với $x \neq 0$, ta có $x^*Ax = \lambda x^*x > 0$ và do đó $\lambda > 0$. Ngược lại, nó có thể được chứng minh rằng nếu một ma trận hermit có tất cả các trị riêng dương, thì nó là xác định dương.

Các vector riêng tương ứng với các trị riêng phân biệt của một ma trận hermit là trực giao. Giả sử $Ax_1 = \lambda_1 x_1$ và $Ax_2 = \lambda_2 x_2$ với $\lambda_1 \neq \lambda_2$. Khi đó

$$\lambda_2 x_1^* x_2 = x_1^* A x_2 = \overline{x_2^* A x_1} = \overline{\lambda_1 x_2^* x_1} = \lambda_1 x_1^* x_2,$$

nên $(\lambda_1 - \lambda_2)x_1^* x_2 = 0$. Vì $\lambda_1 \neq \lambda_2$, ta có $x_1^* x_2 = 0$.

4.4.2 Khử Gauss đối xứng

Bây giờ ta trở lại bài toán phân tích một ma trận xác định dương hermit thành các thừa số tam giác. Để bắt đầu, xét một bước đơn của khử Gauss được áp dụng tới một ma trận hermit A với 1 nằm ở vị trí bên trái trên:

$$A = \begin{bmatrix} 1 & w^* \\ w & K \end{bmatrix} = \begin{bmatrix} 1 & 0 \\ w & I \end{bmatrix} \begin{bmatrix} 1 & w^* \\ 0 & K - ww^* \end{bmatrix}.$$

Như được miêu tả trong mục 4.1, các số 0 đã được đưa vào trong cột đầu tiên của ma trận bằng một phép toán tam giác dưới cơ bản trong vế trái mà nó trừ các bội của dòng đầu tiên từ các dòng sau đó.

Bây giờ khử Gauss sẽ tiếp tục giảm thành dạng tam giác bằng việc đưa các số 0 trong cột thứ hai. Tuy nhiên, để giữ sự đối xứng, đầu tiên phân tích Cholesky đưa các số 0 trong dòng đầu tiên để phù hợp với số 0 vừa được đưa ra trong cột đầu tiên. Ta có thể làm điều này bằng phép toán tam giác trên bên phải mà nó trừ các bội của cột đầu tiên từ các cột sau đó:

$$\begin{bmatrix} 1 & w^* \\ 0 & K - ww^* \end{bmatrix} = \begin{bmatrix} 1 & 0 \\ 0 & K - ww^* \end{bmatrix} \begin{bmatrix} 1 & w^* \\ 0 & I \end{bmatrix}.$$

Chú ý rằng phép toán tam giác trên này một cách chính xác là phụ hợp của phép toán tam giác dưới mà ta thường đưa các số 0 trong cột đầu tiên.

Kết hợp với các phép toán ở trên, ta thấy rằng ma trận A đã được phân tích thành 3 số hạng:

$$A = \begin{bmatrix} 1 & w^* \\ w & K \end{bmatrix} = \begin{bmatrix} 1 & 0 \\ w & I \end{bmatrix} \begin{bmatrix} 1 & 0 \\ 0 & K - ww^* \end{bmatrix} \begin{bmatrix} 1 & w^* \\ 0 & I \end{bmatrix}. \tag{4.4.1}$$

Ý tưởng của phân tích Cholesky là để tiếp tục quá trình này, việc làm các số 0 trong một cột và một dòng của A một cách đối xứng cho tới khi nó được giảm xuống thành ma trận đơn vị.

4.4.3 Phân tích Cholesky

Cho $a_{11} > 0$ bất kì, không phải $a_{11} = 1$. Tổng quát hóa của (4.4.1) được thực hiện bằng việc hiệu chỉnh một vài phần tử của R_1 bằng một thừa số của $\sqrt{a_{11}}$. Cho $\alpha = \sqrt{a_{11}}$ và quan sát:

$$\begin{aligned} A &= \begin{bmatrix} a_{11} & w^* \\ w & K \end{bmatrix} \\ &= \begin{bmatrix} \alpha & 0 \\ w/\alpha & I \end{bmatrix} \begin{bmatrix} 1 & 0 \\ 0 & K - ww^*/a_{11} \end{bmatrix} \begin{bmatrix} \alpha & w^*/\alpha \\ 0 & I \end{bmatrix} = R_1^* A_1 R_1. \end{aligned}$$

Điều này bước cơ sở mà nó được áp dụng lặp lại trong phân tích Cholesky. Nếu phần tử bên trái phía trên của ma trận con $K - ww^*/a_{11}$ là dương, công thức tương tự có thể được sử dụng để phân tích nó; khi đó ta có $A_1 = R_2^* A_2 R_2$ và do đó $A = R_1^* R_2^* A_2 R_2 R_1$. Quá trình được tiếp tục giảm thành góc dưới cùng bên phải, cuối cùng cho chúng ta một phân tích

$$A = \underbrace{R_1^* R_2^* \dots R_m^*}_{R^*} \underbrace{R_m \dots R_2 R_1}_{R}. \tag{4.4.2}$$

Phương trình này có dạng

$$A = R^* R, \; r_{ij} > 0, \tag{4.4.3}$$

với R là ma trận tam giác trên. Sự giảm loại này của một ma trận xác định dương hermit được biết như là môt *phân tích Cholesky*.

Sự miêu tả ở trên đưa ra một điểm đu đưa. Làm thế nào ta biết rằng phần tử trái trên của ma trận con $K - ww^*/a_{11}$ là dương? Câu trả lời là nó phải là dương bởi vì $K - ww^*/a_{11}$ là xác định dương, vì nó là ma trận con chính trái trên $(m-1) \times (m-1)$ của ma trận xác định dương $R_1^{-*} A R_1^{-1}$. Bằng qui nạp, đối số tương tự cho thấy rằng tất cả các ma trận con A_j xuất hiện trong tiến trình phân tích là xác định dương, và do đó quá trình không bị phá hủy. Ta có thể hình thức hóa kết luận này như sau.

Định lý 4.4.1 *Mọi ma trận xác định dương hermit $A \in \mathbb{C}^{m\times m}$ có một phân tích Cholesky (4.4.3) duy nhất.*

Chứng minh Phân tích tồn tại vì thuật toán không thể bị phá hủy. Thật vậy, thuật toán cũng thiết lập sự duy nhất. Tại mỗi bước (4.4.2), giá trị $\alpha = \sqrt{a_{11}}$ được xác định bởi dạng của phân tích R^*R, và α được xác định, dòng đầu tiên của R_1^* cũng được xác định như vậy. Vì các con số tương tự được xác định tại mỗi bước của sự giảm, phân tích hoàn toàn là duy nhất.

4.4.4 Thuật toán

Khi phân tích Cholesky được thực thi, chỉ phân nửa ma trận đang được tính toán cho các nhu cầu được trình bày một cách rõ ràng. Sự rút gọn này cho phép phân nửa phép toán số học được cho phép. Ma trận đầu vào A biểu diễn một nửa siêu đường chéo của ma trận xác định dương hermit $m \times m$ được phân tích. (Trong phần mềm thực hành, hệ thống lưu trữ được nén có thể được sử dụng để tránh việc tàn phá phân nửa phần tử của một mảng vuông.) Ma trận đầu ra R biểu diễn thừa số tam giác trên cho $A = R^*R$. Mỗi bước lặp bên ngoài tương ứng với phân tích cơ bản đơn: phần tam giác trên của ma trận con $R^*_{k:m,k:m}$ đưa ra phần siêu đường chéo của ma trận hermit được lưu trữ tại bước thứ k.

Thuật toán 4.3 Phân tích Cholesky

```
1: R = A
2: for k = 1 to m do
3:     for j = k + 1 to m do
4:         R_{j,j:m} = R_{j,j:m} − R_{k,j:m} conj(R_{kj}) / R_{kk}
5:         R_{k,k:m} = R_{k,k:m} / sqrt(R_{kk})
6:     end for
7: end for
```

4.4.5 Đếm số phép toán

Số học được làm trong phân tích Cholesky được chi phối bởi vòng lặp bên trong. Sự thực thi đơn của dòng lệnh

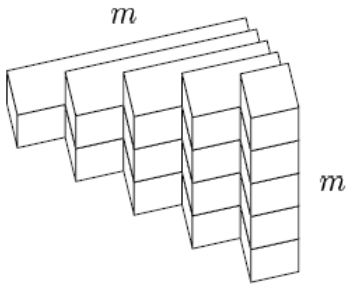
$$R_{j,j:m} = R_{j,j:m} - R_{k,j:m}\overline{R_{kj}}/R_{kk}$$

cần 1 phép chia, $m - j + 1$ phép nhân và $m - j + 1$ phép trừ cho tổng $\sim 2(m - j)$ phép toán dấu chấm động. Sự tính toán này được lặp lại một lần với mỗi j từ $k + 1$ tới m, và vòng lặp đó được lặp lại với mỗi k từ 1 tới m. Tổng là

$$\sum_{k=1}^{m}\sum_{j=k+1}^{m} 2(m-j) \sim 2\sum_{k=1}^{m}\sum_{j=1}^{k} j \sim \sum_{k=1}^{m} k^2 \sim \frac{1}{3}m^3 \text{ phép toán dấu chấm động.}$$

Do đó, phân tích Cholesky chỉ bao gồm một nửa phép toán như khử Gauss, mà nó sẽ yêu cầu $\sim \frac{2}{3}m^3$ phép toán dấu chấm động để phân tích cùng ma trận.

Như thường lệ, đếm số phép toán cũng có thể được xác định bằng đồ thị. Với mỗi k, 2 phép toán dấu chấm động được thực hiện (một phép nhân và một phép trừ) tại mỗi vị trí của một lớp tam giác. Thuật toán hoàn toàn tương ứng với việc xếp chồng lên m lớp:



Khi $m \to \infty$, khối hội tụ về một tứ diện với thể tích $\frac{1}{6}m^3$. Vì mỗi đơn vị thể tích tương ứng với 2 phép toán dấu chấm động, ta thu được

$$\text{Phân tích Cholesky } \sim \frac{1}{3}m^3 \text{ phép toán dấu chấm động} \tag{4.4.4}$$

4.4.6 Tính ổn định

Thuật toán này thường là ổn định. Bằng trực giác, lý do mà các thừa số R không bao giờ có thể tăng lớn. Trong chuẩn 2, ví dụ ta có $\|R\| = \|R^*\| = \|A\|^{1/2}$ (chứng minh: SVD), và trong chuẩn p khác với $1 \leqslant p \leqslant \infty$, $\|R\|$ không thể khác $\|A\|^{1/2}$ bởi nhiều hơn một thừa số của $\sqrt{m}$. Do đó, số phần tử lớn hơn nhiều số phần tử của A có thể không bao giờ xuất hiện.

Chú ý: tính ổn định của phân tích Cholesky đạt được không cần cho quay bất kì. Bằng trực giác, ta có thể quan sát thấy rằng điều này là có liên quan tới tính chất của ma trận xác định dương hermit là trên đường chéo. Ví dụ, không khó để cho thấy rằng phần tử lớn nhất phải xuất hiện trên đường chéo, và tính chất này đưa vào các ma trận con xác định dương được xây dựng trong quá trình qui nạp (4.4.2).

Phân tích tính ổn định của quá trình Cholesky dẫn tới kết quả ổn định ngược như sau.

Định lý 4.4.2 *Cho $A \in \mathbb{C}^{m\times m}$ là xác định dương hermit, và cho một phân tích Cholesky của A được tính bởi Thuật toán 4.3 trong một máy tính thỏa mãn (3.2.5) và (3.2.7). Với $\epsilon_{machine}$ đủ nhỏ, quá trình này được bảo đảm để chạy hoàn toàn (nghĩa là, các phần tử r_{kk} khác 0 và âm sẽ xuất hiện), sinh ra một thừa số được tính $\tilde{R}$ mà nó thỏa mãn*

$$\tilde{R}^*\tilde{R} = A + \delta A, \ \frac{\|\delta A\|}{\|A\|} = O(\epsilon_{machine} \tag{4.4.5}$$

với $\delta A \in \mathbb{C}^{m\times m}$ bất kì.

Giống như nhiều thuật toán của phương pháp số trong đại số tuyến tính, thuật toán này sẽ trông tệ hơn nhiều nếu ta cố gắng thực hiện phân tích sai số tiến hơn là sai số ngược. Nếu A là điều kiện xấu thì $\tilde{R}$ sẽ không gần với R; Tốt nhất ta có thể nói là $\|\tilde{R} - R\|/\|R\| = O(\kappa(A)\epsilon_{machine})$. (Mặc khác, phân tích Cholesky tổng quát là bài toán điều kiện xấu.) Tích $\tilde{R}^*\tilde{R}$ thỏa mãn chặn sai số tốt hơn nhiều (4.4.5). Do đó các sai số được đưa ra trong $\tilde{R}$ bằng việc làm tròn là lớn nhưng "tương quan ranh mãnh" như ta thấy trong mục 3.4 cho phân tích QR.

4.4.7 Giải phương trình $Ax = b$

Nếu A là xác định dương hermit, cách chuẩn để giải một hệ thống các phương trình $Ax = b$ là bằng phân tích Cholesky. Thuật toán 4.3 giảm hệ thống thành $R^*Rx = b$, và khi đó ta giải 2 hệ thống tam giác liên tiếp: đầu tiên $R^*y = b$ với biến y không được biết, khi đó $Rx = y$ với biến x không được biết. Mỗi lời giải tam giác cần $\sim m^2$ phép toán dấu chấm động, nên việc làm tổng cộng là $\sim \frac{1}{3}m^3$ phép toán dấu chấm động.

Định lý 4.4.3 *Nghiệm của các hệ thống $Ax = b$ xác định dương hermit thông qua phân tích Cholesky (Thuật toán 4.3) là ổn định ngược, sinh ra một nghiệm được tính $\tilde{x}$ thõa mãn*

$$(A + \Delta A)\tilde{x} = b, \ \frac{\|\Delta A\|}{\|A\|} = O(\epsilon_{machine}) \tag{4.4.6}$$

với $\Delta A \in \mathbb{C}^{m\times m}$ bất kì.

Bài tập

1. Cho $A \in \mathbb{C}^{m\times m}$ là không suy biến. Chứng minh rằng A có một phân tích LU nếu và chỉ nếu với mọi k, $1 \leqslant k \leqslant m$, khối ở trên bên trái $A_{1:k,1:k}$ cấp $k \times k$ là không suy biến. Chứng minh phân tích LU này là duy nhất.

2. Giả sử ma trận A cấp $m \times m$ được viết thành dạng khối

$$\begin{bmatrix} A_{11} & A_{12} \\ A_{21} & A_{22} \end{bmatrix}$$

với A_{11} là ma trận cấp $n \times n$ và A_{22} là ma trận cấp $(m-n) \times (m-n)$. Giả sử A thỏa các điều kiện của bài tập 1.

   (a) Kiểm tra công thức

$$\begin{bmatrix} I \\ -A_{21}A_{11}^{-1}I \end{bmatrix} \begin{bmatrix} A_{11} & A_{12} \\ A_{21} & A_{22} \end{bmatrix} = \begin{bmatrix} A_{11} & A_{12} \\ & A_{22} - A_{21}A_{11}^{-1}A_{12} \end{bmatrix} \tag{4.4.7}$$

   cho khử khối A_{21}. Ma trận $A_{22} - A_{21}A_{11}^{-1}A_{12}$ là *phần bù Schur* của A.

   (b) Giả sử A_{21} được khử dòng bằng n bước của khử Gauss. Chưng minh rằng khối ở đỉnh bên phải cấp $(m-n) \times (m-n)$ của kết quả là $A_{22} - A_{21}A_{11}^{-1}A_{12}$.

3. Xét khử Gauss tiến hành với pivoting bởi các cột thay vì các dòng, đưa ra một phân tích $AQ = LU$, với Q là ma trận hoán vị.

   (a) Chứng minh rằng nếu A không suy biến thì phân tích như vậy luôn tồn tại.

   (b) Chứng minh rằng nếu A suy biến thì phân tích như vậy không tồn tại.

4. Khử Gauss có thể được sử dụng để tính A^{-1} của một ma trận không suy biến $A \in \mathbb{C}^{m\times m}$.

   (a) Miêu tả một thuật toán tính A^{-1} bằng việc giải hệ thống m phương trình và chứng minh rằng đếm số phép toán tiệm cận của nó là $8m^3/3$ phép toán dấu chấm động.

   (b) Miêu tả một biến thể của thuật toán mà nó giảm số phép toán xuống còn $2m^3$ phép toán dấu chấm động.

(c) Giả sử ta mong muốn giải hệ thống n phương trình $Ax_j = b_j$ hoặc $AX = B$, với $B \in \mathbb{C}^{m\times n}$. Đếm số phép toán tiệm cận (hàm của m và n) cho việc làm này từ phân tích LU và sự tính toán của A^{-1}?

5. Cho A là ma trận vuông không suy biến và cho $A = QR$ và $A^*A = U^*U$ lần lượt là phân tích QR và phân tích Cholesky, với sự chuẩn hóa thông thường $r_{jj}, u_{jj} > 0$. $R = U$?

6. Chứng minh rằng khử Gauss với quay từng phần được áp dụng cho ma trận bất kì $A \in \mathbb{C}^{m\times m}$ thì thừa số tăng (4.3.2) thỏa $\rho \leqslant 2^{m-1}$.

7. Cài đặt Thuật toán 4.2.

8. Cài đặt Thuật toán 4.3.

Chương 5

Trị riêng

5.1 Các bài toán trị riêng

5.1.1 Trị riêng và vector riêng

Cho $A \in \mathbb{C}^{m\times m}$ là một ma trận vuông. Một vector khác không $x \in \mathbb{C}^m$ là một *vector riêng* của A, và $\lambda \in \mathbb{C}$ là *trị riêng* tương ứng của nó, nếu

$$Ax = \lambda x. \tag{5.1.1}$$

Ý tưởng ở đây là tác động của một ma trận A vào một không gian con S của $\mathbb{C}^m$ thỉnh thoảng có thể tương tự phép nhân vô hướng. Khi điều này xảy ra, không gian con đặc biệt S được gọi là *không gian riêng*, và $x \in S$ khác không bất kì là một vector riêng.

Tập hợp tất cả các trị riêng của một ma trận A là *phổ* của A mà nó là một tập con của $\mathbb{C}$ được ký hiệu bởi $\Lambda(A)$.

Các bài toán trị riêng có đặc trưng rất khác với các bài toán bao gồm các hệ thống tuyến tính các phương trình vuông hoặc hình chữ nhật được thảo luận trong mục trước. Cho một hệ thống các phương trình, miền xác định của A có thể là một không gian và hạng có thể là một không gian khác. Trong ví dụ 1.2.3, A ánh xạ n vector của các hệ số đa thức thành m vector của các giá trị đa thức được lấy mẫu.Các bài toán trị riêng làm khả năng phán đoán chỉ khi các không gian hạng và miền xác định là giống nhau. Điều này phản ánh trong các ứng dụng, các trị riêng nói chung được sử dụng ở nơi mà một ma trận là được kết hợp lặp lại, hoặc rõ ràng như một lũy thừa A^k hoặc trong một dạng hàm như e^{tA}.

Nói chung, các trị riêng và các vector riêng là hữu ích với 2 lý do, một là thuật toán, một là vật lý. Theo thuật toán, phân tích trị riêng có thể làm đơn giản các lời giải của các bài toán nào đó bằng việc giảm một hệ thống được ghép đôi thành một sự lựa chọn của các bài toán vô hướng. Theo vật lý, phân tích trị riêng có thể đưa cái nhìn sâu vào xử lý của các hệ thống rút ra được chi phối bởi các phương trình tuyến tính. Các ví dụ quen thuộc nhất trong mục sau là nguyên cứu sự *cộng hưởng* và *tính ổn định* . Trong các trường hợp như vậy các trị riêng hướng tới tính hữu ích cho việc phân tích xử lý cho số lần t lớn.

5.1.2 Phân tích trị riêng

Một *phân tích trị riêng* của một ma trận vuông A là một sự phân tích

$$A = X\Lambda X^{-1} \tag{5.1.2}$$

với X là không suy biến và Λ là ma trận đường chéo.

Định nghĩa này có thể được viết lại

$$AX = X\Lambda, \tag{5.1.3}$$

đó là,

$$\begin{bmatrix} & & \\ & A & \\ & & \end{bmatrix} \left[\begin{array}{c|c|c|c} & & & \\ x_1 & x_2 & \cdots & x_m \\ & & & \end{array}\right] = \left[\begin{array}{c|c|c|c} & & & \\ x_1 & x_2 & \cdots & x_m \\ & & & \end{array}\right] \begin{bmatrix} \lambda_1 & & & \\ & \lambda_2 & & \\ & & \ddots & \\ & & & \lambda_m \end{bmatrix}.$$

Nếu x_j là cột thứ j của X và λ_j là phần thứ j trên đường chéo của Λ thì $Ax_j = \lambda_j x_j$. Do đó cột thứ j của X là một vector riêng của A và phần tử thứ j của Λ là trị riêng tương ứng.

Phân tích trị riêng biểu diễn một sự thay đổi cơ sở thành "các tọa độ vector riêng". Nếu $Ax = b$ và $A = X\Lambda X^{-1}$, ta có

$$(X^{-1}b) = \Lambda(X^{-1}x). \tag{5.1.4}$$

Do đó, để tính Ax, ta có thể mở rộng x trong cơ sở của các cột của X, áp dụng Λ và làm sáng tỏa kết quả như là một vector của các hệ số của một tổ hợp tuyến tính của các cột của X.

