FDU 数值算法 5. 非对称特征值问题的解法

本文根据邵老师授课内容整理而成，并参考了以下教材:

- 数值线性代数 (第二版, 徐树方, 高立, 张平文) 第 6 章

欢迎批评指正!

5.1 幂法

考虑 $A \in \mathbb{C}^{n\times n}$ 的特征方程 $Ax = x\lambda$ $(x \neq 0_n)$ 的求解问题:

- ① 若 $\lambda \in \mathbb{C}$ 已知而 $x \neq 0_n \in \mathbb{C}^n$ 未知，则问题转化为求解线性方程组 $(A - \lambda I)x = 0_n$
  我们一般默认 $\lambda \in \mathbb{C}$ 不是重特征值.
- ② 若 $x \neq 0_n \in \mathbb{C}^n$ 已知而 $\lambda \in \mathbb{C}$ 未知，则问题转化为求解可能矛盾的线性方程组 $x\lambda = Ax$
  我们将其视为最小二乘问题，写出其法方程: $x^{\mathrm{H}}x\lambda = x^{\mathrm{H}}(Ax)$
  解得 $\lambda = \frac{x^{\mathrm{H}}Ax}{x^{\mathrm{H}}x}$，称为 **Raylaigh 商**，这个解可使残量 $\|Ax - x\lambda\|_2^2$ 最小化.
- ③ 本文的重点便是解决 $\lambda \in \mathbb{C}$ 和 $x \neq 0_n \in \mathbb{C}^n$ 都是未知的情况.

幂法用于计算方阵的模最大特征值和对应的特征向量.

为说明幂法的基本思想，首先假设 $A \in \mathbb{C}^{n\times n}$ 是可对角化的，即 A 具有分解 $A = X\Lambda X^{-1}$
其中 $\Lambda = \mathrm{diag}(\lambda_1, \dots, \lambda_n)$，$X = [x_1, \dots, x_n] \in \mathbb{C}^{n\times n}$ 非奇异
假定 $|\lambda_1| > |\lambda_2| \geq \cdots \geq |\lambda_n|$ (λ_1 是模最大特征值)

现取任一向量 $u^{(0)} \in \mathbb{C}^n$
由于 $\mathrm{span}\{x^{(1)}, \dots, x^{(n)}\} = \mathbb{C}^n$,
故存在 $\alpha_j \in \mathbb{C}$ $(j = 1, \dots, n)$ 使得 $u^{(0)} = \alpha_1 x_1 + \cdots + \alpha_n x_n$ (假设 $\alpha_1 \neq 0$)
于是我们有:

$$
\begin{aligned}
A^k u^{(0)} &= A^k \sum_{j=1}^{n} \alpha_j x_j \\
&= \sum_{j=1}^{n} \alpha_j \cdot A^k x_j \\
&= \sum_{j=1}^{n} \alpha_j \cdot \lambda_j^k x_j \qquad\qquad \Rightarrow \lim_{k\to\infty} \frac{A^k u^{(0)}}{\lambda_1^k} = \alpha_1 x_1 \\
&= \lambda_1^k \left[\alpha_1 x_1 + \sum_{j=2}^{n} \alpha_j \left(\frac{\lambda_j}{\lambda_1} \right)^k x_j \right]
\end{aligned}
$$

这表明: 当 $\alpha_1 \neq 0$ 且 k 充分大时，向量 $u^{(k)} = \frac{A^k u^{(0)}}{\lambda_1^k}$ 是 A 关于模最大特征值 λ_1 的近似特征向量.

即使 A 不可对角化，我们依然可以处理.
假设模最大特征值 λ_1 是单重的:

$$
\begin{gathered}
A = P \begin{bmatrix} \lambda_1 & \\ & A_2 \end{bmatrix} P^{-1} \\
|\lambda_1| > |\lambda_2| \geq \cdots \geq |\lambda_n|
\end{gathered}
$$

则我们有:

$$
\begin{aligned}
A^k &= P \begin{bmatrix} \lambda_1^k & \\ & A_2^k \end{bmatrix} P^{-1} \\
&= \lambda_1^k P \begin{bmatrix} 1 & \\ & (\frac{A_2}{\lambda_1})^k \end{bmatrix} P^{-1} \\
&\approx \lambda_1^k P \begin{bmatrix} 1 & \\ & 0_{(n-1)\times(n-1)} \end{bmatrix} P^{-1} \quad (\text{when } k \text{ is very large}) \\
&= \lambda_1^k p_1 q_1^{\mathrm{T}}
\end{aligned}
$$

其中 p_1 是 P 的第一列，即 λ_1 的右特征向量; q_1^{T} 是 P^{-1} 的第一行，即 λ_1 的左特征向量.
因此对于向量 $u^{(0)} \in \mathbb{C}^n$，当 k 很大时，$A^k u^{(0)} \approx \lambda_1^k (q_1^{\mathrm{T}} u^{(0)}) p_1$，近似与特征向量 p_1 同向.