5.1.3 Số bội hình học

Như phát biểu ở trên, tập hợp các vector riêng tương ứng với một trị riêng đơn, cùng với vector 0, tạo thành một không gian con của $\mathbb{C}^m$ được biết như là một *không gian riêng*. Nếu λ là trị riêng của A, ta ký hiệu không gian riêng tương ứng bởi E_λ. Một không gian riêng E_λ là một ví dụ của một *không gian con bất biến* của A; đó là, $AE_\lambda \subseteq E_\lambda$.

Số chiều của E_λ là số lớn nhất của các vector riêng độc lập tuyến tính mà chúng có thể được tìm thấy, với tất cả trị riêng giống nhau λ. Con số này được biết như là *số bội hình học* của λ. Số bội hình học cũng có thể được miêu tả như số chiều của không gian đầy đủ của $A - \lambda I$ vì không gian đầy đủ đó là E_λ.

5.1.4 Đa thức đặc trưng

Đa thức đặc trưng của $A \in \mathbb{C}^{m\times m}$ được ký hiệu bởi p_A hoặc đơn giản là p, là đa thức bậc m xác định bởi

$$p_A(z) = det(zI - A). \tag{5.1.5}$$

Do sự sắp xếp của dấu trừ, p là *đa thức đơn khởi*: hệ số bậc m của nó là 1.

Định lý 5.1.1 *λ là một trị riêng của A nếu và chỉ nếu $p_A(\lambda) = 0$.*

Chứng minh Điều này theo sau từ định nghĩa của một trị riêng:

$$\begin{aligned} \lambda \text{ là một trị riêng} &\iff \text{tồn tại một vector x khác 0 sao cho } \lambda x - Ax = 0 \\ &\iff \lambda I - A \text{ là suy biến} \\ &\iff det(\lambda I - A) = 0. \end{aligned}$$

Định lý 5.1.1 có một kết quả quan trọng. *Ngay cả khi môt ma trận là thực, một vài trị riêng của nó có thể là phức.* Theo vật lý, điều này được liên quan tới hiện tượng mà các hệ thống động thực tế có thể có nhiều hành động mà sự giao động tốt như tăng hoặc giảm. Theo thuật toán, nó nghĩa là ngay cả khi nếu đầu vào của bài toán trị riêng ma trận là thực, đầu ra có thể phải là phức.

5.1.5 Số bội đại số

Do định lý cơ bản của đại số, ta có thể viết p_A dưới dạng

$$p_A(z) = (z - \lambda_1)(z - \lambda_2)\dots(z - \lambda_m) \tag{5.1.6}$$

với $\lambda_i \in \mathbb{C}$ bất kì. Do Định lý 5.1.1, mỗi λ_j là một trị riêng của A, và tất cả các trị riêng của A xuất hiện đâu đó trong danh sách này. Tổng quát, một trị riêng có thể xuất hiện hơn một lần. Ta xác định *số bội đại số* của một trị riêng λ của A là số bội của nó như một nghiệm của p_A. Một trị riêng là *đơn* nếu số bội đại số của nó là 1.

Đa thức đặc trưng cho chúng ta một cách đơn giản để đếm các trị riêng của một ma trận.

Định lý 5.1.2 *Nếu $A \in \mathbb{C}^{m\times m}$ thì A có m trị riêng, kể cả số bội đại số. Đặc biệt, nếu các nghiệm của p_A là đơn thì A có m trị riêng phân biệt.*

Chú ý rằng, mọi ma trận có ít nhất một trị riêng.

Số bội đại số của một trị riêng thường là nhỏ nhất như số bội hình học của nó. Để chứng minh điều này, ta cần biết một vài việc về các biến đổi đồng dạng.

5.1.6 Các biến đổi tương đương

Nếu $X \in \mathbb{C}^{m\times m}$ là không suy biến thì ánh xạ $A \mapsto X^{-1}AX$ được gọi là một *biến đổi tương đương* của A. Ta nói rằng hai ma trận A và B là *tương đương* nếu tồn tại một biến đổi tương đương giữa chúng, nghĩa là, nếu tồn tại một ma trận không suy biến $X \in \mathbb{C}^{m\times m}$ sao cho $B = X^{-1}AX$. Như được miêu tả ở trên trường hợp đặc biệt của chéo hóa (5.1.3), biến đổi tương đương bất kì là một phép toán thay đổi cơ sở.

Định lý 5.1.3 *Nếu X là không suy biến thì A và $X^{-1}AX$ có cùng đa thức đặc trưng, cùng các trị riêng, và có cùng các số bội đại số và hình học.*

Chứng minh Chứng minh rằng sự phù hợp của các đa thức đặc trưng là một sự tính toán không phức tạp:

$$\begin{aligned} p_{X^{-1}AX}(z) &= det(zI - X^{-1}AX) = det(X^{-1}(zI - A)X) \\ &= det(X^{-1})det(zI - A)det(X) = det(zI - A) = p_A(z). \end{aligned}$$

Từ sự phù hợp của các đa thức đặc trưng, sự phù hợp của các trị riêng và các số bội đại số theo sau. Cuối cùng, để chứng minh các số bội hình học phù hợp, ta có thể kiểm tra rằng nếu E_λ là không gian riêng của A thì $X^{-1}E_\lambda$ là một không gian riêng của $X^{-1}AX$, và ngược lại.

Định lý 5.1.4 *Số bội đại số của một trị riêng λ là ít nhất như số bội hình học của nó.*

Chứng minh Cho n là số bội hình học của λ cho ma trận A. Tạo thành một ma trận $\widehat{C}$ cấp $m \times n$ mà n cột của nó tạo thành một cơ sở trực giao của không gian riêng $\{x : Ax = \lambda x\}$. Khi đó, việc mở rộng $\widehat{V}$ thành một ma trận unita vuông V, ta thu được V^*AV trong dạng

$$B = V^*AV = \begin{bmatrix} \lambda I & C \\ 0 & D, \end{bmatrix} \tag{5.1.7}$$

với I là ma trận đơn vị $n \times n$, C là ma trận $n \times (m - n)$, và D là ma trận $(m - n) \times (m - n)$. Do định nghĩa của định thức, $det(zI - B) = det(zI - \lambda I)det(zI - D) = (z - \lambda)^n det(zI - D)$. Do đó số bội đại số của λ như là một trị riêng của B ít nhất là n. Vì các biến đổi tương đương bảo toàn các số bội, kết quả tương tự đúng cho A.

5.1.7 Các ma trận và trị riêng khiếm khuyết

Mặc dù một ma trận có số bội hình học và đại số là bằng nhau (cụ thể, tất cả bằng 1), điều này không có nghĩa là đúng cho mọi ma trận.

Ví dụ 5.1.1. Xét các ma trận

$$A = \begin{bmatrix} 2 & & \\ & 2 & \\ & & 2 \end{bmatrix},\ B = \begin{bmatrix} 2 & 1 & \\ & 2 & 1 \\ & & 2 \end{bmatrix}.$$

Cả A và B có đa thức đặc trưng $(z-2)^3$, nên có một trị riêng đơn là $\lambda = 2$ số bội đại số là 3. Trong trường hợp của A, ta có thể chọn 3 vector riêng độc lập, ví dụ như e_1, e_2 và e_3, nên số bội hình học cũng là 3. Cho B, mặc khác, ta có thể chỉ tìm một vector độc lập đơn (một bội vô hướng của e_1), nên số bội hình học của trị riêng này chỉ là 1.

Một trị riêng mà số bội đại số vượt quá số bội hình học của nó là một *trị riêng khiếm khuyết*. Một ma trận mà nó có nhiều hơn một trị riêng khiếm khuyết là một *ma trận khiếm khuyết*.

Ma trận đường chéo bất kì là không khiếm khuyết. Cho một ma trận như vậy, cả số bội hình học và đại số của một trị riêng λ là bằng số lần xuất hiện của nó trên đường chéo.

5.1.8 Sự chéo hóa

Lớp các ma trận không khiếm khuyết rõ ràng là lớp các ma trận có một phân tích trị riêng (5.1.3).

Định lý 5.1.5 *Một ma trận $m \times m$ A là không khiếm khuyết nếu và chỉ nếu nó có một phân tích trị riêng $A = X\Lambda X^{-1}$.*

Chứng minh ($\Longleftarrow$) Cho một phân tích trị riêng $A = X\Lambda X^{-1}$, ta biết do Định lý 5.1.3 mà Λ là tương đương với A, với cùng các trị riêng và cùng số bội. Vì Λ là một ma trận đường chéo, nó không là ma trận khiếm khuyết, và do đó cũng đúng cho A.
($\Longrightarrow$) Một ma trận không khiếm khuyết phải có m vector riêng độc lập tuyến tính, bởi vì các vector riêng với các trị riêng khác nhau phải độc lập tuyến tính, và mỗi trị riêng có thể đóng góp các vector riêng độc lập tuyến tính cũng như số bội của nó. Nếu m vector riêng độc lập này được tạo thành các cột của một ma trận X thì X là không suy biến và ta có $A = X\Lambda X^{-1}$.

5.1.9 Định thức và vết

Vết của $A \subset \mathbb{C}^{m\times m}$ là tổng của các phần tử trên đường chéo của nó: $tr(A) = \sum_{j=1}^{m} a_{jj}$. Cả hai vết và định thức có quan hệ với các trị riêng.

Định lý 5.1.6 *Định thức $det(A)$ và vết $tr(A)$ tương ứng là bằng với tích và tổng các trị riêng của A, được đếm với số bội đại số:*

$$det(A) = \prod_{j=1}^{m} \lambda_j,\ tr(A) = \sum_{j=1}^{m} \lambda_j. \tag{5.1.8}$$

Chứng minh Từ (5.1.5) và (5.1.6), ta tính

$$det(A) = (-1)^m det(-A) = (-1)^m p_A(0) = \prod_{j=1}^{m} \lambda_j.$$

Điều này ước lượng công thức đầu tiên. Cho công thức thứ 2, từ (5.1.5), hệ số của số hạng z^{m-1} của p_A là số âm của tổng các phần tử trên đường chéo của A, hoặc $-tr(A)$. Mặc khác, từ (5.1.6), hệ số này cũng bằng với $-\sum_{j=1}^{m} \lambda_j$. Do đó $tr(A) = \sum_{j=1}^{m} \lambda_j$.

5.1.10 Chéo hóa Unita

Thỉnh thoảng nó xảy ra mà không chỉ ma trận A cấp $m \times m$ có m vector riêng độc lập tuyến tính, nhưng các vector này có được chọn để là trực giao. Trong trường hợp như vậy, A là *chéo hóa Unita*, nghĩa là, tồn tại một ma trận Unita Q sao cho

$$A = Q\Lambda Q^*. \tag{5.1.9}$$

Phân tích này là phân tích trị riêng và phân tích giá trị suy biến, trừ vấn đề các dấu (phức có thể) của các phần tử của A.

Ta đã thấy sẵn một lớp các ma trận mà chúng là chéo hóa Unita: các ma trận hermit. Kết quả theo sau từ Định lý 5.1.9 bên dưới.

Định lý 5.1.7 *Một ma trận hermit là chéo hóa Unita, và các trị riêng của nó là thực.*

Các ma trận hermit không chỉ là chéo hóa Unita. Các ví dụ khác bao gồm các ma trận hermit lệch, các ma trận unita, các ma trận luân hoàn, và bất kì trong số chúng cộng thêm một bội của đơn vị. Tổng quát, lớp các ma trận mà chúng là chéo hóa Unita có một đặc trưng tao nhã. Do định nghĩa, ta nói rằng một ma trận A là *chuẩn tắc* nếu $A^*A = AA^*$. Kết quả theo sau là phổ biến.

Định lý 5.1.8 *Một ma trận là chéo hóa Unita nếu và chỉ nếu nó là chuẩn tắc.*

5.1.11 Phân tích Schur

Phân tích Schur của một ma trận A là một phân tích

$$A = QTQ^*, \tag{5.1.10}$$

với A là Unita và T là ma trận tam giác trên. Chú ý, A và T là tương đương, các trị riêng của A tất nhiên xuất hiện trong đường chéo của T.

Định lý 5.1.9 *Mọi ma trận vuông A có một phân tích Schur.*

Chứng minh Ta xử lý bằng quy nạp theo số chiều m của A. Trường hợp $m = 1$ là tầm thường, nên giả sử $m \geqslant 2$. Cho x là vector riêng bất kì của A, tương ứng với trị riêng λ. Chuẩn hóa x và cho nó là cột đầu tiên của ma trận Unita U. Khi đó, giống như trong (5.1.7), U^*AU có dạng

$$U^*AU = \begin{bmatrix} \lambda & B \\ 0 & C \end{bmatrix}.$$

Do giả thuyết quy nạp, tồn tại một phân tích Schur VTV^* của C. Bây giờ ta viết

$$Q = U \begin{bmatrix} 1 & 0 \\ 0 & V \end{bmatrix}.$$

Đây là một ma trận Unita, và ta có

$$Q^*AQ = \begin{bmatrix} \lambda & BV \\ 0 & T \end{bmatrix}.$$

Đây là phân tích Schur mà ta tìm kiếm.

5.2 Tổng quan của các thuật toán trị riêng

5.2.1 Sự thiếu sót của các thuật toán hiển nhiên

Mặc dù các trị riêng và vector riêng có các định nghĩa đơn giản và các đặc trưng tao nhã, nhưng nhiều cách tốt nhất để tính chúng là không rõ ràng.

Có thể phương pháp đầu tiên mà ta có thể nghĩ sẽ là để tính các hệ số của đa thức đặc trưng và sử dụng bộ tìm kiếm nghiệm để tách ra các nghiệm của nó. Không may mắn, như được đề cập trong mục trước, kỹ thuật này là một cách tệ, bởi vì việc tìm nghiệm đa thức là một bài toán điều kiện xấu tổng quát, ngay cả khi bài toán trị riêng cơ bản là điều kiện tốt. (Thật vậy, việc tìm nghiệm đa thức không có nghĩa là chủ đề chính trong tính toán khoa học - chính xác bởi vì nó là quá hiếm cách tốt nhất để giải các bài toán được áp dụng.)

Một ý tưởng khác sẽ tận dụng chuỗi

$$\frac{x}{\|x\|}, \frac{Ax}{\|Ax\|}, \frac{A^2x}{\|A^2x\|}, \frac{A^3x}{\|A^3x\|}, \dots$$

hội tụ tới một vector riêng tương ứng với trị riêng lớn nhất của A trong giá trị tuyệt đối. Phương pháp này cho việc tìm các vector riêng được gọi là *bước lặp lũy thừa*. Mặc dù bước lặp lũy thừa là phổ biến, nhưng nó không có nghĩa là một công cụ hiệu quả cho sử dụng chung. Ngoại trừ các ma trận đặc biệt, nó rất chậm.

Các thuật toán trị riêng mục đích chung tốt nhất được dựa vào một nguyên lý khác: tính toán một phân tích trị riêng cho thấy của A, trong đó các trị riêng xuất hiện như là các phần tử của một trong các thừa số. Ta thấy 3 phân tích trị riêng cho thấy trong mục trước: chéo hóa, chéo hóa Unita, và tam giác hóa Unita (phân tích Schur). Trong thực hành, các trị riêng thường được tính bằng việc xây dựng một trong những phân tích này. Khái niệm, cái phải được làm để đạt được điều này là để áp dụng một chuỗi các biến đổi tới A để đưa ra các số 0 ở những nơi cần thiết, giống như trong các thuật toán ta đã xét trong các mục trước của sách này. Do đó ta thấy rằng việc tìm các trị riêng kết thúc hơn là việc giải hệ thống các phương trình hoặc các bài toán bình phương nhỏ nhất.

5.2.2 Sự khác nhau cơ bản

Để thấy là khó khăn, chú ý rằng ngay khi các bài toán trị riêng có thể được giảm thành các bài toán tìm nghiệm đa thức, ngược lại, bài toán tìm nghiệm đa thức bất kì có thể được phát biểu như một bài toán trị riêng. Giả sử ta có đa thức đơn khởi

$$p(z) = z^m + a_{m-1}z^{m-1} + \ldots + a_1 z + a_0. \tag{5.2.1}$$

Bằng việc mở rộng trong các định thức, không khó để kiểm tra rằng $p(z)$ bằng $(-1)^m$ nhân định thức của ma trận $m \times m$

$$\begin{bmatrix} -z & & & & & -a_0 \\ 1 & -z & & & & -a_1 \\ & 1 & -z & & & -a_2 \\ & & 1 & \ddots & & \vdots \\ & & & \ddots & -z & -a_{m-2} \\ & & & & 1 & (-z - a_{m-1}) \end{bmatrix}. \tag{5.2.2}$$

Điều này có nghĩa là các nghiệm của p là bằng với các trị riêng của ma trận

$$\begin{bmatrix} 0 & & & & & -a_0 \\ 1 & 0 & & & & -a_1 \\ & 1 & 0 & & & -a_2 \\ & & 1 & \ddots & & \vdots \\ & & & \ddots & 0 & -a_{m-2} \\ & & & & 1 & -a_{m-1} \end{bmatrix}. \tag{5.2.3}$$

(Ta cũng có thể có (5.2.3) một cách trực tiếp, không qua (5.2.2), bằng việc kí hiệu nếu z là một nghiêm của p, thì nó theo sau từ (5.2.1) mà $(1, z, z^2, \ldots, z^{m-1}$ là một vector riêng trái của A với trị riêng z.) A được gọi là một *ma trận đồng hành* tương ứng với p.

Bây giờ sự khó khăn xuất hiện. Nó phổ biến mà không công thức nào tồn tại cho việc biểu diễn các nghiệm của một đa thức bất kì mà các hệ số của nó được cho. Kết quả không thể làm được này là một trong những sự đạt được hoàn thiện của một vật thể làm việc toán học được thực hiện bởi Abel, Galois, và những người khác ở thế kỉ 19. Abel chứng minh trong năm 1824 rằng không có sự tương tự của công thức bậc hai có thể tồn tại cho các đa thức bậc lớn hơn hoặc bằng 5.

Định lý 5.2.1 *Cho $m \geqslant 5$ bất kì, tồn tại một đa thức $p(z)$ bậc m với các hệ số hữu tỉ mà nó có một nghiệm thực $p(r) = 0$ với tính chất r không thể được viết bằng việc sử dụng sự biểu diễn bất kì bao gồm các số hữu tỉ, phép cộng, phép trừ, phép nhân, phép chia và k nghiệm.*

Định lý này kéo theo rằng ngay cả khi ta có thể làm việc trong số học chính xác, không có một chương trình máy tính nào sẽ đưa ra các nghiệm chính xác của một đa thức bất kì trong một số hữu hạn bước. Kết luận tương tự áp dụng cho bài toán tổng quát hơn của việc tính toán các trị riêng của các ma trận.

Điều này không có nghĩa rằng ta không thể viết một chương trình giải trị riêng tốt. Tuy nhiên, nó có nghĩa là một chương trình giải như vậy không thể được dựa vào cùng loại của các kỹ thuật mà ta đã sử dụng cho đến nay cho việc giải hệ thống tuyến tính. Các phương pháp giống như các phản xạ Householder và khử Gauss sẽ giải các hệ thống tuyến tính của các phương trình một cách chính xác trong một số hữu hạn bước nếu chúng có thể được thực thi trong số học chính xác. Ngược lại,

Một chương trình giải trị riêng bất kì phải là lặp.

Mục tiêu của một chương trình giải trị riêng là để đưa ra *các chuỗi số mà chúng hội tụ nhanh về hướng các trị riêng*. Trong các tính toán trị riêng theo phương diện này là nhiều biểu diễn của việc tính toán khoa học hơn là các lời giải của các hệ thống tuyến tính của các phương trình.

Sự cần thiết để lặp có thể dường như làm nản lòng tại bước đầu tiên, nhưng các thuật toán có thể dùng được trong phạm vi này hội tụ nhanh. Trong hầu hết các trường hợp nó là có thể thực hiện được để tính các chuỗi số mà các số bội hai hoặc bội ba của các chữ số của sự đúng đắn tại mỗi bước. Do đó, mặc dù việc tính toán các trị riêng là một bài toán "không thể giải quyết được" nói chung, trong thực hành nó suy ra từ lời giải của các hệ thống tuyến tính chỉ bằng một thừa số hằng nhỏ, gần 1 hơn 10. Nói về mặt lý thuyết, sự phụ thuộc của đếm phép toán vào $\epsilon_{machine}$ gồm các số hạng yếu như $\log(|\log(\epsilon_{machine})|)$.

5.2.3 Phân tích Schur và sự chéo hóa

Hầu hết các thuật toán trị riêng sử dụng ngày nay xử lý bằng việc tính phân tích Schur. Ta tính một phân tích Schur $A = QTQ^*$ bằng việc biến đổi A bởi một chuỗi các biến đổi tương

tự Unita cơ bản $X \to Q_j^* X Q_j$, để tích

$$\underbrace{Q_j^* \dots Q_2^* Q_1^*}_{Q^*} A \underbrace{Q_1 Q_2 \dots Q_j}_{Q} \tag{5.2.4}$$

hội tụ về một ma trận tam giác trên T khi $j \to \infty$.

Nếu A là một ma trận thực nhưng không đối xứng, thì tổng quát nó có thể có các trị riêng phức trong các cặp liên hợp với nhau, mà trong đó trường hợp dạng Schur của nó sẽ là phức. Do đó một thuật toán tính phân tích Schur sẽ phải có khả năng phát sinh ra các đầu ra phức từ các đầu vào thực. Điều này có thể chắc chắn được làm; sau tất cả, tìm các nghiệm cho các đa thức với các hệ số thực có tính chất tương tự nhau. Ngoài ra, nó có thể để thực thi tính toán phần tử trong số học thực nếu nó tính toán cái được biết như là một *phân tích Schur thực*. Ở đây, T được cho phép để có các khối 2×2 dọc theo đường chéo, một cho mỗi cặp liên hợp phức của các trị riêng. Tùy chọn này là quan trọng trong thực hành, và được bao gồm trong tất cả các thư viện phần mềm, nhưng ta sẽ không cho chi tiết ở đây.

Mặt khác, giả sử A là hermit. Khi đó $Q_j^* \dots Q_2^* Q_1^* A Q_1 Q_2 \dots Q_j$ cũng là hermit, và do đó giới hạn của chuỗi hội tụ là cả 2 dạng tam giác và hermit, do đó là đường chéo. Điều này kéo theo các thuật toán giống nhau tính một tam giác hóa Unita của môt ma trận tổng quát cũng tính toán một chéo hóa Unita của một ma trận hermit. Trong thực hành, điều này về cơ bản cho thấy trường hợp hermit điển hình được xử lý, mặc dù các bổ sung khác nhau được đưa ra để tận dụng cấu trúc hermit tại mỗi bước.

5.2.4 Hai giai đoạn của sự tính toán trị riêng

Cho dù A có là hermit hay không, chuỗi (5.2.4) thường được chia thành 2 giai đoạn. Trong giai đoạn đầu tiên, một phương pháp trực tiếp được áp dụng để đưa ra một ma trận *Hessenberg trên* H, một ma trận với các số 0 ở bên dưới đường chéo phụ đầu tiên. Trong giai đoạn thứ hai, một phép lặp được áp dụng để sinh ra một chuỗi vô hạn hình thức của các ma trận Hessenberg mà chúng hội tụ về một dạng tam giác. Dưới dạng biểu đồ, quá trình trông giống điều này:

$$\underset{A \neq A^*}{\begin{bmatrix} \times & \times & \times & \times & \times \\ \times & \times & \times & \times & \times \\ \times & \times & \times & \times & \times \\ \times & \times & \times & \times & \times \\ \times & \times & \times & \times & \times \end{bmatrix}} \xrightarrow{\text{Giai đoạn 1}} \underset{H}{\begin{bmatrix} \times & \times & \times & \times & \times \\ \times & \times & \times & \times & \times \\ & \times & \times & \times & \times \\ & & \times & \times & \times \\ & & & \times & \times \end{bmatrix}} \xrightarrow{\text{Giai đoạn 2}} \underset{T}{\begin{bmatrix} \times & \times & \times & \times & \times \\ & \times & \times & \times & \times \\ & & \times & \times & \times \\ & & & \times & \times \\ & & & & \times \end{bmatrix}}.$$

Giai đoạn thứ nhất, giảm trực tiếp, yêu cầu $O(m^3)$ phép toán dấu chấm động. Giai đoạn thứ hai, giai đoạn lặp không bao giờ kết thúc về nguyên tắc, và nếu di chuyển để chạy mãi mãi sẽ cần một số vô hạn các phép toán dấu chấm động. Tuy nhiên, trong thực hành, sự hội tụ tới độ chính xác của máy được đạt được trong $O(m)$ bước lặp. Mỗi bước lặp yêu cầu $O(m^2)$ phép toán dấu chấm động, và do đó tổng số việc làm cần thiết là $O(m^3)$ phép toán dấu chấm động. Các số liệu này giải thích sự quan trọng của Giai đoạn 1. Không có bước sơ bộ đó, mỗi bước lặp của Giai đoạn 2 sẽ bao gồm một ma trận đầy đủ, yêu cầu $O(m^3)$ việc làm, và điều này sẽ đem đến tổng số thành $O(m^4)$ - hoặc cao hơn, vì sự hội tụ cũng có thể đôi khi yêu cầu nhiều hơn $O(m)$ bước lặp.

Nếu A là hermit, phương pháp xấp xỉ 2 giai đoạn trở nên nhanh hơn. Ma trận lặp bây giờ là một ma trận Hessenberg hermit, ma trận *ba đường chéo*. Kết quả cuối cùng là một ma trận

tam giác hermit, ma trận đường chéo, như được đề cập ở trên. Dưới dạng biểu đồ:

$$\underset{A=A^*}{\begin{bmatrix}\times & \times & \times & \times & \times\\ \times & \times & \times & \times & \times\\ \times & \times & \times & \times & \times\\ \times & \times & \times & \times & \times\\ \times & \times & \times & \times & \times\end{bmatrix}} \xrightarrow{\text{Giai đoạn 1}} \underset{T}{\begin{bmatrix}\times & \times & & & \\ \times & \times & \times & & \\ & \times & \times & \times & \\ & & \times & \times & \times\\ & & & \times & \times\end{bmatrix}} \xrightarrow{\text{Giai đoạn 2}} \underset{D}{\begin{bmatrix}\times & & & & \\ & \times & & & \\ & & \times & & \\ & & & \times & \\ & & & & \times\end{bmatrix}}.$$

Trong trường hợp hermit này ta sẽ thấy rằng nếu chỉ các trị riêng được yêu cầu (không phải các vector riêng), khi đó mỗi bước của Giai đoạn 2 có thể được thực hiện chỉ với $O(m)$ phép toán dấu chấm động, làm cho tổng số việc làm ước lượng cho Giai đoạn 2 là $O(m^2)$ phép toán dấu chấm động. Do đó, cho các bài toán trị riêng hermit, ta làm trong tình huống nghịch lý mà phần "vô hạn" của thuật toán là không chỉ nhanh như phần "hữu hạn" trong thực hành, nhưng thứ tự của độ lớn nhanh hơn.

5.3 Sự giảm thành dạng Hessenberg hoặc dạng đường chéo

Bây giờ ta sẽ miêu tả giai đoạn đầu tiên của 2 giai đoạn tính toán được phác thảo trong mục trước: giảm một ma trận đầy đủ thành dạng Hessenberg bằng một chuỗi các biến đổi tương tự Unita. Nếu ma trận ban đầu là hermit, thì ma trận kết quả là 3 đường chéo.

5.3.1 Ý tưởng xấu

Để tính phân tích Schur $A = QTQ^*$, ta muốn áp dụng các biến đổi tương tự Unita tới A để đưa các số 0 vào bên dưới đường chéo. Ý tưởng đầu tiên tự nhiên có thể là để cố gắng tam giác hóa trực tiếp bằng việc sử dụng các phản xạ Householder để đưa các số 0 này, liên tiếp.

Phản xạ Householder đầu tiên Q_1^* được nhân vào bên trái của A sẽ đưa ra các số 0 bên dưới đường chéo trong cột đầu tiên của A. Trong quá trình này, nó sẽ thay đổi tất cả các dòng của A. Trong điều này và các biểu đồ theo sau, như thường lệ, các phần tử bị thay đổi tại mỗi bước được viết in đậm:

$$\underset{A}{\begin{bmatrix}\times & \times & \times & \times & \times\\ \times & \times & \times & \times & \times\\ \times & \times & \times & \times & \times\\ \times & \times & \times & \times & \times\\ \times & \times & \times & \times & \times\end{bmatrix}} \xrightarrow{Q_1^*\cdot} \underset{Q_1^*A}{\begin{bmatrix}\times & \times & \times & \times & \times\\ \mathbf{0} & \times & \times & \times & \times\\ \mathbf{0} & \times & \times & \times & \times\\ \mathbf{0} & \times & \times & \times & \times\\ \mathbf{0} & \times & \times & \times & \times\end{bmatrix}}.$$

Không may mắn, để hoàn thành biến đổi tương tự, ta cũng phải nhân Q_1 vào vế phải của A:

$$\underset{Q_1^*A}{\begin{bmatrix}\times & \times & \times & \times & \times\\ & \times & \times & \times & \times\\ & \times & \times & \times & \times\\ & \times & \times & \times & \times\\ & \times & \times & \times & \times\end{bmatrix}} \xrightarrow{\cdot Q_1} \underset{Q_1^*AQ_1}{\begin{bmatrix}\times & \times & \times & \times & \times\\ \times & \times & \times & \times & \times\\ \times & \times & \times & \times & \times\\ \times & \times & \times & \times & \times\\ \times & \times & \times & \times & \times\end{bmatrix}}.$$

Điều này có hiệu quả của việc thay thế mỗi cột của ma trận bằng một tổ hợp tuyến tính của tất cả các cột. Kết quả là các số 0 mà chúng được đưa vào được triệt tiêu; không tốt hơn khi ta tắt đầu.

Dĩ nhiên, ta biết rằng ý tưởng này phải thất bại, bởi vì "sự khó khăn cơ bản" được miêu tả trong mục trước. Không quá trình hữu hạn có thể phát hiện các trị riêng của A một cách chính xác.

Chiến lược quá đơn giản này xuất hiện không có hiệu quả như ta đã thảo luận , tiêu biểu, việc giảm kích thước của các phần tử bên dưới đường chéo, ngay cả khi nó không làm chúng thành số 0. Ta sẽ trở lại điều này "ý tưởng tệ" khi ta thảo luận thuật toán QR.

5.3.2 Ý tưởng tốt

Chiến lược thích hợp cho việc đưa ra các số 0 trong Giai đoạn 1 là ít tham vọng và hoạt động trong hơn một vài phần tử của ma trận. Ta chỉ sẽ đoạt phạm vi mà ta chắc chắn ta có thể bảo vệ.

Tại bước đầu tiên, ta chọn một phản xạ Householder Q_1^* mà nó cho phép dòng đầu tiên không thay đổi. Khi nó được nhân vào vế trái của A, nó hình thành các tổ hợp tuyến tính chỉ của các dòng $2, \ldots, m$ đưa ra các số 0 thành các dòng $3, \ldots, m$ của cột đầu tiên. Do đó, khi Q_1 được nhân vào vế phải của Q_1^*A, nó cho phép cột đầu tiên không thay đổi. Nó hình thành các tổ hợp tuyến tính của các cột $2, \ldots, m$ và không thay đổi các số 0 đã được đưa ra:

$$\underset{A}{\begin{bmatrix} \times & \times & \times & \times & \times \\ \times & \times & \times & \times & \times \\ \times & \times & \times & \times & \times \\ \times & \times & \times & \times & \times \\ \times & \times & \times & \times & \times \end{bmatrix}} \xrightarrow{Q_1^*\cdot} \underset{Q_1^*A}{\begin{bmatrix} \times & \times & \times & \times & \times \\ \times & \times & \times & \times & \times \\ 0 & \times & \times & \times & \times \\ 0 & \times & \times & \times & \times \\ 0 & \times & \times & \times & \times \end{bmatrix}} \xrightarrow{\cdot Q_1} \underset{Q_1^*AQ_1}{\begin{bmatrix} \times & \times & \times & \times & \times \\ \times & \times & \times & \times & \times \\ & \times & \times & \times & \times \\ & \times & \times & \times & \times \\ & \times & \times & \times & \times \end{bmatrix}}.$$

Ý tưởng này được lặp lại để đưa ra các số 0 thành các cột phân dãy. Ví dụ, phản xạ Householder thứ hai, Q_2, cho phép dòng thứ nhất và thứ hai không thay đổi và các cột thứ nhất và thứ hai không thay đổi:

$$\underset{Q_1^*AQ_1}{\begin{bmatrix} \times & \times & \times & \times & \times \\ \times & \times & \times & \times & \times \\ & \times & \times & \times & \times \\ & \times & \times & \times & \times \\ & \times & \times & \times & \times \end{bmatrix}} \xrightarrow{Q_2^*\cdot} \underset{Q_2^*Q_1^*AQ_1}{\begin{bmatrix} \times & \times & \times & \times & \times \\ \times & \times & \times & \times & \times \\ & \times & \times & \times & \times \\ & 0 & \times & \times & \times \\ & 0 & \times & \times & \times \end{bmatrix}} \xrightarrow{\cdot Q_2} \underset{Q_2^*Q_1^*AQ_1Q_2}{\begin{bmatrix} \times & \times & \times & \times & \times \\ \times & \times & \times & \times & \times \\ & \times & \times & \times & \times \\ & & \times & \times & \times \\ & & \times & \times & \times \end{bmatrix}}.$$

Sau khi lặp lại quá trình này $m-2$ lần, ta có một tích dạng Hessenberg, như được miêu tả:

$$\begin{bmatrix} \times & \times & \times & \times & \times \\ \times & \times & \times & \times & \times \\ & \times & \times & \times & \times \\ & & \times & \times & \times \\ & & & \times & \times \end{bmatrix}$$

$$\underbrace{Q_{m-2}^* \ldots Q_2^* Q_1^*}_{Q^*} A \underbrace{Q_1 Q_2 \ldots Q_{m-2}}_{Q} = H.$$

Thuật toán 5.1 Giảm Householder thành dạng Hessenberg

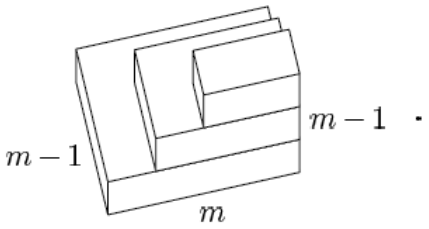
for $k = 1$ to $m-2$ **do**
  $x = A_{k+1:m,k}$
  $v_k = sign(x_1)\|x\|_2 e_1 + x$
  $v_k = v_k/\|v_k\|_2$
  $A_{k+1:m,k:m} = A_{k+1:m,k:m} - 2v_k(v_k^* A_{k+1:m,k:m})$
  $A_{1:m,k|1:m} = A_{1:m,k+1:m} - 2(A_{1:m,k+1:m}v_k)v_k^*$
end for

Thuật toán được đề ra bên dưới, sao sánh với Thuật toán 2.3.
Ngay như trong Thuật toán 2.3, ở đây ma trận $Q = \prod_{k=1}^{m-2} Q_k$ không bao giờ được hình thành rõ ràng. Các vector phản xạ v_k được lưu trữ thay thế, và có thể được sử dụng để nhân với Q hoặc xây dựng lại Q nếu cần thiết. Cụ thể thấy trong mục 2.5.

5.3.3 Đếm số phép toán

Số phép toán được yêu cầu bởi Thuật toán 5.1 có thể được đếm với cùng lý do hình học mà ta đã sử dụng trước đây. Các phép toán unita yêu cầu 4 phép toán dấu chấm động cho mỗi phần tử được thực hiện ở trên.

Việc làm này được chi phối bởi 2 cập nhật của các ma trận con của A. Vòng lập đầu tiên áp dụng một phản xạ Householder trong vế trái của ma trận. Phản xạ thứ k như vậy thực hiện trong $m-k$ dòng cuối. Vì tại thời điểm này phản xạ được áp dụng, các dòng này có các số 0 trong $k-1$ cột đầu tiên, số học phải được thực thi chỉ trong $m-k+1$ phần tử cuối của mỗi dòng. Hình ảnh như sau:
Khi $m \to \infty$, thể tích hội tụ về $\frac{1}{3}m^3$. Tại 4 phép toán dấu chấm động trên phần tử, tổng số

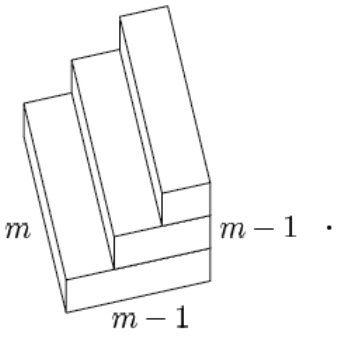


việc làm trong vòng lặp này là $\sim \frac{4}{3}m^3$ phép toán dấu chấm động.

Vòng lặp bên trong thứ hai áp dụng một phản xạ Householder vào vế phải của ma trận. Tại bước thứ k, phản xạ thực thi bằng việc hình thành các tổ hợp tuyến tính của $m-k$ cột sau cùng. Vòng lặp này bao gồm nhiều việc làm hơn vòng lặp thứ nhất bởi vì không có số 0 mà chúng có thể được lờ đi. Số học phải được thực thi trong tất cả m phần tử của mỗi cột được thực thi ở trên, tổng số $m(m-k)$ phần tử cho một giá trị đơn của k. Hình ảnh trông giống điều này:
Thể tích hội tụ về $\frac{1}{2}m^3$ khi $m \to \infty$, nên, tại 4 phép toán dấu chấm động trên phần tử, vòng lặp thứ hai yêu cầu $\sim 2m^3$ phép toán dấu chấm động.

Cùng với tất cả, tổng số việc làm cho sự giảm unita của một ma trận $m \times m$ thành dạng Hessenberg là:

$$\text{Sự giảm Hessenberg: } \sim \frac{10}{3}m^3 \text{ phép toán dấu chấm động.} \tag{5.3.1}$$



5.3.4 Trường hợp Hermit: Sự giảm thành dạng 3 đường chéo

Nếu A là hermit thì thuật toán vừa được miêu tả sẽ giảm A thành dạng 3 đường chéo (ít nhất, trong sự vắng mặt của các sai số làm tròn). Điều này là dễ thấy: vì A là hermit, Q^*AQ cũng là hermit, và ma trận Hessenberg hermit bất kì là 3 đường chéo.

Vì các số 0 được đưa ra trong các dòng tốt như các cột, số học thêm vào có thể được tránh bằng việc lờ đi các số 0 thêm vào này. Với sự tối ưu này, việc áp dụng một phản xạ Householder vào vế phải là xấu như việc áp dụng phản xạ vào vế phải, và chi phí tổng cộng của việc áp dụng các phản xạ vào vế phải được giảm từ $2m^3$ xuống thành $\frac{4}{3}m^3$ phép toán dấu chấm động. Ta có hai hình chóp để lấy tổng thay vì một hình chớp và một lăng trụ, và tổng số phép toán số học được giảm xuống thành $\frac{8}{3}m^3$ phép toán dấu chấm động.

Tuy nhiên, việc lưu trữ này chỉ được dựa vào sự thưa thớt, không đối xứng. Thật vậy, tại mỗi giai đoạn của sự tính toán, ma trận đang được thực thi ở trên là hermit. Điều này cho một thừa số khác của hai giai đoạn mà chúng có thể được tận dụng, mang lại tổng số việc làm ước lượng thành

$$\text{Sự giảm xuống thành 3 đường chéo: } \sim \frac{4}{3}m^3 \text{ phép toán số học dấu chấm động.} \tag{5.3.2}$$

5.3.5 Tính ổn định

Giống như thuật toán Householder cho phân tích QR, thuật toán vừa miêu ta là ổn định ngược. Nhắc lại từ Định lý 3.4.1 rằng, cho ma trận $A \in \mathbb{C}^{m\times n}$ bất kì, thuật toán Householder cho phân tích QR tính các vector phản xạ tương đương với một sự ẩn, chính xác thừa số unita $\tilde{Q}$ (3.4.2), cũng như một thừa số tam giác trên $\tilde{Q}$ rõ ràng, sao cho

$$\tilde{Q}\tilde{R} = A + \delta A, \ \frac{\|\delta A\|}{\|A\|} = O(\epsilon_{machine}).$$

Loại tương tự của ước lượng sai số có thể được thiết lặp cho Thuật toán 5.1. Cho $\tilde{H}$ là ma trận Hessenberg thực sự được tính trong số học dấu chấm động, và cho $\tilde{Q}$ như trên, là ma trận unita chính xác (3.4.2) tương ứng với các vector phản xạ $\tilde{v_k}$ được tính trong số học dấu chấm động. Kết quả theo sau có thể được chứng minh.

Định lý 5.3.1 *Cho sự giảm Hessenberg $A = QHQ^*$ của một ma trận $A \in \mathbb{C}^{m\times m}$ được tính bởi Thuật toán 5.1 trong một máy tính thỏa mãn các tiên đề (3.2.5) và (3.2.7), và cho các thừa số được tính $\tilde{Q}$ và $\tilde{H}$ được xác định như ở trên. Khi đó ta có*

$$\tilde{Q}\tilde{H}\tilde{Q}^* = A + \delta A, \ \frac{\|\delta A\|}{\|A\|} = O(\epsilon_{machine}) \tag{5.3.3}$$

với $\delta A \in \mathbb{C}^{m\times m}$ bất kì.

5.4 Tỷ số Rayleigh, bước lặp khả nghịch

5.4.1 Sự hạn chế của các ma trận đối xứng thực

Phần lớn các ý tưởng thuật toán là có thể dùng được hoặc tới các ma trận tổng quát hoặc, với các trường hợp đơn giản hóa nào đó, tới các ma trận hermit. Điều này tiếp tục đúng ít nhất một phần nào đó, nhưng một vài cái khác giữa các trường hợp tổng quát và hermit là khá lớn. Do đó, ta đơn giản các vấn đề này bằng việc chỉ xét các ma trận mà chúng là thực và đối xứng. Ta cũng giả sử $\|\cdot\| = \|\cdot\|_2$.

Khi đó, $A = A^T \in \mathbb{R}^{m\times m}, x \in \mathbb{R}^m, x^* = x^T, \|x\| = \sqrt{x^T x}$. Đặc biệt, điều này có nghĩa A có các trị riêng thực và một tập đầy đủ của các vector trực giao. Ta sử dụng ký hiệu theo sau:

$$\text{các trị riêng thực: } \lambda_1, \ldots, \lambda_2,$$
$$\text{các vector riêng trực giao: } q_1, \ldots, q_m.$$

Các vector riêng được cho là được chuẩn hóa bởi $\|q_j\| = 1$, và thứ tự của các trị riêng sẽ được chỉ rõ khi cần thiết.