然而在实际计算中，这是行不通的:

- 我们事先不知道 A 的模最大特征值 λ_1
- 直接计算 A^k 的工作量太大

注意到 $\frac{A^k u^{(0)}}{\lambda_1^k}$ 中 λ_1^k 仅改变向量 $A^k u^{(0)}$ 的模长，而不影响其方向.
因此我们不必用 λ_1^k 来控制 $A^k u^{(0)}$ 的模长，而可用其他方便得到的常数来控制 (为防止溢出，这种控制是必要的)
另外 A^k 的计算可以迭代地进行.
基于上述观察，我们可设计如下的迭代格式，称为**幂法** (power interation):
(Homework 07 Problem 03)

Given initial vector $u^{(0)}$ such that $\|u^{(0)}\|_\infty = 1$

$$y^{(k)} = Au^{(k-1)}$$

$$\rho_k = \|y^{(k)}\|_\infty = y^{(k)}_{j_{\max}} \text{ where } j_{\max} = \arg\max_{1\le j\le n} |y^{(k)}_j|$$

$$u^{(k)} = \frac{1}{\rho_k} y^{(k)}$$

$$\lambda^{(k)} = \frac{(u^{(k)})^{\mathrm{T}} A u^{(k)}}{(u^{(k)})^{\mathrm{T}}(u^{(k)})} \text{ (Rayleigh Quotient)}$$

$$r^{(k)} = Au^{(k)} - u^{(k)}\lambda^{(k)} \text{ (Residual)}$$

Converge if $\|r^{(k)}\|_\infty \le \tau \cdot (\|A\|_\infty \|u^{(k)}\|_\infty + \|u^{(k)}\|_\infty |\lambda^{(k)}|)$ where $\tau > 0$ is predefined tolerance

其 Matlab 代码为:

```
function [lambda, u] = Power_Iteration(A, max_iter, tolerance)
    % Computes the dominant eigenvalue and eigenvector of matrix A using the power iteration method.
    %
    % Inputs:
    %   A - The input matrix (n x n)
    %   max_iter - Maximum number of iterations
    %   tolerance - Convergence tolerance
    %
    % Outputs:
    %   lambda - Estimated dominant eigenvalue
    %   u - Corresponding dominant eigenvector

    % Initialize a random vector u
    n = size(A, 1); % Get the dimension of the matrix A
    u = rand(n, 1); % Random initial vector
    u = u / norm(u, inf); % Normalize u such that ||u^(0)||_∞ = 1

    % Power iteration loop
    for k = 1:max_iter
        % Compute y^(k)
        y = A * u;

        % Compute rho_k (the infinity norm of y)
        rho = norm(y, "inf");

        % Update u^(k)
        u = y / rho; % Normalize y to update u

        % Compute the Rayleigh quotient (to estimate the eigenvalue)
        lambda = (u' * A * u) / (u' * u);

        % Compute the residual
        r = A * u - u * lambda;

        % Check for convergence
        if norm(r, inf) <= tolerance * (norm(A, inf) * norm(u, inf) + norm(u, inf) * abs(lambda))
            fprintf('Power Iteration converged in %d iterations.\n', k);
            break;
        end
    end
end
```

(幂法的收敛性, 数值线性代数, 定理 6.2.1)
设 $A \in \mathbb{C}^{n\times n}$ 有 p 个互不相同的特征值满足 $|\lambda_1| > |\lambda_2| \ge \cdots \ge |\lambda_p|$
并且模最大特征值 λ_1 是半简单的 (即 λ_1 的几何重数等于其代数重数)
若初始向量 $u^{(0)}$ 在 λ_1 的特征子空间上的投影不为零,
则幂法产生的向量序列 $\{u^{(k)}\}$ 收敛到 λ_1 的一个特征向量 x_1, 且数值序列 $\{\rho_k\}$ 收敛到 λ_1