Hầu hết các ý tưởng được miêu tả trong một vài mục tiếp theo đi đôi với Giai đoạn 2 của hai giai đoạn được miêu tả trong mục 5.2. Điều này nghĩa là vào thời điểm đó ta bắt đầu áp dụng các ý tưởng này, A sẽ không phải là thực và đối xứng mà là ba đường chéo. Cấu trúc ba đường chéo này là ngẫu nhiên của sự quan trọng toán học. ví dụ trong việc chọn các dịch chuyển cho phân tích QR, và nó thường là quan trọng mang tính thuật toán, việc giảm nhiều bước từ $O(m^3)$ thành $O(m)$ phép toán dấu chấm động, như được miêu tả tại phần cuối của mục này.

5.4.2 Tỷ số Rayleigh

Tỷ số Rayleigh của một vector $x \in \mathbb{R}^m$ là một vô hướng

$$r(x) = \frac{x^T A x}{x^T x}. \tag{5.4.1}$$

Nếu x là một vector riêng thì $r(x) = \lambda$ là trị riêng tương ứng. Một cách để thúc đẩy công thức này là để hỏi: cho x, vô hướng α "tác động giống một trị riêng nhất" cho x trong hướng cực tiểu hóa $\|Ax - \alpha x\|_2$ là gì? Đây là một bài toán bình phương nhỏ nhất $m \times 1$ của dạng $x\alpha \approx Ax$ (x là ma trận, α là vector không được biết, Ax là vế bên phải). Bằng việc viết các phương trình chính tắc (2.6.9) cho hệ thống này, ta thu được câu trả lời: $\alpha = r(x)$. Do đó $r(x)$ là ước lượng trị riêng tự nhiên để xét nếu x là gần với, nhưng không cần thiết bằng với, một trị riêng.

Để làm số lượng các ý tưởng này, nó là thành công để xem $x \in \mathbb{R}^m$ như là biến số, để r là một hàm $\mathbb{R}^m \to \mathbb{R}$. Ta bị hấp dẫn trong xử lý cục bộ của $r(x)$ khi x gần với một vector riêng. Một cách để xấp xỉ phương trình này là để tính các đạo hàm riêng của $r(x)$ tương ứng với các tọa độ x_j:

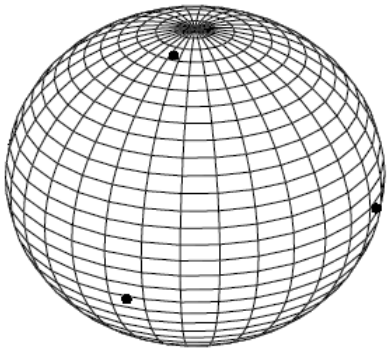
$$\begin{aligned}\frac{\partial r(x)}{\partial x_j} &= \frac{\frac{\partial}{\partial x_j}(x^T A x)}{x^T x} - \frac{(x^T A x)\frac{\partial}{\partial x_j}(x^T x)}{(x^T x)^2} \\ &= \frac{2(Ax)_j}{x^T x} - \frac{(x^T A x)2x_j}{(x^T x)^2} = \frac{2}{x^T x}(Ax - r(x)x)_j.\end{aligned}$$

Nếu ta tập hợp các đạo hàm riêng này thành một vector m chiều, ta tính *gradient* của $r(x)$, ký hiệu bởi $\nabla r(x)$

$$\nabla r(x) = \frac{2}{x^T x}(Ax - r(x)x). \tag{5.4.2}$$

Từ công thức này ta thấy rằng tại một vector riêng x của A, gradient của $r(x)$ là vector 0. Ngược lại, nếu $\nabla r(x) = 0$ với $x \neq 0$, thì x là một vector riêng và $r(x)$ là trị riêng tương ứng.

Nói về mặt hình học, các vector riêng của A là *các điểm dừng* của hàm $r(x)$, và các trị riêng của A là các giá trị của $r(x)$ tại các điểm dừng này. Trên thực tế, vì $r(x)$ là không phụ thuộc vào vô hướng của x, nên các điểm dừng này nằm dọc theo các đường qua gốc tọa độ trong $\mathbb{R}^m$. Nếu ta chuẩn tắc bằng việc hạn chế sự chú ý tới quả cầu đơn vị $\|x\| = 1$, thì chúng trở thành các điểm cô lập (giả sử rằng các trị riêng của A là đơn giản) như được thấy trong Hình 5.1



Hình 5.1: Tỷ số Rayleigh $r(x)$ là một hàm liên tục trong quả cầu đơn vị $\|x\| = 1$ trong $\mathbb{R}^m$, và các điểm dừng của $r(x)$ là các vector riêng được chuẩn hóa của A. Trong ví dụ này với $m = 3$, có 3 điểm dừng (tốt như các sự tương phản của chúng).

Cho q_J là một trong số các vector riêng của A. Từ $\nabla r(q_J) = 0$, cùng với tính trơn của hàm $r(x)$ (hầu khắp nơi ngoại trừ tại gốc $x = 0$), ta suy ra một kết quả quan trọng:

$$r(x) - r(q_J) = O(\|x - q_J\|^2) \text{ khi } x \to q_J. \tag{5.4.3}$$

Do đó tỷ số Rayleigh là một ước lượng *chính xác khả tích* của một trị riêng. Ở đây nằm ở lũy thừa của nó.

Một cách rõ ràng hơn để suy ra (5.4.3) là để mở rộng x như một tổ hợp tuyến tính của các vector riêng $q_1, \ldots, q_m$ của A. Nếu $x = \sum_{j=1}^m a_j q_j$, thì $r(x) = \sum_{j=1}^m a_j^2 \lambda_j / \sum_{j=1}^m a_j^2$. Do đó $r(x)$ là một trung bình có trọng số của các trị riêng của A, với các trọng số bằng các bình phương các tọa độ của x trong cơ sở vector riêng. Bởi vì việc bình phương của các tọa độ, không khó để thấy rằng nếu $|a_j/a_J| \leqslant \epsilon$ với mọi $j \neq J$, khi đó $r(x) - r(q_J) = O(\epsilon^2)$.

5.4.3 Bước lặp lũy thừa

Giả sử $v^{(0)}$ là một vector với $\|v^{0)}\| = 1$. Bước lặp lũy thừa được trích dẫn như một ý tưởng không tốt lúc bắt đầu tại mục 5.2. Nó có thể được mong đợi để đưa ra một chuỗi $v^{(i)}$ hội tụ về một vector riêng tương ứng với trị riêng lớn nhất của A.

Trong thuật toán này và các thuật toán theo sau, ta không chú ý tới các điều kiện kết thúc, miêu tả vòng lặp chỉ bằng biểu thức đề nghị "for $k = 1, 2, \ldots$ do". Dĩ nhiên, trong thực hành, các điều kiện kết thúc là rất quan trọng và điều này là một trong số những điểm mà phần mềm chất lượng cao như là có thể được tìm thấy trong LAPACK hoặc MATLAB có khuynh hướng là chất lượng cao tới một chương trình mà một cá nhân có thể viết.

Ta có thể phân tích bước lặp lũy thừa một cách dễ dàng. Viết $v^{(0)}$ như là một tổ hợp tuyến tính của các vector riêng trực giao q_i:

$$v^{(0)} = a_1 q_1 + a_2 q_2 + \ldots + a_m q_m.$$

Thuật toán 5.2 Bước lặp lũy thừa

1: $v^{(0)} =$ vector bất kì với $\|v^{(0)}\| = 1$
2: **for** $k = 1, 2, \ldots$ **do**
3: $\quad w = Av^{(k-1)}$ ▷ áp dụng A
4: $\quad v^{(k)} = w/\|w\|$ ▷ chuẩn hóa
5: $\quad \lambda^{(k)} = (v^{(k)})^T Av^{(k)}$ ▷ tỷ số Rayleigh
6: **end for**

Vì $v^{(k)}$ là một bội của $A^k v^{(0)}$ nên ta có một vài hằng số c_k

$$\begin{aligned} v^{(k)} &= c_k A^k v^{(0)} \\ &= c_k(a_1\lambda_1^k q_1 + a_2\lambda_2^k q_2 + \ldots + a_m\lambda_m^k q_m) \\ &= c_k\lambda_1^k(a_1 q_1 + a_2(\lambda_2/\lambda_1)^k q_2 + \ldots + a_m(\lambda_m/\lambda_1)^k q_m). \end{aligned} \tag{5.4.4}$$

Từ đây ta thu được kết quả theo sau.

Định lý 5.4.1 *Giả sử* $|\lambda_1| > |\lambda_2| \geqslant \cdots \geqslant |\lambda_m| \geqslant 0$ *và* $q_1^T v^{(0)} \neq 0$. *Khi đó các bước lặp của Thuật toán 5.2 thỏa mãn*

$$\|v^{(k)} - (\pm q_1)\| = O\left(\left|\frac{\lambda_2}{\lambda_1}\right|^k\right), \quad |\lambda^{(k)} - \lambda_1| = O\left(\left|\frac{\lambda_2}{\lambda_1}\right|^{2k}\right) \tag{5.4.5}$$

khi $k \to \infty$. *Dấu* $\pm$ *nghĩa là tại mỗi bước* k, *một hoặc sự lựa chọn khác của dấu là để được lấy, và khi đó biên được cho biết là đúng.*

Chứng minh Phương trình đầu tiên theo sau từ (5.4.4), vì $a_1 = q_1^T v^{(0)} \neq 0$ do giả thiết. Phương trình thứ 2 theo sau từ điều này và (5.4.3). Nếu $\lambda_1 > 0$ thì dấu $\pm$ là + hoặc - tất cả, trong khi nếu $\lambda_1 < 0$, chúng thay phiên nhau.

Dấu $\pm$ trong (5.4.5) và các phương trình tương tự bên dưới là không lôi cuốn. Có một cách tao nhã để tránh các sự phức tạp này, mà điều này đề cập tới sự hội tụ của các không gian con, không phải không gian vector - ví dụ, $< v^{(k)} >$ hội tụ về $< q_1 >$. Tuy nhiên, ta sẽ không làm điều này, để tránh việc đi vào các chi tiết của sự hội tụ như thế nào của các không gian con có thể được làm rõ ràng.

Bước lặp lũy thừa sử dụng có giới hạn vì một vài lý do. Đầu tiên, nó có thể chỉ tìm vector riêng ứng với trị riêng lớn nhất. Thứ hai, sự hội tụ là tuyến tính, làm giảm sai số chỉ bằng một thừa số hằng $\approx |\lambda_2/\lambda_1|$ tại mỗi bước lặp. Cuối cùng, số lượng của thừa số này phụ thuộc vào việc có một trị riêng lớn nhất mà nó lớn hơn những trị riêng khác. Nếu 2 trị riêng lớn nhất là gần nhau trong độ lớn, sự hội tụ sẽ là rất chậm.

5.4.4 Bước lặp nghịch đảo

Cho $\mu \in \mathbb{R}$ bất kì không là một trị riêng của A, các vector riêng của $(A - \mu I)^{-1}$ giống như các vector riêng của A, và các trị riêng tương ứng là $\{(\lambda_j - \mu)^{-1}\}$, với $\{\lambda_j\}$ là các trị riêng của A. Điều này đề nghị một ý tưởng. Giả sử μ là gần với một trị riêng λ_J của A. Khi đó $(\lambda_J - \mu)^{-1}$ có thể là lớn hơn nhiều $(\lambda_j - \mu)^{-1}$ với mọi $j \neq J$. Do đó, nếu ta áp dụng bước lặp lũy thừa tới $(A - \mu I)^{-1}$ thì quá trình sẽ hội tụ nhanh tới q_J. Ý tưởng ngày được gọi là *bước lặp nghịch đảo*.
Chuyện gì sẽ xảy ra nếu μ là một trị riêng của A, sao cho $A - \mu I$ là kỳ dị? Chuyện gì sẽ xảy ra nếu nó gần với một trị riêng, sao cho $A - \mu I$ là điều kiện quá xấu mà một lời giải chính xác

Thuật toán 5.3 Bước lặp nghịch đảo

1: $v^{(0)} =$ vector bất kì với $\|v^{(0)}\| = 1$
2: **for** $k = 1, 2, \ldots$ **do**
3: Giải $(A - \mu I)w = v^{(k-1)}$ cho biến w ▷ áp dụng $(A - \mu I)^{-1}$
4: $v^{(k)} = w/\|w\|$ ▷ chuẩn hóa
5: $\lambda^{(k)} = (v^{(k)})^T A v^{(k)}$ ▷ tỷ số Rayleigh
6: **end for**

của $(A - \mu I)w = v^{(k-1)}$ không thể được mong đợi không? Các cạm bẫy xuất hiện này của bước lặp nghịch đảo không có rắc rối nào cả.

Giống như bước lặp lũy thừa, bước lặp nghịch đảo chỉ đưa ra sự hội tụ tuyến tính. Không giống như bước lặp lũy thừa, ta có thể chọn vector riêng mà nó sẽ được tìm thấy bằng việc cung cấp một ước lượng μ của trị riêng tương ứng. Hơn nữa, tỉ lệ của sự hội tụ tuyến tính có thể được điều khiển vì nó phụ thuộc vào độ lớn của μ. Nếu μ gần với một trị riêng của A nhiều hơn những cái khác thì trị riêng lớn nhất của $(A - \mu I)^{-1}$ sẽ là lớn hơn nhiều phần còn lại. Việc sử dụng cùng một lý do như với bước lặp lũy thừa, ta thu được định lý theo sau.

Định lý 5.4.2 *Giả sử λ_J là trị riêng gần nhất với μ và λ_K là trị riêng gần nhất thứ hai, nghĩa là, $|\mu - \lambda_J| < |\mu - \lambda_K| \leqslant |\mu - \lambda_j|$ với mỗi $j \neq J$. Hơn nữa, giả sử $q_J^T v^{(0)} \neq 0$. Khi đó, các bước lặp của Thuật toán 5.3 thỏa mãn*

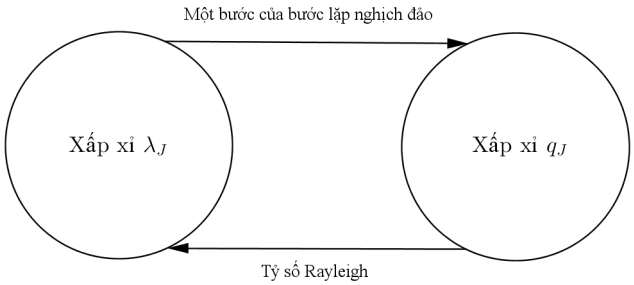
$$\|v^{(k)} - (\pm q_J)\| = O\left(\left|\frac{\mu - \lambda_J}{\mu - \lambda_K}\right|^k\right), \quad |\lambda^{(k)} - \lambda_J| = O\left(\left|\frac{\mu - \lambda_J}{\mu - \lambda_K}\right|^{2k}\right)$$

khi $k \to \infty$, dấu $\pm$ có nghĩa tương tự như trong Định lý 5.4.1.

Bước lặp nghịch đảo là một trong những công cụ có giá trị nhất của phương pháp số trong đại số tuyến tính vì nó là phương pháp tiêu chuẩn cho việc tính toán một hoặc nhiều vector riêng của một ma trận nếu các trị riêng đã biết. Trong trường hợp này Thuật toán 5.3 được áp dụng như được viết, ngoại trừ sự tính toán của tỷ số Rayleigh được bỏ qua.

5.4.5 Bước lặp tỷ số Rayleigh

Ta đã đưa ra một phương pháp thu được một ước lượng trị riêng từ một ước lượng vector riêng (tỷ số Rayleigh), và một phương pháp khác thu được ước lượng vector riêng từ ước lượng trị riêng (bước lặp nghịch đảo). Khả năng kết hợp các ý tưởng này là bất khả kháng: (Hình này



là quá đơn giản; để có được từ một xấp xỉ λ_J thành một xấp xỉ q_J bằng một bước của bước

lặp nghịch đảo, điều này cũng cần một sự xấp xỉ mở đầu thành q_J.) Ý tưởng là để sử dụng tiếp tục việc cải thiện các ước lượng trị riêng để tăng cường tỷ số hội tụ của bước lặp nghịch đảo tại mỗi bước. Thuật toán này được gọi là *bước lặp tỷ số Rayleigh.*
Sự hội tụ của thuật toán này: mỗi bước lặp nhân ba số chữ số của sự đúng đắn.

Thuật toán 5.4 Bước lặp tỷ số Rayleigh

1: $v^{(0)} =$ vector bất kì với $\|v^{(0)}\| = 1$
2: $\lambda^{(0)} = (v^{(0)})^T A v^{(0)} =$ tỷ số Rayleigh tương ứng
3: **for** $k = 1, 2, \ldots$ **do**
4: Giải $(A - \lambda^{(k-1)} I)w = v^{(k-1)}$ cho biến w ▷ áp dụng $(A - \lambda^{(k-1)} I)^{-1}$
5: $v^{(k)} = w/\|w\|$ ▷ chuẩn hóa
6: $\lambda^{(k)} = (v^{(k)})^T A v^{(k)}$ ▷ tỷ số Rayleigh
7: **end for**

Định lý 5.4.3 *Bước lặp tỷ số Rayleigh hội tụ tới một trị riêng/cặp vector riêng với mọi ngoại trừ một tập độ đo 0 của các vector bắt đầu $v^{(0)}$. Khi nó hội tụ, sự hội tụ là mặt bậc 3 trù mật trong hướng mà nếu λ_J là một trị riêng của A và $v^{(0)}$ là đủ gần với vector riêng q_J, khi đó*

$$\|v^{(k+1)} - (\pm q_J)\| = O(\|v^{(k)} - (\pm q_J)\|^3) \tag{5.4.6}$$

và

$$|\lambda^{(k+1)} - \lambda_J| = O(|\lambda^{(k)} - \lambda_J|^3) \tag{5.4.7}$$

khi $k \to \infty$. Dấu $\pm$ không cần thiết giống nhau trong 2 vế của (5.4.6).

Chứng minh Ta sẽ không chứng minh khẳng định về sự hội tụ cho hầu hết tất cả các vector bắt đầu. Tuy nhiên, đây là một chứng minh mà nếu sự hội tụ xuất hiện, nó là mặt bậc 3 trù mật. Cho đơn giản, ta giả sử rằng trị riêng λ_J là đơn. Do (5.4.3), nếu $\|v^{(k)} - q_J\| \leqslant \epsilon$ với ϵ đủ nhỏ, khi đó tỷ số Rayleigh cho một ước lượng trị riêng $\lambda^{(k)}$ với $|\lambda^{(k)} - \lambda_J| = O(\epsilon^2)$. Do đối số sử dụng để chứng minh Định lý 5.4.2, nếu bây giờ ta lấy một bước của bước lặp nghịch đảo để thu được $v^{(k+1)}$ mới từ $v^{(k)}$ và $\lambda^{(k)}$, khi đó

$$\|v^{(k+1)} - q_J\| = O(|\lambda^{(k)} - \lambda_J| \|v^{(k)} - q_J\|) = O(\epsilon^3).$$

Hơn nữa, các hằng số ẩn trong các ký hiệu O là giống nhau thông qua các lân cận đủ nhỏ của λ_J và q_J. Do đó ta có sự hội tụ trong mô hình theo sau:

$$\begin{array}{rr} \|v^{(k)} - (\pm q_J)\| & |\lambda^{(k)} - \lambda_J| \\ \epsilon \to & O(c^2) \\ \downarrow \swarrow & \\ O(\epsilon^3) \to & O(\epsilon^6) \\ \downarrow \swarrow & \\ O(\epsilon^9) \to & O(\epsilon^{18}) \\ \vdots & \vdots \end{array}$$

Các ước lượng (5.4.6) - (5.4.7) theo sau từ tính chất đồng nhất vừa được đề cập.

Ví dụ 5.4.1. Xét ma trận đối xứng

$$A = \begin{bmatrix} 2 & 1 & 1 \\ 1 & 3 & 1 \\ 1 & 1 & 4 \end{bmatrix},$$

và cho $v^{(0)} = (1,1,1)^T/\sqrt{3}$ là ước lượng vector riêng ban đầu. Khi bước lặp tỷ số Rayleigh được áp dụng cho A, các giá trị $\lambda^{(k)}$ theo sau được tính bằng 3 bước lặp đầu tiên:

$$\lambda^{(0)} = 5, \quad \lambda^{(1)} = 5.2131..., \quad \lambda^{(2)} = 5.214319743184....$$

Giá trị thực sự của trị riêng tương ứng với vector riêng gần với $v^{(0)}$ nhất là $\lambda = 5.214319743377$. Sau đó chỉ có 3 bước lặp, bước lặp tỷ số Rayleigh đã đưa ra một kết quả chính xác tới 10 chữ số. Các bước lặp lớn hơn 3 sẽ tăng con số này lên khoảng 270 chữ số, nếu sự đúng đắn máy của chúng ta là đủ cao.

5.4.6 Đếm số phép toán

Đầu tiên, giả sử $A \in \mathbb{R}^{m\times m}$ là một ma trận đầy đủ. Khi đó mỗi bước của bước lặp lũy thừa bao gồm một phép nhân ma trận với vector, yêu cầu $O(m^2)$ phép toán dấu chấm động. Mỗi bước của bước lặp nghịch đảo bao gồm lời giải của một hệ thống tuyến tính, yêu cầu $O(m^3)$ phép toán dấu chấm động, nhưng con số này giảm xuống $O(m^2)$ nếu ma trận được xử lý trước bởi phân tích LU hoặc phân tích QR hoặc phương pháp khác. Trong trường hợp của bước lặp tỷ số Rayleigh, ma trận để là các thay đổi được đảo ngược tại mỗi bước, và dao động $O(m^3)$ phép toán dấu chấm động trên bước là không quá đơn giản.

Các con số này cải thiện rất nhiều nếu A là 3 đường chéo. Cả 3 bước lặp yêu cầu đúng $O(m)$ phép toán dấu chấm động trên bước. Cho các bước lặp tương tự bao gồm các ma trận không đối xứng, ngẫu nhiên, ta phải có liên quan tới Hessenberg thay vì cấu trúc 3 đường chéo, và con số này tăng lên $O(m^2)$.

5.5 Phân tích QR không dịch chuyển

5.5.1 Phân tích QR

Kiểu cơ bản nhất của thuật toán QR

Thuật toán 5.5 Thuật toán QR "thuần túy"

1: $A^{(0)} = A$
2: **for** $k = 1, 2, \ldots$ **do**
3: $\quad Q^{(k)}R^{(k)} = A^{(k-1)}$ ▷ Phân tích QR của $A^{(k-1)}$
4: $\quad A^{(k)} = R^{(k)}Q^{(k)}$ ▷ Kết hợp các thừa số lại trong thứ tự ngược lại
5: **end for**

Ta lấy một phân tích QR, nhân các thừa số được tính Q và R với nhau trong thứ tự ngược lại RQ, và lặp lại. Thuật toán đơn giản này hội tụ về một dạng Schur cho ma trận tam giác trên A nếu A là bất kì, đường chéo nếu A là hermit. Ta sẽ tiếp tục giả sử A là ma trận thực và đối xứng, với các trị riêng thực λ_j và các vector riêng trực giao q_j. Do đó, các ma trận $A^{(k)}$ hội tụ thành dạng đường chéo.

Vì sự hội tụ thành dạng đường chéo là hữu ích cho việc tìm các trị riêng nên các phép toán được bao gồm phải là các biến đổi tương đương. Điều này được kiểm tra dễ dàng: thuật toán QR đầu tiên tam giác hóa $A^{(k)}$ bằng việc hình thành dạng $R^{(k)} = (Q^{(k)})^T A^{(k-1)}$, và nhân $Q^{(k)}$ vào vế phải cho $A^{(k)} = (Q^{(k)})^T A^{(k-1)} Q^{(k)}$. Thật vậy, ta đã thấy biến đổi tương đương này trước đây: nó là "Ý tưởng xấu" được đề cập trong mục 5.3. Mặc dù biến đổi này là ý tưởng xấu khi cố gắng để giảm A thành dạng 3 đường chéo trong một bước đơn, nó hóa ra là có tác động mạnh hoàn toàn như cơ sở của một bước lặp.

Giống như tỷ số Rayleigh, thuật toán QR cho các ma trận đối xứng thực hội tụ về dạng lập phương. Tuy nhiên, để đạt được sự thực thi này,thuật toán như được đưa ra ở trên phải được sửa đổi bằng sự đưa vào các dịch chuyển tại mỗi bước. Sử dụng các dịch chuyển là một trong 3 sự thay đổi của Thuật toán 5.5 mà chúng được yêu cầu để đưa nó gần hơn với một thuật toán thực tế:

1. Trước khi bắt đầu bước lặp, A được giảm thành dạng 3 đường chéo, như được thảo luận trong mục 5.3.

2. Thay vì $A^{(k)}$, một ma trận được dịch chuyển $A^{(k)} - \mu^{(k)}I$ được phân tích tại mỗi bước, với $\mu^{(k)}$ là ước lượng trị riêng bất kì.

3. Mỗi khi có thể thực hiện được, và đặc biệt mỗi khi một trị riêng được tìm thấy, bài toán được "giảm xuống" bằng việc biến $A^{(k)}$ thành các ma trận con.

Thuật toán QR kết hợp chặt chẽ với các sửa đổi này có trình bày sơ lược theo sau.

Thuật toán 5.6 Thuật toán QR "thực tế"

$(Q^{(0)})^T A^{(0)} Q^{(0)} = A$ ▷ $A^{(0)}$ là một tam giác hóa của A
for $k = 1, 2, \ldots$ **do**
  Chọn một dịch chuyển $\mu^{(k)}$ ▷ Ví dụ, chọn $\mu^{(k)} = A^{(k-1)}_{mm}$
  $Q^{(k)}R^{(k)} = A^{(k-1)} - \mu^{(k)}I$ ▷ Phân tích QR của $A^{(k-1)} - \mu^{(k)}I$
  $A^{(k)} = R^{(k)}Q^{(k)} + \mu^{(k)}I$ ▷ Kết hợp các thừa số lại trong thứ tự ngược lại
  if phần tử ngoài đường chéo bất kì $A^{(k)}_{j,j+1}$ là đủ gần 0 **then**
    Đặt $A_{j,j+1} = A_{j+1,j} = 0$ để thu được
    $$\begin{bmatrix} A_1 & 0 \\ 0 & A_2 \end{bmatrix} = A^{(k)}$$
    và bây giờ áp dụng thuật toán QR cho A_1 và A_2
  end if
end for

Thuật toán QR với các dịch chuyển được chọn tốt đã là một phương pháp tiêu chuẩn cho việc tính toán tất cả các trị riêng của một ma trận từ đầu những năm 1960. Chỉ trong những năm 1990 có một đối thủ mới nổi lên, thuật toán chia để trị miêu tả trong mục 5.7.

Tam giác hóa được miêu tả trong mục 5.3, các dịch chuyển được miêu tả trong mục tiếp theo, và sự giảm xuống không được thảo luận xa hơn trong sách này.Ta hãy hạn chế sự chú ý của chúng ta tới thuật toán QR "thuần túy" và giải thích làm thể nào tìm các trị riêng.

5.5.2 Bước lặp đồng thời không được chuẩn hóa

Xấp xỉ của chúng ta sẽ liên hệ thuật toán QR với phương pháp khác gọi là *bước lặp đồng thời*, mà xử lý của nó là rõ ràng hơn.

Ý tưởng của bước lặp đồng thời là để áp dụng bước lặp lũy thừa tới một vài vector tại một lần. (Số hạng tương đương là *bước lặp lũy thừa khối*.) Giả sử ta bắt đầu với một tập n vector độc lập tuyến tính $v_1^{(0)}, \ldots, v_n^{(0)}$. Ngay khi $A^k v_1^{(0)}$ hội tụ về vector riêng tương ứng với trị riêng lớn nhất của A trong giá trị tuyệt đối khi $k \to \infty$ (dưới các giả thiết phù hợp), không gian $< A^k v_1^{(0)}, \ldots, A^k v_n^{(0)} >$ nên hội tụ (nhắc lại dưới các giả thiết phù hợp) về một không gian $< q_1, \ldots, q_n >$ được sinh bởi các vector riêng $q_1, \ldots, q_n$ của A tương ứng với n trị riêng lớn nhất trong giá trị tuyệt tối.

Trong ký hiệu ma trận, định nghĩa $V^{(0)}$ là ma trận khởi tạo $m \times n$

$$V^{(0)} = \left[v_1^{(0)} | \ldots | v_n^{(0)}\right], \tag{5.5.1}$$

và định nghĩa $V^{(k)}$ là kết quả sau khi k lần áp dụng A:

$$V^{(k)} = A^k V^{(0)} = \left[v_1^{(k)} | \ldots | v_n^{(k)}\right]. \tag{5.5.2}$$

Trong không gian cột của $V^{(k)}$, ta hãy rút ra một cơ sở vận hành tốt cho không gian này bằng việc tính toán một phân tích QR được sửa đổi của $V^{(k)}$:

$$\hat{Q}^{(k)} \hat{R}^{(k)} = V^{(k)}. \tag{5.5.3}$$

Ở đây $\hat{Q}^{(k)}$ và $\hat{R}^{(k)}$ có số chiều tương ứng là $m \times n$ và $n \times n$. Khi $k \to \infty$, dưới các giả thiết phù hợp, các cột liên tiếp của $\hat{Q}^{(k)}$ sẽ hội tụ về các vector riêng $\pm q_1, \pm q_2, \ldots, \pm q_n$.

Nếu ta mở rộng $v_j^{(0)}$ và $v_j^{(k)}$ trong các vector riêng của A, ta có

$$\begin{aligned} v_j^{(0)} &= a_{1j} q_1 + \ldots + a_{mj} q_m, \\ v_j^{(k)} &= \lambda_1^k a_{1j} q_1 + \ldots + \lambda_m^k a_{mj} q_m \end{aligned}$$

Các kết quả hội tụ đơn giản sẽ được chứng minh rằng 2 điều kiện được thỏa mãn. Giả thiết đầu tiên ta làm $n+1$ trị riêng đầu được phân biệt trong giá trị tuyệt đối:

$$|\lambda_1| > |\lambda_2| > \ldots > |\lambda_n| > |\lambda_{n+1}| \geqslant |\lambda_{n+2}| \geqslant \ldots \geqslant |\lambda_m|. \tag{5.5.4}$$

Giả thiết thứ 2 của chúng ta là sự tập hợp các hệ số mở rộng a_{ij} trong xấp xỉ hướng không suy biến. Định nghĩa $\hat{Q}$ là ma trận $m \times n$ mà các cột của nó là các vector riêng $q_1, q_2, \ldots, q_n$. (Do đó $\hat{Q}$, một ma trận của các vector riêng, là khác nhau hoàn toàn với $\hat{Q}^{(k)}$, một thừa số trong phân tích QR được sửa đổi.) Ta giả sử theo sau:

$$\text{Tất cả các ma trận con cơ bản đầu của } \hat{Q}^T V^{(0)} \text{ là không suy biến.} \tag{5.5.5}$$

Các ma trận con cơ bản đầu của $\hat{Q}^T V^{(0)}$, nghĩa là các ma trận con vuông ở trên bên trái của nó có các số chiều $1 \times 1, 2 \times 2, \ldots, n \times n$. (Điều kiện (5.5.5) xảy ra là tương đương với điều kiện $\hat{Q}^T V^{(0)}$ có một phân tích LU)

Định lý 5.5.1 *Giả sử rằng bước lặp (5.5.1) - (5.5.3) được thực hiện và các giả thiết (5.5.4) và (5.5.5) được thỏa mãn. Do đó khi $k \to \infty$, các cột của các ma trận $\hat{Q}^{(k)}$ hội tụ tuyến tính về các vector riêng của A:*

$$\|q_j^{(k)} - \pm q_j\| = O(C^k) \tag{5.5.6}$$

với mọi j thỏa $1 \leqslant j \leqslant n$, $C < 1$ là hằng số $max_{1 \leqslant k \leqslant n} |\lambda_{k+1}|/|\lambda_k|$. Dấu $\pm$ nghĩa là tại mỗi bước k, một hoặc sự lựa chọn khác của dấu là được lấy, và khi đó biên được cho là đúng.

Chứng minh Mở rộng $\hat{Q}$ thành một ma trận trực giao $m \times m$ đầy đủ của các vector riêng của A, và cho Λ là ma trận đường chéo tương ứng của các trị riêng; do đó $A = Q \Lambda Q^T$. Ngay khi $\hat{Q}$ là phần $m \times n$ đầu của Q, định nghĩa $\hat{\Lambda}$ (vẫn đường chéo) là phần $n \times n$ đầu của Λ. Khi đó ta có

$$V^{(k)} = A^k V^{(0)} = Q \Lambda^k Q^T V^{(0)} = \hat{Q} \hat{\Lambda}^k \hat{Q}^T V^{(0)} + O(|\lambda_{n+1}|^k)$$

khi $k \to \infty$. Nếu (5.5.5) đúng thì đặc biệt, $\hat{Q}^T V^{(0)}$ là không suy biến, nên ta có thể nhân $(\hat{Q}^T V^{(0)})^{-1} \hat{Q}^T V^{(0)}$ vào vế phải của số hạng $O(\lambda_{n+1}^k)$ để biến đổi phương trình này thành

$$V^{(k)} = (\hat{Q} \hat{\Lambda}^k + O(|\lambda_{n+1}|^k)) \hat{Q}^T V^{(0)}.$$

Vì $\hat{Q}^T V^{(0)}$ là không suy biến nên không gian cột của ma trận này là giống như không gian cột của

$$\hat{Q}\hat{\Lambda}^k + O(|\lambda_{n+1}|^k).$$

Từ dạng của $\hat{Q}\hat{\Lambda}^k$ và giả thiết (5.5.4), không gian cột này hội tụ một cách tuyến tính về $\hat{Q}$. Sự hội tụ này có thể được xác định số lượng, ví dụ, bằng việc xác định các góc giữa các không gian con; ta bỏ các chi tiết.

Thật vậy, ta đã giả sử rằng không chỉ $\hat{Q}^T V^{(0)}$ là không suy biến mà còn tất cả các ma trận con cơ bản đầu của nó. Các đối số ở trên cũng áp dụng để đưa ra các tập con của các cột của $V^{(k)}$ và $\hat{Q}$: các cột đầu tiên, cột thứ nhất và cột thứ hai, cột thứ nhất và thứ hai và cột thứ 3, ... Trong mỗi trường hợp ta kết luận rằng không gian được sinh bởi các cột được chỉ ra của $V^{(k)}$ hội tụ tuyến tính về không gian được sinh bởi các cột tương ứng của $\hat{Q}$. Từ sự hội tụ này của tất cả các không gian cột liên tiếp, cùng với định nghĩa của phân tích QR (5.5.3), (5.5.6) theo sau.

5.5.3 Bước lặp đồng thời

Khi $k \to \infty$, các vector $v_1^{(k)}, \ldots, v_n^{(k)}$ trong thuật toán (5.5) - (5.7) tất cả hội tụ về các bội của vector riêng trội giống nhau q_1 của A. Do đó, mặc dù không gian mà chúng sinh ra, $< v_1^{(k)}, \ldots, v_j^{(k)} >$, hội tụ về một vài thứ hữu ích, các vector này tạo thành cơ sở điều kiện xấu ở mức độ rất cao của không gian đó. Nếu ta thực sự tiến hành bước lặp đồng thời trong số học dấu chấm động như vừa được miêu tả, thông tin được miêu tả sẽ làm hao hụt nhanh các sai số làm tròn.

Biện pháp sửa chữa sai lầm là đơn giản: ta phải trực giao hóa tại mỗi bước hơn 1 lần và cho tất cả. Do đó ta sẽ không xây dựng $V^{(k)}$ như được xác định ở trên, nhưng một chuỗi khác nhau của các ma trận $Z^{(k)}$ với các không gian cột giống nhau. Từ dạng của thuật toán này,

Thuật toán 5.7 Bước lặp đồng thời

1: Chọn $\hat{Q^{(0)}} \in \mathbb{R}^{m \times n}$ với các cột trực giao
2: **for** $k = 1, 2, \ldots$ **do**
3: $\quad Z = A\hat{Q}^{(k-1)}$
4: $\quad Q^{(k)}R^{(k)} = Z$ ▷ Phân tích QR được sửa đổi của Z
5: **end for**

các không gian cột của $\hat{Q}^{(k)}$ và $Z^{(k)}$ là giống nhau, cả hai đều tương đương không gian cột của $A^k\hat{Q}^{(0)}$. Do đó, sự hình thành công thức mới này của bước lặp đồng thời hội tụ dưới cùng các điều kiện như cũ.

Định lý 5.5.2 *Thuật toán 5.7 sinh ra các ma trận giống nhau $\hat{Q}^{(k)}$ như bước lặp (5.5.1) - (5.5.3) được xét trong Định lý 5.5.1 (giả sử rằng các ma trận khởi tạo $\hat{Q}^{(0)}$ là giống nhau), và dưới cùng các giải thiết (5.5.4) và (5.5.5), nó hội tụ như được miêu tả trong định lý đó.*

5.5.4 Bước lặp đồng thời ⇔ Phân tích QR

Bây giờ ta có thể giải thích thuật toán QR. Nó tương đương với bước lặp đồng thời được áp dụng tới một tập đầy đủ của $n = m$ vector khởi tạo, cụ thể, ma trận đơn vị, $\hat{Q}^{(0)} = I$. Vì các ma trận $\hat{Q}^{(k)}$ là vuông nên ta đang giải quyết các phân tích QR đầy đủ và có thể giảm các mũ trong $\hat{Q}^{(k)}$ và $\hat{R}^{(k)}$. Thật vậy, ta sẽ thay thế $\hat{R}^{(k)}$ bằng $R^{(k)}$ nhưng $\hat{Q}^{(k)}$ bằng $\underline{Q}^{(k)}$ để phân biệt các ma trận Q của bước lặp đồng thời từ những cái này của thuật toán QR.

Ở đây là 3 công thức mà nó xác định bước lặp đồng thời với $\underline{Q}^{(0)} = I$, được theo sau bởi công thức thứ 4 mà ta sẽ lấy như một định nghĩa của ma trận $A^{(k)}$ cấp $m \times m$:

Bước lặp đồng thời:

$$\underline{Q}^{(0)} = I, \tag{5.5.7}$$

$$Z = A\underline{Q}^{(k-1)}, \tag{5.5.8}$$

$$Z = \underline{Q}^{(k)} R^{(k)} \tag{5.5.9}$$

$$A^{(k)} = (\underline{Q}^{(k)})^T A \underline{Q}^{(k)}. \tag{5.5.10}$$

Và đây là 3 công thức xác định thuật toán QR thuần túy, được theo sau bởi một công thức thứ 4 mà ta sẽ lấy như là một định nghĩa của ma trận $\underline{Q}^{(k)}$ có cấp $m \times m$:

Thuật toán QR không được dịch chuyển:

$$A^{(0)} = A, \tag{5.5.11}$$

$$A^{(k-1)} = Q^{(k)} R^{(k)}, \tag{5.5.12}$$

$$A^{(k)} = R^{(k)} Q^{(k)}, \tag{5.5.13}$$

$$\underline{Q}^{(k)} = Q^{(1)} Q^{(2)} \dots Q^{(k)}. \tag{5.5.14}$$

Hơn nữa, cho cả hai thuật toán này, ta hãy xác định một ma trận $\underline{R}^{(k)}$ có cấp $m \times m$ xa hơn,

$$\underline{R}^{(k)} = R^{(k)} R^{(k-1)} \dots R^{(1)}. \tag{5.5.15}$$

Bây giờ ta có thể đưa ra sự tương đương của hai thuật toán này.

Định lý 5.5.3 *Các quá trình (5.5.7) - (5.5.10) và (5.5.11) - (5.5.14) sinh ra các chuỗi đồng nhất của các ma trận $\underline{R}^{(k)}, \underline{Q}^{(k)}$ và $A^{(k)}$, cụ thể, các ma trận này được xác định bởi phân tích QR của lũy thừa thứ k của A,*

$$A^k = \underline{Q}^{(k)} \underline{R}^{(k)}, \tag{5.5.16}$$

cùng với phép chiếu

$$A^{(k)} = (\underline{Q}^{(k)})^T A \underline{Q}^{(k)}. \tag{5.5.17}$$

Chứng minh Ta tiến hành bằng qui nạp theo k. Trường hợp cơ sở $k = 0$ là tầm thường. Cho cả hai bước lặp đồng thời và thuật toán QR, các phương trình (5.5.7) - (5.5.15) kéo theo $A^0 = \underline{Q}^{(0)} = \underline{R}^{(0)} = I$ và $A^{(0)} = A$, từ (5.5.16) và (5.5.17) là trực tiếp.

Bây giờ xét trường hợp $k \geqslant 1$ cho bước lặp đồng thời. Công thức (5.5.17) là chắc chắn do định nghĩa (5.5.10) (chúng là đồng nhất), nên ta chỉ cần kiểm tra (5.5.16), mà nó có thể được làm như sau:

$$A^k = A\underline{Q}^{(k-1)} \underline{R}^{(k-1)} = \underline{Q}^{(k)} R^{(k)} \underline{R}^{(k-1)} = \underline{Q}^{(k)} \underline{Q}^{(k)}.$$

Phương trình đầu tiên theo sau từ giả thiết qui nạp trong (5.5.16), phương trình thứ 2 từ (5.5.8) và (5.5.9), và phương trình thứ 3 từ (5.5.15).

Mặc khác, xét trường hợp $k \geqslant 1$ cho phân tích QR. Ta có thể kiểm tra (5.5.16) bằng chuỗi

$$A^k = A\underline{Q}^{(k-1)} \underline{R}^{(k-1)} = \underline{Q}^{(k-1)} A^{(k-1)} \underline{R}^{(k-1)} = \underline{Q}^{(k)} \underline{R}^{(k)}.$$

Phương trình đầu tiên theo sau từ giả thiết qui nạp trong (5.5.16), phương trình thứ 2 từ giả thiết qui nạp (5.5.17), và phương trình thứ 3 từ (5.5.12), cùng với (5.5.14) và (5.5.15). Cuối cùng, ta có thể kiểm tra (5.5.17) bằng chuỗi

$$A^{(k)} = (Q^{(k)})^T A^{(k-1)} Q^{(k)} = (\underline{Q}^{(k)})^T A \underline{Q}^{(k)}.$$

Phương trình đầu tiên theo sau từ (5.5.12) và (5.5.13), và phương trình thứ 2 từ giả thiết qui nạp (5.5.17).

5.5.5 Sự hội tụ của thuật toán QR

Đầu tiên, (5.5.16) và (5.5.17) là khóa. Điều tiên của các công thức này giải thích vì sao thuật toán QR có thể được biểu diễn để tìm các vector riêng: nó xây dựng các cơ sở trực giao cho các lũy thừa A^k liên tiếp. Thứ hai giải thích vì sao thuật toán tìm các trị riêng. Từ (5.5.17) nó theo sau rằng các phần tử trên đường chéo của A^k là các tỷ số Rayleigh của A tương ứng với các cột của $Q^{(k)}$. Khi các cột này hội tụ về các vector riêng, các tỷ số Rayleigh hội tụ (nhanh gấp đôi, do (5.4.3)) tới các trị riêng tương ứng. Trong lúc đó, (5.5.17) kéo theo rằng các phần tử ngoài đường chéo của $A^{(k)}$ tương ứng sinh ra các tỷ số Rayleigh bao gồm các xấp xỉ của các vector riêng phân biệt của A trong vế trái và vế phải. Vì các xấp xỉ này phải trực giao khi chúng hội tụ về các vector riêng phân biệt, các phần tử ngoài đường chéo của $A^{(k)}$ phải hội tụ về 0.