- **证明:**
  设 A 的 Jordan 分解为 $A = X\mathrm{diag}(J_1, \ldots, J_p)X^{-1}$
  其中 $X \in \mathbb{C}^{n\times n}$ 非奇异
  $J_i = \mathrm{diag}(J_i^{(1)}, \ldots, J_i^{(m_i)}) \in \mathbb{C}^{n_i \times n_i}$ 是属于 λ_i 的 m_i 个 Jordan 块构成的 Jordan 块对角阵.

  $$J_i^{(m)} = \begin{bmatrix} \lambda_i & 1 & & \\ & \lambda_i & \ddots & \\ & & \ddots & 1 \\ & & & \lambda_i \end{bmatrix} \in \mathbb{C}^{n_i^{(m)} \times n_i^{(m)}} \text{ 且 } \begin{cases} n = n_1 + \cdots + n_p \\ n_i = n_i^{(1)} + \cdots + n_i^{(m_i)} \end{cases} (i = 1, \ldots, p)$$

  模最大特征值 λ_1 是半简单的 (即 λ_1 的几何重数等于其代数重数),
  表明属于 λ_1 的 Jordan 块只有一个 (即 $m_1 = 1$), 于是 Jordan 块对角阵 $J_1 = \lambda_1 I_{n_1}$
  记 $y = X^{-1}u^{(0)}$, 并将 y 和 X 对应分块为:

  $$X = [X_1, \ldots, X_p] \text{ where } X_i \in \mathbb{C}^{n\times n_i} \ (i = 1, \ldots, p)$$

  $$y = X^{-1}u^{(0)} = \begin{bmatrix} y_1 \\ \vdots \\ y_p \end{bmatrix} \text{ where } y_i \in \mathbb{C}^{n_i \times 1} \ (i = 1, \ldots, p)$$

于是我们有:

$$\begin{aligned}
A^k u^{(0)} &= [X\mathrm{diag}(J_1,\dots,J_p)X^{-1}]^k u^{(0)} \\
&= X\mathrm{diag}(J_1^k,\dots,J_p^k)X^{-1}u^{(0)} \\
&= [X_1,\dots,X_p]\begin{bmatrix} J_1^k & & \\ & \ddots & \\ & & J_p^k \end{bmatrix}\begin{bmatrix} y_1 \\ \vdots \\ y_p \end{bmatrix} \\
&= X_1J_1^ky_1 + X_2J_2^ky_2 + \cdots + X_pJ_p^ky_p \\
&= X_1\cdot\lambda_1^kI_{n_1}\cdot y_1 + X_2J_2^ky_2 + \cdots + X_pJ_p^ky_p \\
&= \lambda_1^k[X_1y_1 + X_2(\frac{J_2}{\lambda_1})^ky_2 + \cdots + X_p(\frac{J_p}{\lambda_1})^ky_p]
\end{aligned}$$

注意到 $\frac{J_i}{\lambda_1}$ 的谱半径 $\rho(\frac{J_i}{\lambda_1}) = \frac{|\lambda_i|}{|\lambda_1|} < 1 \ \ (i=2,\dots,p)$,
因此 $\lim_{k\to\infty}(\frac{J_i}{\lambda_1})^k = 0_{n_i\times n_i} \ \ (i=2,\dots,p)$
于是我们有:

$$\begin{aligned}
\lim_{k\to\infty}\frac{A^ku^{(0)}}{\lambda_1^k} &= \lim_{k\to\infty}\{X_1y_1 + X_2(\frac{J_2}{\lambda_1})^ky_2 + \cdots + X_p(\frac{J_p}{\lambda_1})^ky_p\} \\
&= \lim_{k\to\infty}X_1y_1 + X_2\lim_{k\to\infty}(\frac{J_2}{\lambda_1})^ky_2 + \cdots + X_p\lim_{k\to\infty}(\frac{J_p}{\lambda_1})^ky_p \\
&= X_1y_1 + X_20_{n_2\times n_2}y_2 + \cdots + X_p0_{n_p\times n_p}y_p \\
&= X_1y_1
\end{aligned}$$