Ta không thể làm nổi bật quá mạnh các phương trình cơ bản (5.5.16) và (5.5.17) là để hiểu thuật toán QR không được dịch chuyển như thế nào.

Định lý 5.5.4 *Cho thuật toán QR thuần túy (Thuật toán 5.5) được áp dụng tới một ma trận đối xứng thực A mà các trị riêng của nó thỏa mãn $|\lambda_1 > |\lambda_2| > \ldots > |\lambda_m|$ và ma trận vector riêng tương ứng của nó Q có tất cả các ma trận con cơ bản đầu không suy biến. Do đó, khi $k \to \infty$, $A^{(k)}$ hội tụ tuyến tính với hằng số $max_j|\lambda_{j|1}/|\lambda_j|$ về $diag(\lambda_1, \ldots, \lambda_m)$, và $\underline{Q}^{(k)}$ (với dấu của các cột của nó được điều chỉnh khi cần thiết) hội tụ về Q tại cùng tỷ số.*

5.6 Phân tích QR với các dịch chuyển

5.6.1 Sự kết hợp với bước lặp nghịch đảo

Ta tiếp tục giả sử rằng $A \in \mathbb{R}^{m\times m}$ là thực và đối xứng, với các trị riêng thực $\{\lambda_j\}$ và các vector trực giao $\{q_j\}$.

Thuật toán QR "thuần túy" (Thuật toán 5.5) là tương đương với bước lặp đồng thời được áp dụng cho ma trận đơn vị, và đặc biệt, cột đầu tiên của kết quả suy ra theo bước lặp lũy thừa được áp dụng cho e_1. Có một đối ngẫu tới sự quan sát này. Thuật toán 5.5 cũng tương đương với *bước lặp nghịch đảo đồng thời* được áp dụng cho một ma trận đơn vị "được lật nhanh", và đặc biệt, cột thứ m của kết quả suy ra theo bước lặp nghịch đảo được áp dụng cho e_m.

Ta có thể thiết lập đòi hỏi này như sau. Cho $Q^{(k)}$ là thừa số trực giao tại bước thứ k của thuật toán QR. Trong mục cuối cùng, ta chứng minh rằng tích được tích lũy (5.5.14) của các ma trận này,

$$\underline{Q}^{(k)} = \prod_{j=1}^{k} Q^{(j)} = \left[q_1^{(k)}|q_2^{(k)}|\ldots|q_m^{(k)}\right],$$

là ma trận trực giao giống nhau xuất hiện tại bước k (5.5.8) của bước lặp đồng thời. Cách khác để làm điều này nói rằng $\underline{Q}^{(k)}$ là thừa số trực giao trong phân tích QR (5.5.16),

$$A^k = \underline{Q}^{(k)}\underline{R}^{(k)}.$$

Bây giờ xét cái gì xảy ra nếu ta lấy nghịch đảo công thức này. Ta tính

$$A^{-k} = (\underline{R}^{(k)})^{-1}\underline{Q}^{(k)T} = \underline{Q}^{(k)}(\underline{R}^{(k)})^{-T}; \tag{5.6.1}$$

vì phương trình thứ hai ta đã sử dụng A^{-1} là đối xứng. Cho P ký hiệu là ma trận hoán vị $m \times m$ mà nó đảo ngược thứ tự dòng hoặc cột:

$$P = \begin{bmatrix} & & & 1 \\ & & 1 & \\ & \cdots & & \\ 1 & & & \end{bmatrix}.$$

Vì $P^2 = I$, (5.6.1) có thể được viết lại như

$$A^{-k}P = [\underline{Q}^{(k)}P][P(\underline{R}^{(k)})^{-T}P]. \tag{5.6.2}$$

Thừa số đầu tiên trong tích này, $\underline{Q}^{(k)}P$, là trực giao. Thứ hai, $P(\underline{R}^{(k)})^{-T}P$, là ma trận tam giác trên (bắt đầu với ma trận tam giác dưới $(\underline{R}^{(k)})^{-T}$, lật nhanh nó từ trên xuống dưới, khi đó lật nhanh nó từ trái qua phải lần nữa). Do đó (5.6.2) có thể được hiểu như một phân tích QR của $A^{-k}P$. Mặt khác, ta đang tiến hành một cách hiệu quả bước lặp đồng thời trong A^{-1} được áp dụng cho ma trận khởi tạo P, bước lặp nghịch đảo đồng thời trong A. Đặc biệt, cột thứ nhất của $\underline{Q}^{(k)}P$ - cột cuối cùng của $\underline{Q}^{(k)}$ - là kết quả của việc áp dụng k bước của bước lặp nghịch đảo cho vector e_m.

5.6.2 Sự kết hợp với bước lặp khả nghịch được dịch chuyển

Do đó thuật toán QR là cả hai bước lặp đồng thời và bước lặp nghịch đảo đồng thời: đối xứng là đầy đủ. Nhưng, như ta thấy trong mục 5.4, có một sự khác nhau lớn giữa bước lặp lũy thừa và bước lặp nghịch đảo: cái sau có thể được làm nhanh một cách tùy ý thông qua sử dụng các dịch chuyển. Ta càng có thể ước lượng một trị riêng $\mu \approx \lambda_J$, ta sẽ càng hoàn thành bằng một bước của bước lặp nghịch đảo với ma trận được dịch chuyển $A - \mu I$. Thuật toán 5.6 cho thấy các dịch chuyển được đưa ra thành một bước của thuật toán QR như thế nào. Việc làm điều này tương ứng một cách chính xác tới các dịch chuyển trong các xử lý bước lặp đồng thời và bước lặp nghịch đảo tương ứng, và do đó hiệu quả có ích của chúng là giống nhau.

Cho $\mu^{(k)}$ ký hiệu ước lượng trị riêng được chọn tại bước thứ k của thuật toán QR. Từ Thuật toán 5.6, quan hệ giữa các bước $k-1$ và k của thuật toán QR được dịch chuyển là

$$\begin{aligned} A^{(k-1)} - \mu^{(k)}I &= Q^{(k)}R^{(k)}, \\ A^{(k)} &= R^{(k)}Q^{(k)} + \mu^{(k)}I. \end{aligned}$$

Điều này kéo theo

$$A^{(k)} = (Q^{(k)})^T A^{(k-1)} Q^{(k)}, \tag{5.6.3}$$

và do qui nạp,

$$A^{(k)} = (\underline{Q}^{(k)})^T A \underline{Q}^{(k)}, \tag{5.6.4}$$

mà nó không được thay đổi từ (5.5.17). Tuy nhiên, (5.5.16) không còn đúng. Thay thế, ta có phân tích

$$(A - \mu^{(k)}I)(A - \mu^{(k-1)}I) \cdot (A - \mu^{(1)}I) = \underline{Q}^{(k)}\underline{R}^{(k)}, \tag{5.6.5}$$

một sự biến đổi được dịch chuyển trong bước lặp đồng thời (ta bỏ qua chứng minh). Mặt khác, $\underline{Q}^{(k)} = \prod_{j=1}^{k} Q^{(j)}$ là một sự trực giao hóa của $\prod_{j=k}^{1}(A - \mu^{(j)}I)$. Cột đầu tiên của $\underline{Q}^{(k)}$ là kết quả của việc áp dụng bước lặp lũy thừa được dịch chuyển cho e_1 sử dụng các dịch chuyển $\mu^{(j)}$, và cột cuối cùng là kết quả của việc áp dụng k bước của bước lặp nghịch đảo được dịch chuyển cho e_m với cùng các dịch chuyển. Nếu các dịch chuyển là các ước lượng trị riêng tốt thì cột cuối cùng này của $\underline{Q}^{(k)}$ hội tụ nhanh về một vector riêng.

5.6.3 Sự kết hợp với xấp xỉ tỷ số Rayleigh

Ta đã phát hiện một công cụ mạnh ẩn trong thuật toán QR được dịch chuyển: bước lặp nghịch đảo được dịch chuyển. Để hoàn thành ý tưởng, bây giờ ta cần một cách của việc chọn các dịch chuyển để đạt được sự hội tụ nhanh trong cột cuối cùng của $\underline{Q}^{(k)}$.

Tỷ số Rayleigh là một nơi tốt để bắt đầu. Để ước lượng trị riêng tương ứng với vector riêng được xấp xỉ bởi cột cuối cùng của $\underline{Q}^{(k)}$, nó là tự nhiên để áp dụng tỷ số Rayleigh tới cột cuối cùng này. Điều này cho chúng ta

$$\mu^{(k)} = \frac{(q_m^{(k)})^T A q_m^{(k)}}{(q_m^{(k)})^T q_m^{(k)}} = (q_m^{(k)})^T A q_m^{(k)}. \tag{5.6.6}$$

Nếu con số này được chọn như dịch chuyển tại mỗi bước thì các ước lượng trị riêng và vector riêng $\mu^{(k)}$ và $q_m^{(k)}$ là đồng nhất với cái mà chúng được tính bằng bước lặp tỷ số Rayleigh bắt đầu với e_m. Do đó, thuật toán QR có sự hội tụ mặt bậc 3 trong ý nghĩa mà $q_m^{(k)}$ hội tụ về một vector riêng.

Chú ý rằng, trong thuật toán QR, tỷ số Rayleigh $r(q_m^{(k)})$ xuất hiện như phần tử m, m của $A^{(k)}$.Bắt đầu với (5.6.3), ta có

$$A_{mm}^{(k)} = e_m^T A^{(k)} e_m = e_m^T \underline{Q}^{(k)T} A \underline{Q}^{(k)} e_m = q_m^{(k)T} A q_m^{(k)}. \tag{5.6.7}$$

Do đó, (5.6.6) là giống nhau khi việc đặt một cách đơn giản $\mu^{(k)} = A_{mm}^{(k)}$. Điều này được biết như *dịch chuyển tỷ số Rayleigh.*

5.6.4 Dịch chuyển Wilkinson

Mặc dù dịch chuyển tỷ số Rayleigh cho sự hội tụ mặt bậc 3 trong trường hợp tổng quát, nhưng sự hội tụ không được bảo đảm với mọi điều kiện khởi tạo. Ta có thể thấy điều này với một ví dụ đơn giản. Xét ma trận

$$A = \begin{bmatrix} 0 & 1 \\ 1 & 0 \end{bmatrix} \tag{5.6.8}$$

Thuật toán QR không được dịch chuyển không hội tụ tất cả:

$$A = Q^{(1)} R^{(1)} = \begin{bmatrix} 0 & 1 \\ 1 & 0 \end{bmatrix} \begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix},$$
$$A^{(1)} = R^{(1)} Q^{(1)} = \begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix} \begin{bmatrix} 0 & 1 \\ 1 & 0 \end{bmatrix} = A.$$

Tuy nhiên, dịch chuyển tỷ số Rayleigh $\mu = \Lambda_{mm}$ không có hiệu quả vì $A_{mm} = 0$. Do đó, trong trường hợp xấu nhất, thuật toán QR với dịch chuyển tỷ số Rayleigh có thể thất bại.

Bài toán xuất hiện bởi vì sự đối xứng của các trị riêng. Một trị riêng là +1, và một trị riêng khác là -1, nên khi ta cố gắng để cải thiện ước lượng trị riêng 0, hướng tới sự cho phép mỗi trị riêng là bằng nhau, và ước lượng không được chứng minh. Cái gì được cần là một ước lượng trị riêng mà nó có phá vỡ tính đối xứng. Một sự lựa chọn như vậy được xác định như sau. Cho B ký hiệu ma trận con 2×2 thấp hơn bên phải nhất của $A^{(k)}$:

$$B = \begin{bmatrix} a_{m-1} & b_{m-1} \\ b_{m-1} & a_m \end{bmatrix}.$$

Dịch chuyển Wilkinson được xác định như là trị riêng của B mà nó gần hơn với a_m, trong đó trường hợp của một quan hệ, một trong hai trị riêng của B đươc chọn một cách tùy ý. Một công thức ổn định số cho dịch chuyển Wilkinson là

$$\mu = a_m - sign(\delta) b_{m-1}^2 \Big/ \left(|\delta| + \sqrt{\delta^2 + b_{m-1}^2} \right), \tag{5.6.9}$$

với $\delta = (a_{m-1} - a_m)/2$. Nếu $\delta = 0$ thì $sign(\delta)$ có thể là một tập tùy ý bằng 1 hoặc -1.

Giống như dịch chuyển Rayleigh, dịch chuyển Wilkinson đạt được sự hội tụ mặt bậc 3 trong trường hợp chung. Hơn nữa, nó có thể được cho thấy rằng nó đạt được tại sự hội tụ bình phương nhỏ nhất trong trường hợp xấu nhất. Đặc biệt, thuật toán QR với dịch chuyển Wilkinson thường hội tụ (trong số học chính xác).

Trong ví dụ (5.6.8), dịch chuyển Wilkinson hoặc là 1 hoặc là -1. Do đó sự đối xứng bị phá vỡ, và sự hội tụ diễn ra trong một bước.

5.6.5 Tính ổn định và sự đúng đắn

Điều này hoàn thành thảo luận của chúng ta về cơ học của thuật toán QR, mặc dù nhiều chi tiết thiết thực đã bị bỏ đi, như các điều kiện cho sự giảm xuống và các chiến lược "ẩn" cho việc dịch chuyển.

Khi ta có thể mong đợi từ sự sử dụng các ma trận trực giao của nó, thuật toán QR là ổn định ngược. Như trong các mục trước, cách đơn giản nhất để đưa ra kết quả này là đặt $\tilde{\Lambda}$ ký hiệu sự trực giao hóa của A khi được tính toán trong số học dấu chấm động, và $\tilde{Q}$ ma trận trực giao một cách chính xác được kết hợp với tích của tất cả các phản xạ Householder được tính về số lượng (hoặc các ký hiệu đã cho) được sử dụng cho cách này.

Định lý 5.6.1 *Cho ma trận thực, đối xứng, 3 đường chéo $A \in \mathbb{R}^{m\times m}$ được chéo hóa bởi thuật toán QR (Thuật toán 5.6) trong một máy tính thỏa (3.2.5) và (3.2.7), và cho $\tilde{\Lambda}$ và $\tilde{Q}$ được xác định như ở trên. Khi đó ta có*

$$\tilde{Q}\tilde{\Lambda}\tilde{Q}^* = A + \delta A, \quad \frac{\|\delta A\|}{\|A\|} = O(\epsilon_{machine}) \tag{5.6.10}$$

với $\delta A \in \mathbb{C}^{m\times m}$ bất kì.

Giống như hầu hết các thuật toán trong sách này, khi đó thuật toán QR đưa ra một lời giải chính xác của một bài toán được làm nhiễu nhỏ. Kết hợp Định lý 5.3.1 và 5.6.1, ta thấy rằng sự giảm thành dạng 3 đường chéo theo sau bởi thuật toán QR là thuật toán ổn định ngược cho tính toán các trị riêng của các ma trận. Các trị riêng được tính $\tilde{\lambda}_j$ thỏa mãn

$$\frac{|\tilde{\lambda}_j - \lambda_j|}{\|A\|} = O(\epsilon_{machine}). \tag{5.6.11}$$

Đây là kết quả không tệ cho một thuật toán mà nó yêu cầu đúng $\frac{4}{3}m^3$ phép toán dấu chấm động, 2/3 chi phí của việc tính toán tích của một cặp các ma trận $m \times m$.

5.7 Các thuật toán trị riêng khác

5.7.1 Thuật toán Jacobi

Một trong những ý tưởng cũ nhất cho việc tính toán các trị riêng của các ma trận là *thuật toán Jacobi*, được đưa ra bởi Jacobi vào năm 1845. Phương pháp này đã lôi cuốn sự chú ý khắp kỷ nguyên máy tính, đặc biệt từ sự xuất hiện của tính toán song song, mặc dù nó đã không bao giờ sử dụng hầu hết để thể hiện sự cạnh tranh.

Ý tưởng như sau. Cho các ma trận có số chiều lớn hơn hoặc bằng 5, ta biết rằng các trị riêng có thể chỉ được đạt được bằng bước lặp (mục 5.2). Tuy nhiên, các ma trận nhỏ hơn điều này có thể được sử dụng trong một bước. Vì sao không chéo hóa một ma trận con nhỏ hơn của A, sau đó là một cái khác, và v.v, hy vọng cuối cùng để hội tụ về một sự chéo hóa của ma trận đầy đủ?

Ý tưởng đã được thử với các ma trận con 4×4, nhưng xấp xỉ tiêu chuẩn được dựa vào các ma trận con 2×2. Một ma trận đối xứng thực 2×2 có thể được chéo hóa trong dạng

$$J^T \begin{bmatrix} a & d \\ d & b \end{bmatrix} J = \begin{bmatrix} \neq 0 & 0 \\ 0 & \neq 0 \end{bmatrix}, \tag{5.7.1}$$

với J là trực giao. Bây giờ có một vài cách để chọn J. Ta có thể lấy nó là một phản xạ Householder 2×2 của dạng

$$F = \begin{bmatrix} -c & s \\ s & c \end{bmatrix}, \tag{5.7.2}$$

với $s = \sin\theta$ và $c = \cos\theta$ với θ bất kì. Chú ý rằng $detF = -1$, xác nhận tiêu chuẩn của một phản xạ. Ngoài ra, ta có thể không sử dụng một phản xạ nhưng dùng một phép quay,

$$J = \begin{bmatrix} c & s \\ -s & c \end{bmatrix}, \tag{5.7.3}$$

với $detJ = 1$. Điều này là một xấp xỉ tiêu chuẩn cho thuật toán Jacobi. Nó có thể được cho thấy rằng sự chéo hóa (5.7.1) được thực hiện nếu θ thỏa mãn

$$\tan(2\theta) = \frac{2d}{b-a}, \tag{5.7.4}$$

và ma trận J được dựa vào sự lựa chọn này được gọi là *phép quay Jacobi*. (Nó có dạng giống như phép quay Givens; chỉ khác nhau là θ được chọn để làm đường chéo J^TAJ hơn là tam giác J^TA.)

Bây giờ cho $A \in \mathbb{R}^{m\times m}$ là đối xứng. Thuật toán Jacobi bao gồm sự áp dụng bước lặp của các biến đổi (5.7.1) dựa vào các ma trận được xác định bởi (5.7.3) và (5.7.4). Ma trận J được mở rộng thành một ma trận $m \times m$ mà nó là đồng nhất trong tất cả nhưng cho 4 phần tử, mà nó có dạng (5.7.3). Việc áp dụng J^T trong vế trái thay đổi 2 dòng của A, và áp dụng J vào trong vế phải thay đổi 2 cột. Tại mỗi bước một cặp đối xứng của các số 0 được đưa vào ma trận, nhưng các số 0 trước bị triệt tiêu. Ngay như với thuật toán QR, hiệu quả chung là các độ lớn của các số khác 0 này rút lại đều đặn.

Các phần tử ngoài đường chéo a_{ij} sẽ được làm thành 0 tại mỗi bước? Xấp xỉ phù hợp một cách tự nhiên để đưa sự tính toán là để chọn phần tử lớn nhất ngoài đường chéo tại mỗi bước. Khi đó phân tích sự hội tụ trở nên không tầm thường, vì ta có thể cho thấy rằng tổng của các bình phương của các phần tử ngoài đường chéo giảm xuống ít nhất một thừa số $1-2/(m^2-m)$ tại mỗi bước. Sau $O(m^2)$ bước, mỗi $O(m)$ phép toán yêu cầu, tổng của các bình phương phải giảm xuống một thừa số hằng, và hội tụ về độ chính xác $\epsilon_{machine}$ được đảm bảo sau $O(m^3 \log(\epsilon_{machine}))$ phép toán. Thật vậy, nó được biết mà sự hội tụ là tốt hơn điều này, toàn phương trù mật hơn là tuyến tính, nên đếm số phép toán thực sự là $O(m^3 \log(|\log(\epsilon_{machine})|))$.

Trong một máy tính, các phần tử ngoài đường chéo nói chung được giới hạn trong một dạng vòng mà nó tránh tìm kiếm $O(m^2)$ cho phần tử lớn nhất. Ví dụ, nếu $m(m-1)/2$ phần tử siêu đường chéo được giới hạn trong thứ tự mức độ dòng đơn giản nhất, bắt đầu với $a_{12}, a_{13}, \ldots$, khi đó sự hội tụ tiệm cận nhanh được đảm bảo lần nữa. Sau *sự rà sót* của các phép toán 2×2 gồm tất cả $m(m-1)/2$ cặp của các phần tử ngoài đường chéo chính, sự đúng đắn nói chung đã cải thiện bằng một thừa số hằng tốt hơn, và sự hội tụ là bình phương trù mật.

Phương pháp Jacobi hấp dẫn bởi vì nó chỉ giải quyết các cặp của các dòng và các cột một lần, làm nó song song dễ dàng. Ma trận không được chéo hóa 3 đường chéo trước; các phép quay Jacobi sẽ triệt tiêu cấu trúc đó. Sự hội tụ cho các ma trận có số chiều $m \leqslant 1000$ được đạt được một cách tiêu biểu trong ít hơn 10 sự rà sót, và sự đúng đắn theo từng thành phần cuối cùng nói chúng là tốt hơn có thể được đạt được bởi thuật toán QR. Không may mắn, các máy song song, thuật toán Jacobi thường không nhanh như chéo hóa 3 đường chéo bởi thuật toán QR hoặc thuật toán chia để trị (được thảo luận bên dưới), mặc dù nó thường đến trong 1 thừa số của 10.

5.7.2 Thuật toán chia đôi

Thuật toán trị riêng tiếp theo của chúng ta, phương pháp *chia đôi*, là quan trọng thiết thực tuyệt với. Sau khi một ma trận đối xứng đã được chéo hóa 3 đường chéo, đó là bước tiếp theo tiêu chuẩn nếu ta không muốn tất cả các trị riêng nhưng chỉ một tập con của chúng. Ví dụ, chia đôi có thể tìm 10% lớn nhất của các trị riêng, hoặc 30 trị riêng nhỏ nhất, hoặc tất cả các trị riêng trong khoảng $[1, 2]$. Một lần nữa các trị riêng mong muốn được tìm thấy, các vector riêng tương ứng có thể đạt được bằng một bước của bước lặp nghịch đảo.

Điểm bắt đầu là cơ bản. Vì các trị riêng của một ma trận đối xứng thực là thực, ta có thể tìm thấy chúng bằng việc tìm dòng thực cho các nghiệm của đa thức $p(x) = det(A - xI)$. Sự khác nhau mà các nhận xét này đi đôi với ý tưởng của việc tìm các nghiệm từ *các hệ số* của đa thức. Bây giờ, ý tưởng là để tìm các nghiệm bằng ước lượng $p(x)$ tại các điểm thay đổi x, không tìm các hệ số của nó, và việc áp dụng quá trình chia đôi thông thường cho các hàm không tuyến tính. Ví dụ, điều này có thể được làm bằng khử Gauss với quay và thuật toán kết quả sẽ là ổn tình cao.

Phương pháp chia đôi cho các lũy thừa của nó và sự quyến rũ của nó là một vài tính chất thêm vào của các trị riêng và các định thức mà chúng không làm rõ ràng trực tiếp.

Cho một ma trận đối xứng $A \in \mathbb{R}^{m\times m}$, cho $A^{(1)}, \ldots, A^{(m)}$ ký hiệu các ma trận con vuông cơ bản (nghĩa là, trên bên trái) của nó có các số chiều $1, \ldots, m$. Nó có thể được cho thấy rằng các trị riêng của các ma trận này *đan xen*. Trước khi xác định tính chất này, đầu tiên ta hãy làm tăng thêm nó bằng việc giả sử rằng A là 3 đường chéo và *bất khả quy* trong ý nghĩa mà tất cả các phần tử ngoài đường chéo của nó là khác 0:

$$A = \begin{bmatrix} a_1 & b_1 & & & \\ b_1 & a_2 & b_2 & & \\ & b_2 & a_3 & \ddots & \\ & & \ddots & \ddots & b_{m-1} \\ & & & b_{m-1} & a_m \end{bmatrix}, \quad b_j \neq 0. \tag{5.7.5}$$

(Nếu có các số 0 nằm ngoài đường chéo chính thì bài toán trị riêng có thể được hạ cấp, như trong Thuật toán 5.6). Các trị riêng của $A^{(k)}$ là phân biệt và được ký hiệu bởi $\lambda_1^{(k)} < \lambda_2^{(k)} < \ldots < \lambda_k^{(k)}$. Tính chất chủ yếu mà nó làm chia đôi có hiệu lực là *đan xen một cách chặt chẽ* các trị riêng này, thỏa mãn bất phương trình

$$\lambda_j^{(k+1)} < \lambda_j^{(k)} < \lambda_{j+1}^{(k+1)} \tag{5.7.6}$$

với $k = 1, 2, \ldots, m-1$ và $j = 1, 2, \ldots, k-1$. Xử lý này được phác thảo trong Hình 5.2.

Tính chất đan xen làm nó có thể thực hiện được để đếm chính xác số các trị riêng của một ma trận trong một khoảng đặc biệt. Ví dụ, xét ma trận 3 đường chéo 4×4

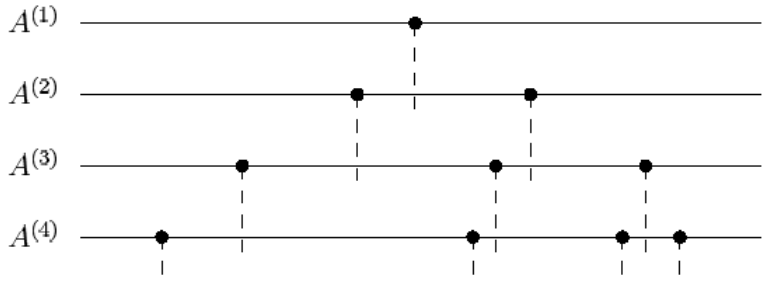
$$A = \begin{bmatrix} 1 & 1 & & \\ 1 & 0 & 1 & \\ & 1 & 2 & 1 \\ & & 1 & -1 \end{bmatrix}. \tag{5.7.7}$$

Từ các số

$$det(A^{(1)}) = 1, \quad det(A^{(2)}) = -1, \quad det(A^{(3)}) = -3, \quad det(A^{(4)} = 4,$$

ta biết rằng $A^{(1)}$ không có các trị riêng âm, $A^{(2)}$ có một trị riêng âm, $A^{(3)}$ có một trị riêng âm, và $A^{(4)}$ có 2 trị riêng âm. Tổng quát, cho các ma trận 3 đường chéo đối xứng bất kì $A \in \mathbb{R}^{m\times m}$, *số các trị riêng âm là bằng với số các thay đổi dấu trong chuỗi*

$$1, det(A^{(1)}), det(A^{(2)}), \ldots, det(A^{(m)}), \tag{5.7.8}$$



Hình 5.2: Minh họa của tính chất đan xen trị riêng chặt chẽ (5.7.6) cho các ma trận con cơ bản $\{A^{(j)}\}$ của một ma trận đối xứng thực 3 đường chéo bất khả qui A. Các trị riêng của $A^{(k)}$ đan xen nhau này của $A^{(k+1)}$. Thuật toán chia đôi tận dụng tính chất này.

mà nó được biết như là một *chuỗi Sturm*. (Quy định này làm việc ngay cả khi các định thức 0 được bắt gặp dọc theo cách này, nếu ta xác định một "sự thay đổi dấu" để muốn nói một sự chuyển tiếp từ + hoặc 0 tới - hoặc từ - hoặc 0 tới + nhưng không từ + hoặc - tới 0.) Do A dịch chuyển bằng một bội của đơn vị, ta có thể xác định số trị riêng trong khoảng bất kì $[a, b)$: nó là số trị riêng trong khoảng $(-\infty, b)$ trừ số trị riêng trong khoảng $(-\infty, a)$.

Cho một ma trận 3 đường chéo, các định thức của các ma trận $\{A^{(k)}\}$ có quan hệ bởi quan hệ hồi quy 3 số hạng. Việc mở rộng $det(A^{(k)})$ bằng các định thức con tương ứng với các phần tử b_{k-1} và a_k của nó trong dòng k cho, từ (5.7.5),

$$det(A^{(k)}) = a_k det(A^{(k-1)}) - b_{k-1}^2 det(A^{(k-2)}). \tag{5.7.9}$$

Việc đưa ra dịch chuyển bởi xI và viết $p^{(k)}(x) = det(A^{(k)} - xI)$, ta được

$$p^{(k)}(x) = (a_k - x)p^{(k-1)}(x) - b_{k-1}^2 p^{(k-2)}(x). \tag{5.7.10}$$

Nếu ta xác định $p^{(-1)}(x) = 0$ và $p^{(0)}(x) = 1$, thì hồi quy này là có hiệu quả cho mọi $k = 1, 2, \ldots, m$.

Do việc áp dụng 5.7.10 cho một chuỗi các giá trị của x và đếm các thay đổi dấu theo cách này, thuật toán chia đôi xác định các trị riêng trong các khoảng nhỏ tùy ý. Chi phí là $O(m)$ phép toán dấu chấm động cho mỗi ước lượng của chuỗi, do đó $O(m \log(\epsilon_{machine}))$ phép toán dấu chấm động tổng cộng để tìm một trị riêng tới sự đúng đắn tương đối $\epsilon_{machine}$. Nếu một số nhỏ của các trị riêng được yêu cầu, điều này là một sự cải thiện phân biệt trên $O(m^2)$ đếm phép toán cho thuật toán QR. Trong một máy tính có nhiều bộ xử lý, các trị riêng bội có thể được tìm thấy không phụ thuộc vào các bộ xử lý tách biệt.

5.7.3 Thuật toán chia để trị

Thuật toán chia để trị, được dựa vào một quá trình chia nhỏ đệ quy của một bài toán 3 đường chéo đối xứng thành các bài toán số chiều nhỏ hơn, đưa ra thuận lợi quan trọng nhất trong các thuật toán trị riêng ma trận từ những năm 1960. Đầu tiên được giới thiệu bởi Cuppen trong năm 1981, phương pháp này là nhanh hơn 2 lần thuật toán QR nếu các vector riêng cũng như các trị riêng được yêu cầu.

Cho $T \in \mathbb{R}^{m \times m}$ với $m \geqslant 2$ là ma trận đối xứng, 3 đường chéo và bất khả quy trong ý nghĩa mà chỉ có các số khác 0 ngoài đường chéo. (Mặc khác, bài toán có thể được giảm xuống.) Khi đó cho n nằm trong $1 \leqslant n \leqslant m$, T có thể được chia ra thành các ma trận con như sau: T_1 là một ma trận con cơ bản $n \times n$ ở trên bên trái của T, T_2 là ma trận con cơ bản $(m-n) \times (m-n)$ ở dưới bên phải, và $\beta = t_{n+1,n} = t_{n,n+1} \neq 0$. Sự khác nhau duy nhất giữa T_1 và T_2 là phần tử

$$T = \begin{bmatrix} T_1 & \beta \\ \beta & T_2 \end{bmatrix} = \begin{bmatrix} \hat{T}_1 & \\ & \hat{T}_2 \end{bmatrix} + \begin{bmatrix} \beta & \beta \\ \beta & \beta \end{bmatrix}.$$

ở dưới bên phải t_{nn} đã được thay thế bởi $t_{nn} - \beta$, và sự khác nhau duy nhất giữa T_2 và $\hat{T}_2$ là phần tử ở trên bên trái $t_{n+1,n+1}$ đã được thay thế bởi $t_{n+1,n+1} - \beta$. Các sự bổ sung này của 2 phần tử được đưa ra để làm ma trận bên phải nhất của Hình 5.7.3 có hạng 1.

Một ma trận 3 đường chéo có thể được viết như là tổng của một ma trận đường chéo khối 2×2 *với các khối 3 đường chéo và sự điều chỉnh hạng 1.*

Thuật toán chia để trị tiến hành như sau. Chọn ma trận T như trong hình trên với $n \approx m/2$. Giả sử rằng các trị riêng của $\hat{T}_1$ và $\hat{T}_2$ được biết. Vì ma trận điều chỉnh là hạng 1, không tuyến tính nhưng tính toán nhanh có thể sử dụng để thu được các trị riêng của $\hat{T}_1$ và $\hat{T}_2$ thành các trị riêng của T. Việc tìm các trị riêng của $\hat{T}_1$ và $\hat{T}_2$ bằng các sự chia nhỏ hơn nữa với các điều chỉnh hạng 1, và v.v. Trong phương pháp này một bài toán trị riêng $m \times m$ được giảm thành một tập các bài toán trị riêng 1×1 cùng với một sự tập hợp của các điều chỉnh hạng 1. (Đặc biệt, cho hiệu quả lớn nhất, nó là thói quen để chuyển thành thuật toán QR khi các ma trận con có số chiều đủ nhỏ hơn là thực hiện đệ quy tất cả cách này.)

Trong quá trình này có một điểm toán học quan trọng. Nếu các trị riêng của $\hat{T}_1$ và $\hat{T}_2$ được biết, các trị riêng của T có thể được tìm thấy như thế nào? Để trả lời điều này, giả sử rằng các sự chéo hóa

$$\hat{T}_1 = Q_1 D_1 Q_1^T, \quad \hat{T}_2 = Q_2 D_2 Q_2^T$$

đã được tính. Khi đó, ta có

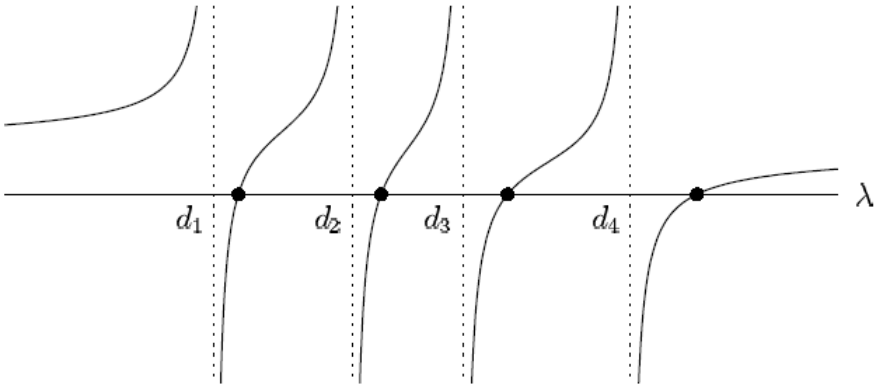
$$T = \begin{bmatrix} Q_1 & \\ & Q_2 \end{bmatrix} \left(\begin{bmatrix} D_1 & \\ & D_2 \end{bmatrix} + \beta z z^T \right) \begin{bmatrix} Q_1^T & \\ & Q_2^T \end{bmatrix} \tag{5.7.11}$$

với $z^T = (q_1^T, q_2^T)$, q_1^T là dòng cuối cùng của Q_1 và q_2^T là dòng đầu tiên của Q_2. Vì phương trình này là một biến đổi tương đương nên ta đã giảm bài toán toán học này thành bài toán tìm các trị riêng của ma trận đường chéo cộng với sự điều chỉnh hạng 1.

Giả sử ta mong muốn tìm các trị riêng của $D + ww^T$, với $D \in \mathbb{R}^{m \times m}$ là một ma trận đường chéo với các phần tử đường chéo $\{d_j\}$ phân biệt và $w \in \mathbb{R}^m$ là một vector. (Sự lựa chọn của một dấu + tương ứng với $\beta > 0$ ở trên; với $\beta < 0$ ta sẽ xét $D - ww^T$.) Ta có thể giả sử $w_j \neq 0$ với mọi j, cho trường hợp khác, bài toán là rút gọn được. Khi đó các trị riêng của $D + ww^T$ là các nghiệm của hàm hữu tỷ

$$f(\lambda) = 1 + \sum_{j=1}^{m} \frac{w_j^2}{d_j - \lambda}, \tag{5.7.12}$$

được minh họa như trong Hình 5.3. Khẳng định này có thể được chứng minh bằng việc kiểm tra điều đó nếu $(D + ww^T)q = \lambda q$ với $q \neq 0$ bất kì, khi đó $(D - \lambda I)q + w(w^T q) = 0$, kéo theo $q + (D - \lambda I)^{-1} w(w^T q) = 0$, đó là, $w^T q + w^T (D - \lambda I)^{-1} w(w^T q) = 0$. Cái đó chẳng khác gì là $f(\lambda)(w^T q) = 0$, trong đó $w^T q$ phải khác 0, cho trường hợp khác q sẽ là một vector riêng của D, do đó khác không chỉ trong 1 vị trí, kéo theo $w^T q \neq 0$. Ta kết luận rằng nếu q là một vector riêng của $D + ww^T$ với trị riêng λ, khi đó $f(\lambda)$ phải bằng 0, và ngược lại theo sau bởi dạng của $f(\lambda)$ đảm bảo rằng nó có chính xác m số không. Phương trình $f(\lambda) = 0$ được biết như là *phương trình secular*.



Hình 5.3: Đồ thị của hàm $f(\lambda)$ của (5.7.12) cho bài toán 4 chiều. Các cực của $f(\lambda)$ là các trị riêng $\{d_j\}$ của D, và các nghiệm của $f(\lambda)$ (các dấu chấm khối) là các trị riêng của $D + ww^T$. Sự xác định nhanh chóng của các nghiệm này là cơ sở của mỗi bước đệ qui của thuật toán chia để trị.

Tại mỗi bước đệ quy của thuật toán chia để trị, các nghiệm của (5.7.12) được xác định bởi một quá trình lặp nhanh có liên quan với phương pháp của Newton. Chỉ $O(1)$ bước lặp được yêu cầu cho mỗi nghiệm (hoặc $O(\log(|\log(\epsilon_{machine})|))$ bước lặp nếu $\epsilon_{machine}$ được xem như là một biến số), đếm phép toán $O(m)$ phép toán dấu chấm động trên nghiệm cho một ma trận $m \times m$, hoặc $O(m^2)$ phép toán dấu chấm động tất cả cùng nhau. Nếu ta cho rằng một sự đệ qui mà trong đó một ma trận m chiều được chọn một cách chính xác trong một nửa tại mỗi bước thì tổng số đếm phép toán cho việc tìm các trị riêng của một ma trận 3 đường chéo bằng thuật toán chia để trị trở thành

$$O\left(m^2 + 2\left(\frac{m}{2}\right)^2 + 4\left(\frac{m}{4}\right)^2 + 8\left(\frac{m}{8}\right)^2 + \ldots + m\left(\frac{m}{m}\right)^2\right), \tag{5.7.13}$$

một chuỗi mà nó hội tụ về $O(m^2)$ (không $O(m^2 \log m)$) vì các bình phương trong các mẫu số. Do đó đếm số phép toán sẽ xuất hiện để là cùng bậc $O(m^2)$ như cho thuật toán QR.

Hơn nữa, nó không rõ ràng vì sao thuật toán chia để trị là thuận lợi. Vì sự giảm một ma trận đầy đủ thành dạng 3 đường chéo ("Giai đoạn 1" trong thuật ngữ của mục 5.2) yêu cầu $4m^3/3$ phép toán dấu chấm động (5.3.2), nó sẽ dường như rằng sự cải thiện bất kì trong $O(m^2)$ đếm phép toán cho sự chéo hóa mà ma trận 3 đường chéo ("Giai đoạn 2") là hầu như không quan trọng. Tuy nhiên, các lợi nhuận thay đổi nếu ta tính toán các vector riêng tốt như các trị riêng. Bây giờ, Giai đoạn 1 yêu cầu $8m^3/3$ phép toán dấu chấm động nhưng Giai đoạn 2 cũng yêu cầu $O(m^3)$ phép toán dấu chấm động - cho thuật toán QR, $\approx 6m^3$. Thuật toán chia để trị giảm con số này xuống, bởi vì các bước lặp không tuyến tính của nó bao gồm hàm vô hướng (5.7.12), không bao gồm các ma trận trực giao Q_j, trong khi thuật toán QR phải thao tác các ma trận Q_j tại mỗi bước lặp.

$O(m^3)$ phần của tính toán thuật toán chia để trị là phép nhân với Q_j và Q_j^T trong (5.7.11). Tổng số đếm phép toán, được lấy tổng trên tất cả các bước của sự đệ qui, là $4m^3/3$, một sự thực thi tuyệt vời trên $\approx 6m^3$ phép toán dấu chấm động. Cộng thêm $8m^3/3$ phép toán dấu chấm động cho Giai đoạn 1 cho một sự cải thiện từ $\approx 9m^3$ thành $4m^3$.

Trên thực tế, thuật toán chia để trị thường là tốt hơn điều này, vì một lý do không cơ bản. Cho hầu hết các ma trận A, nhiều vector z và các ma trận Q_j xuất hiện trong (5.7.11) đưa ra để là thưa thớt về số lượng trong ý nghĩa mà nhiều phần tử của chúng có các tính chất ít có liên quan hơn sự đúng đắn của máy. Sự thưa thớt này cho phép một quá trình của *sự giảm*

xuống số lượng, do đó các bài toán trị riêng 3 đường chéo liên tiếp được giảm thành các bài toán không liên kết có các số chiều nhỏ hơn. Trong các trường hợp điển hình điều này giảm đếm phép toán của Giai đoạn 2 thành một bậc nhỏ hơn m^3 phép toán dấu chấm động, giảm đếm phép toán của Giai đoạn 1 và 2 được kết hợp lại thành $8m^3/3$. Chỉ cho có các trị riêng, (5.7.13) trở thành một đánh giá quá cao và đếm phép toán của Giai đoạn 2 được giảm xuống thành một bậc thấp hơn m^2 phép toán dấu chấm động.

Ta đã nói như thể có một thuật toán chia để trị đơn, nhưng trong thực tế, có nhiều biến thể. Các cập nhật hạng 1 được làm phức tạp hơn thường được sử dụng cho các lý do ổn định, và các cập nhật hạng 2 cũng thỉnh thoảng được sử dụng. Các phương pháp khác nhau được dùng cho việc tìm các nghiệm của $f(\lambda)$, và cho m lớn, cách nhanh nhất để thực thi các phép nhân với Q_j là thông qua các sự mở rộng đa cực hơn là thuật toán rõ ràng. Một sự thực thi chất lượng cao của một thuật toán chia để trị có thể được tìm thấy trong thư viện Lalack.

5.8 Tính SVD

5.8.1 SVD của A và các trị riêng của A^*A

Như được phát biểu trong Định lý 1.5.5, SVD của ma trận A có cấp $m \times n$ ($m \geqslant n$), $A = U\sum V^*$, có quan hệ với phân tích trị riêng của ma trận A^*A,

$$A^*A = V\overset{*}{\sum}\sum V^*. \tag{5.8.1}$$

Do đó, ta có thể tính SVD của A như sau:

1. Thiết lập A^*A.
2. Tính phân tích trị riêng $A^*A = V\Lambda V^*$;
3. Cho $\sum$ là căn bậc hai trên đường chéo không âm $m \times n$ của Λ;
4. Giải hệ $U\sum = AV$ cho ma trận unita U (ví dụ, thông qua phân tích QR).

Thuật toán này được sử dụng thường xuyên. Ma trận A^*A được biết như là *ma trận hiệp phương sai* của A, và nó có các sự giải thích quen thuộc trong thống kê và các lĩnh vực khác. Thuật toán là không ổn định, tuy nhiên, bởi vì nó giảm bài toán SVD thành bài toán trị riêng có thể là dễ bị hỏng nhiều hơn tới các các nhiễu.

Sự khó khăn có thể được giải thích như sau. Ta đã thấy điều đó khi một ma trận hermit A^*A được làm nhiễu bởi δB, trị tuyệt đối thay đổi trong mỗi trị riêng được chặn bởi chuẩn 2 của nhiễu, $|\lambda_k(A^*A+\delta B)-\lambda_k(A^*A)| \leqslant \|\delta B\|_2$. Như được suy ra bởi phương trình (5.8.3) ở bên dưới, một chặn tương tự đúng cho các giá trị suy biến của chính A, $|\sigma_k(A+\delta A)-\sigma_k(A)| \leqslant \|\delta A\|_2$. Do đó một thuật toán ổn định ngược cho việc tính toán các giá trị suy biến sẽ thu được $\tilde{\sigma}_k$ thỏa mãn

$$\tilde{\sigma}_k = \sigma_k(A+\delta A), \quad \frac{\|\delta A\|}{\|A\|} = O(\epsilon_{machine}), \tag{5.8.2}$$

mà nó sẽ kéo theo

$$|\tilde{\sigma}_k - \sigma_k| = O(\epsilon_{machine}\|A\|).$$

Bây giờ quan sát cái gì xảy ra nếu ta tiếp tục bằng việc tính $\lambda_k(A^*A)$. Nếu $\lambda_k(A^*A)$ được tính một cách ổn định, ta phải mong đợi các bậc

$$|\tilde{\lambda}_k - \lambda_k| = O(\epsilon_{machine}\|A^*A\|) = O(\epsilon_{machine}\|A\|^2).$$

Khai căn bậc hai để thu được σ_k, ta được

$$|\tilde{\sigma}_k - \sigma_k| = O(|\tilde{\lambda_k} - \lambda_k|/\sqrt{\lambda_k}) = O(\epsilon_{machine}\|A\|^2/\sigma_k).$$

Điều là tệ hơn kết quả trước bởi một thừa số $O(\|A\|/\sigma_k)$. Điều này không là vấn đề cho các giá trị suy biến ưu thế của A, với $\sigma_k \approx \|A\|$, nhưng nó là một bài toán lớn cho các giá trị suy biến bất kì với $\sigma_k \ll \|A\|$. Cho giá trị suy biến nhỏ nhất σ_n, ta phải mong đợi một hao hụt của sự đúng đắng bậc $\kappa(A)$ - một "bình phương của số điều kiện," giống như trong sử dụng các phương trình chính tắc cho các bài toán bình phương nhỏ nhất nào đó.

5.8.2 Một sự giảm khác nhau thành một bài toán trị riêng

Giả sử A là ma trận vuông, với $m = n$; điều này không hạn chế cần thiết, vì ta sẽ thấy rằng các bài toán giá trị suy biến hình chữ nhật có thể được giảm xuống thành các bài toán giá trị suy biến vuông. Xét ma trận hermit $2m \times 2m$

$$H = \begin{bmatrix} 0 & A^* \\ A & 0 \end{bmatrix} \tag{5.8.3}$$

Vì $A = U\sum V^*$ kéo theo $AV = U\sum$ và $A^*U = V\sum^* = V\sum$, ta có phương trình 2×2 khối

$$\begin{bmatrix} 0 & A^* \\ A & 0 \end{bmatrix} \begin{bmatrix} V & V \\ U & -U \end{bmatrix} = \begin{bmatrix} V & V \\ U & -U \end{bmatrix} \begin{bmatrix} \sum & 0 \\ 0 & -\sum \end{bmatrix}, \tag{5.8.4}$$

mà nó có nghĩa là một phân tích trị riêng của H. Do đó ta thấy rằng các giá trị suy biến của A là các giá trị tuyệt đối của các trị riêng của H, và các vector suy biến của A có thể được rút ra từ các vector riêng của H.

Do đó ta có thể thu được SVD của một ma trận vuông bằng viêc tạo thành ma trận H và việc tính phân tích trị riêng của nó. Ngược lại để sử dụng AA^* hoặc A^*A, xấp xỉ này là ổn định. Các thuật toán tiêu chuẩn cho SVD được dựa vào ý tưởng này, không ma trận nào có số chiều lớn như $m + n$ được hình thành một cách rõ ràng. Và bước quan trọng để làm quá trình nhanh là giảm unita ban đầu thành dạng song đường chéo.

5.8.3 Hai quá trình

Ta thấy rằng các bài toán trị riêng hermit thường được giải bằng tính toán 2 giai đoạn: đầu tiên giảm ma trận thành dạng 3 đường chéo, khi đó chéo hóa ma trận 3 đường chéo. Vì việc làm của Golub, Kahan, và những người khác trong những năm 1960, một xấp xỉ 2 giai đoạn tương tự đã là tiêu chuẩn của SVD. Ma trận được làm thành dạng song đường chéo, và khi đó ma trận song đường chéo được chéo hóa:

$$\underset{A}{\begin{bmatrix} \times & \times & \times & \times \\ \times & \times & \times & \times \\ \times & \times & \times & \times \\ \times & \times & \times & \times \\ \times & \times & \times & \times \\ \times & \times & \times & \times \end{bmatrix}} \xrightarrow{\text{Giai đoạn 1}} \underset{B}{\begin{bmatrix} \times & \times & & \\ & \times & \times & \\ & & \times & \times \\ & & & \times \\ & & & \\ & & & \end{bmatrix}} \xrightarrow{\text{Giai đoạn 2}} \underset{\sum}{\begin{bmatrix} \times & & & \\ & \times & & \\ & & \times & \\ & & & \times \\ & & & \\ & & & \end{bmatrix}}.$$

Giai đoạn 1 bao gồm một số hữu hạn của các phép toán, $O(mn^2)$ phép toán dấu chấm động. Giai đoạn 2 đặc biệt yêu cầu một số vô hạn các phép toán, nhưng các thuật toán tiêu chuẩn

hội tụ siêu tuyến tính, và do đó chỉ $O(n \log(|\log(\epsilon_{machine})|))$ bước lặp được yêu cầu cho sự hội tụ tới bậc $\epsilon_{machine}$. Trong thực hành, ta nghĩ $\epsilon_{machine}$ như là một hằng số và nói rằng sự hội tụ được đạt được trong $O(n)$ bước lặp. Bởi vì ma trận được tính toán là song đường chéo, mỗi bước của bước lặp này yêu cầu chỉ $O(n)$ phép toán dấu chấm động. Do đó Giai đoạn 2 yêu cầu $O(n^2)$ phép toán dấu chấm động (giả sử các giá trị suy biến không phải các vector được yêu cầu). Do đó, mặc dù Giai đoạn 1 là hữu hạn và Giai đoạn 2 đặc biệt là vô hạn, trong thực hành sau cùng là ít tốn kém nhất, ngay khi ta tìm thấy cho bài toán trị riêng đối xứng.

5.8.4 Song chéo hóa Golub-Kahan

Trong Giai đoạn 1 của tính toán SVD, ta đưa A thành dạng song đường chéo bằng việc áp dụng các phép toán unita phân biệt trong vế trái và vế phải. Chú ý làm thế nào điều này suy ra từ sự tính toán của các trị riêng, nơi mà các phép toán unita giống nhau phải được áp dụng trong cả 2 vế để mỗi bước là một biến đổi tương đương. Trong trường hợp đó, nó chỉ có thể thực hiện được để đưa ra các số 0 bên dưới đường chéo phụ đầu tiên. Ở đây, ta sẽ tam giác hóa một cách đầy đủ và cũng đưa ra các số 0 ở trên siêu đường chéo đầu tiên.

Phương pháp đơn giản cho việc hoàn thành điều này, *song chéo hóa Golub-Kahan*, tiếp tục như sau. Các phản xạ Householder được áp dụng thay phiên nhau trong vế trái và vế phải. Mỗi phản xạ trái đưa ra một cột 0 bên dưới đường chéo. Phản xạ phải theo sau đó đưa ra một dòng 0 vào bên phải của siêu đường chéo đầu tiên, bỏ đi không hư hại các số 0 chỉ được đưa vào cột. Ví dụ, cho ma trận 6×4, 2 cặp đầu tiên của các phản xạ giống điều này:

$$\begin{bmatrix} \times & \times & \times & \times \\ \times & \times & \times & \times \\ \times & \times & \times & \times \\ \times & \times & \times & \times \\ \times & \times & \times & \times \\ \times & \times & \times & \times \end{bmatrix} \xrightarrow{U_1^*\cdot} \begin{bmatrix} \times & \times & \times & \times \\ 0 & \times & \times & \times \\ 0 & \times & \times & \times \\ 0 & \times & \times & \times \\ 0 & \times & \times & \times \\ 0 & \times & \times & \times \end{bmatrix} \xrightarrow{\cdot V_1} \begin{bmatrix} \times & \times & 0 & 0 \\ & \times & \times & \times \\ & \times & \times & \times \\ & \times & \times & \times \\ & \times & \times & \times \\ & \times & \times & \times \end{bmatrix}$$

$$A \qquad\qquad U_1^* A \qquad\qquad U_1^* A V_1$$

$$\xrightarrow{U_2^*\cdot} \begin{bmatrix} \times & \times & & & \\ & \times & \times & \times \\ & 0 & \times & \times \\ & 0 & \times & \times \\ & 0 & \times & \times \\ & 0 & \times & \times \end{bmatrix} \xrightarrow{\cdot V_2} \begin{bmatrix} \times & \times & & & \\ & \times & \times & 0 \\ & & \times & \times \\ & & \times & \times \\ & & \times & \times \\ & & \times & \times \end{bmatrix}.$$

$$U_2^* U_1^* A V_1 \qquad\qquad U_2^* U_1^* A V_1 V_2$$

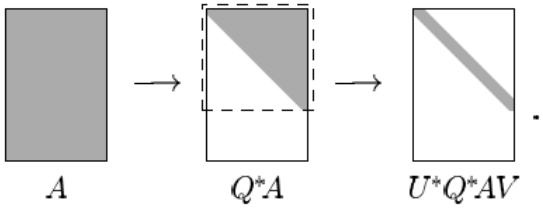
Phép nhân trái với U_1^* thay đổi các dòng 1 tới m, đưa ra các số 0 trong cột 1 bên dưới đường chéo. Phép nhân phải với V_1 thay đổi các cột 2 tới m, đưa các số 0 vào dòng 1 không triệt tiêu các số 0 trong cột 1. Quá trình tiếp tục với các phép toán trong các dòng 2 tới m, khi đó các cột 3 tới n, và v.v.

Kết thúc quá trình này, n phản xạ đã được áp dụng vào vế trái và $n-2$ phản xạ vào vế phải. Kiểu của các phép toán dấu chấm động tương tự hai phân tích QR và Householder được đan xen với nhau, một cái tính toán trên ma trận A cấp $m \times n$, một tính toán khác trên ma trận A^* cấp $n \times m$. Đếm phép toán tổng cộng là 2 lần của thuật toán QR (2.5.10), nghĩa là,

$$\text{Song chéo hóa Golub - Kahan: } \sim 4mn^2 - \frac{4}{3}n^3 \text{ phép toán dấu chấm động.}$$

5.8.5 Các phương pháp nhanh hơn cho Giai đoạn 1

Cho $m \gg n$, đếm phép toán này là không nhất thiết lớn. Một phân tích QR đơn sẽ đưa các số 0 khắp nơi bên dưới đường chéo, và cho $m \gg n$, những cái này là đại đa số các số 0 được yêu cầu. Đếm phép toán cho phương pháp Golbu-Kahan là 2 lần. Sự quan sát này đề nghị một phương pháp thay phiên nhau cho song chéo hóa với $m \gg n$, được đề nghị đầu tiên bởi Lawson và Hanson và sau đó được phát triển bởi Chan. Ý tưởng, *song chéo hóa LHC*, được minh họa như sau: Ta bắt đầu bằng việc tính phân tích QR $A = QR$. Khi đó ta tính song chéo

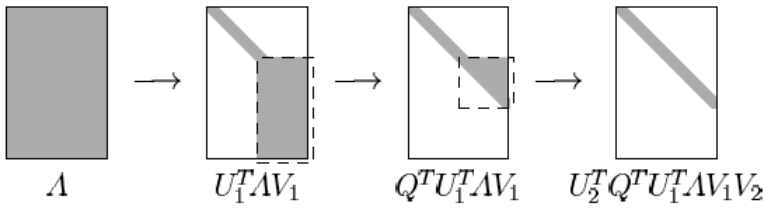


Hình 5.4: Song chéo hóa Lawson-Hanson-Chan

hóa Golub-Kahan $B = U^*RV$ cho R. Phân tích QR yêu cầu $2mn^2 - \frac{2}{3}n^3$ phép toán dấu chấm động (2.5.10), và thủ tục Golub-Kahan, mà nó chỉ phải tính ma trận con $n \times n$ trên, yêu cầu $\frac{8}{3}n^3$ phép toán dấu chấm động. Đếm số phép toán tổng cộng là

$$\text{Song chéo hóa LHC: } \sim 2mn^2 + 2n^3 \text{ phép toán dấu chấm động.} \tag{5.8.5}$$

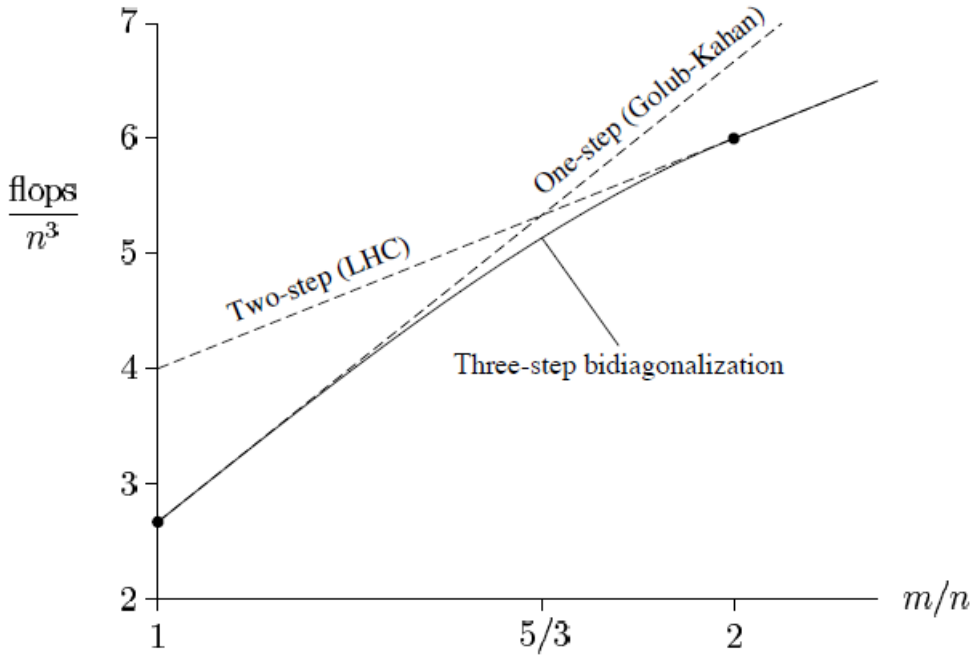
Điều này là ít tốn kém hơn song đường chéo Golub-Kahan cho $m > \frac{5}{3}n$. Thủ tục LHC khởi tạo các số 0 và khi đó triệt tiêu chúng lần nữa (trong tam giác dưới hơn của ma trận vuông $n \times n$ trên của A), nhưng ở đây là tăng thêm cuối cùng.



Hình 5.5: Song chéo hóa 3 bước

Thủ tục LHC chỉ thuận lợi khi $m > \frac{5}{3}n$, nhưng ý tưởng có thể được tổng quát hóa để thực hiện một việc lưu trữ cho $m > n$ bất kì. Thủ thuật là để áp dụng phân tích QR không bắt đầu của sự tính toán, nhưng tại điểm phù hợp ở giữa. Điều này là thuận lợi bởi vì trong xử lý Golub-Kahan, một ma trận với $m > n$ trở thành rất gầy như các xử lý song chéo hóa. Nếu tỷ lệ phương diện ban đầu là, cụ thể, $m/n = 3/2$, nó sẽ tăng đều đều thành 5/3 và 2 và ở phía bên kia. Sau bước k, tỷ lệ phương diện của ma trận còn lại là $(m-k)/(n-k)$, và khi con số này đủ lớn, thực hiện một phân tích QR để giảm bài toán thành một ma trận vuông.

Khi nào phân tích QR sẽ được thực thi? Nếu ta hướng vào cực tiểu hóa đếm phép toán, câu trả lời là đơn giản: khi tỷ lệ phương diện đạt được $(m-k)/(n-k) = 2$. Sự lựa chọn này



Hình 5.6: Đếm số phép toán cho 3 thuật toán song chéo hóa áp dụng cho các ma trận $m \times n$, từ (5.8.4), (5.8.5) và (5.8.6). Song chéo hóa 3 bước cung cấp một interpolant trơn dễ chịu giữa 2 phương pháp khác, thông qua sự cải thiện là hầu như không lớn.

đưa ra công thức

$$\text{Song chéo hóa 3 bước: } \sim 4mn^2 - \frac{4}{3}n^3 - \frac{2}{3}(m-n)^3 \text{ phép toán dấu chấm động.} \qquad (5.8.6)$$

một sự cải thiện vừa phải trên 2 phương pháp khác cho $n < m < 2n$.

Đếm phép toán cho 3 phương pháp được vẽ đồ thị như một hàm của m/n trong Hình 5.6. Nó phải được thừa nhận rằng sự cải thiện đạt được bởi phương pháp 3 bước là đủ nhỏ mà trong thực hành, ngoài ra các vấn đề khác đếm phép toán có thể xác định mà phương pháp là tốt nhất trong một máy thực sự.

5.8.6 Giai đoạn 2

Trong Giai đoạn 2 của tính toán SVD, SVD của ma trận song đường chéo B được xác định. Từ những năm 1960 tới những năm 1990, thuật toán tiêu chuẩn cho điều này là một biến thể của thuật toán QR. Gần đây hơn, các thuật toán chia để trị cũng đã trở nên cạnh tranh nhau, và trong tương lai, chúng có thể trở thành tiêu chuẩn. Ta sẽ không cho chi tiết.

Bài tập

1. Cho $A \in \mathbb{C}^{m\times m}$ tùy ý và chuẩn $\|\cdot\|$, sử dụng Định lý 5.1.9 để chứng minh:

   a) $lim_{n\to\infty}\|A^n\| = 0 \iff \rho(A) < 1$, với ρ là bán kính phổ.

   b) $lim_{t\to\infty}\|e^{tA}\| = 0 \iff \alpha(A) < 0$, với α là hoành độ phổ.

2. Cho $A \in \mathbb{C}^{m\times m}$ không nhất thiết hermit. Chứng minh rằng một số $x \in \mathbb{C}$ là tỷ số Rayleigh của A nếu và chỉ nếu nó là một phần tử đường chéo của Q^*AQ, với Q là ma trận unita bất kỳ.

3. Một ma trận đối xứng thực A có một trị riêng là 1 bội 8, trong khi tất cả các trị riêng còn lại $\leqslant 0.1$ trong giá trị tuyệt đối. Miêu tả một thuật toán cho việc tìm một cơ sở trực giao của không gian riêng 8 chiều tương ứng với trị riêng trội nhất.

4. Chứng minh rằng nếu phần tử ngoài đường chéo lớn nhất bị triệt tiêu tại mỗi bước của thuật toán Jacobi thì tổng bình phương của các phần tử ngoài đường chéo giảm xuống ít nhất một thừa số $1 - \dfrac{2}{m^2 - m}$ tại mỗi bước.

5. Cài đặt Thuật toán 5.1 trong Matlab.

6. Cài đặt các Thuật toán 5.2, 5.3 và 5.4 trong Matlab.

7. Cài đặt các Thuật toán 5.5, 5.6 và 5.7 trong Matlab.

8. Viết một chương trình tìm các trị riêng của một ma trận đối xứng thực cấp $m \times m$ bằng Thuật toán Jacobi và vẽ đồ thị tổng của các bình phương của các phần tử ngoài đường chéo. Áp dụng chương trình cho các ma trận ngẫu nhiên với các chiều 20, 40 và 80.

9. Cho
$$A = \begin{bmatrix} 1 & 1 & 0 & 0 \\ 1 & 1 & 1 & 0 \\ 0 & 1 & 2 & 1 \\ 0 & 0 & 1 & 3 \end{bmatrix}$$
có bao nhiêu trị riêng của A trong khoảng [1, 2]?

10. Cho A là một ma trận tam giác trên với 0.1 trên đường chéo và 1 ở mọi nơi trên đường chéo. Viết một chương trình tính giá trị suy biến nhỏ nhất của A trong 2 cách: bằng việc gọi một phần mềm SDV tiêu chuẩn, và bằng việc thiết lập A^*A và tính căn bậc hai của trị riêng nhỏ nhất của nó. Chạy chương trình với $1 \leqslant m \leqslant 30$ và vẽ đồ thị của các kết quả.

Tài liệu tham khảo

[1] Lloyd N. Trefethen, and David Bau, III (1997), Numerical Linear Algebra, *Society for Industrial and Applied Mathematics*.

[2] Gene H. Golub, and Charles F. Van Loan (2013), Matrix Computations, *The Johns Hopkins University*.