由于初始向量 $u^{(0)}$ 在 λ_1 的特征子空间上的投影不为零，故 $X_1y_1 \neq 0_n$

注意到幂法产生的 $u^{(k)}$ 满足:

$$\|u^{(k)}\|_\infty = \left\|\frac{y^{(k)}}{\rho_k}\right\|_\infty = \frac{\|y^{(k)}\|_\infty}{|\rho_k|} = \frac{\|y^{(k)}\|_\infty}{\|y^{(k)}\|_\infty} = 1$$

$$u^{(k)} = \frac{y^{(k)}}{\rho_k} = \frac{Au^{(k-1)}}{\rho_k} = \cdots = \frac{A^ku^{(0)}}{\rho_k\cdots\rho_1}$$

注意到 ρ_k 是 $y^{(k)}$ 分量的模最大值，于是 $u^{(k)}$ 至少有一个分量为 1
因此 $\zeta_k = \rho_k\cdots\rho_1$ 必定是 $A^ku^{(0)}$ 分量的模最大值，
从而 $\frac{\zeta_k}{\lambda_1^k}$ 是 $\frac{A^ku^{(0)}}{\lambda_1^k}$ 分量的模最大值.

根据 $\lim_{k\to\infty}\frac{A^ku^{(0)}}{\lambda_1^k} = X_1y_1$ 可知极限 $\zeta = \lim_{k\to\infty}\frac{\zeta_k}{\lambda_1^k}$ 是存在的.
于是我们可知 $\{u^{(k)}\}$ 收敛，且有:

$$\begin{aligned}
\lim_{k\to\infty}u^{(k)} &= \lim_{k\to\infty}\frac{A^ku^{(0)}}{\zeta_k} \quad (\text{note that } u^{(k)} = \frac{A^ku^{(0)}}{\rho_k\cdots\rho_1} = \frac{A^ku^{(0)}}{\zeta_k}) \\
&= \lim_{k\to\infty}\left(\frac{\frac{A^ku^{(0)}}{\lambda_1^k}}{\frac{\zeta_k}{\lambda_1^k}}\right) \\
&= \frac{\lim_{k\to\infty}\frac{A^ku^{(0)}}{\lambda_1^k}}{\lim_{k\to\infty}\frac{\zeta_k}{\lambda_1^k}} \\
&= \frac{X_1y_1}{\zeta} \\
&\triangleq x_1
\end{aligned}$$

显然 $\{u^{(k)}\}$ 的极限 x_1 是 λ_1 的一个特征向量，其分量的模最大值为 1，因此 $\|x_1\|_\infty = 1$
结合等式 $\begin{cases} Au^{(k-1)} = \rho_ku^{(k)} \\ \lim_{k\to\infty}u^{(k-1)} = \lim_{k\to\infty}u^{(k)} = x_1 \end{cases}$ 可知 $\{\rho_k\}$ 收敛，且 $\lim_{k\to\infty}\rho_k = \lambda_1$

命题得证.

从**数值线性代数 定理** $6.2.1$ 的证明我们可以看出，
$\{\frac{A^ku^{(0)}}{\lambda_1^k}\}$ 的收敛速度主要取决于 $\{(\frac{J_2}{\lambda_1})^k\}$ 的收敛速度，即 $\rho(\frac{J_2}{\lambda_1}) = \frac{|\lambda_2|}{|\lambda_1|} \in (0,1)$ 的大小 (在该定理的条件下它总小于 1)

当 $\frac{|\lambda_2|}{|\lambda_1|}$ 接近于 1 时，收敛是很慢的，而且计算出的特征向量的敏感性会很大.
(特征向量的敏感性取决于模第一大和模第二大特征值 λ_1,λ_2 的分离度)
为加快幂法的收敛速度，通常可采用位移的方法，即应用幂法于 $A-\mu I$ 上.
通过适当选取位移因子 μ，可使 $A-\mu I$ 的模最大特征值与其他特征值的模长相对差距更大，从而起到加速收敛的效果.

例如对下面的例子取位移因子 $\mu = 3$，即可快速收敛到 A 的模最大特征值 $\lambda_1 = 3$ 对应的特征向量:

$$A = X\Lambda X^{-1}$$

$$X = \begin{bmatrix} 1 & & -1 & \\ & 1 & -1 & \\ 1 & 2 & 1 & 1 \\ -1 & & & 1 \end{bmatrix} \quad \Lambda = \begin{bmatrix} -3 & & & \\ & 2.999 & & \\ & & 2.99 & \\ & & & 2.9 \end{bmatrix} \quad X^{-1} = \frac{1}{5}\begin{bmatrix} 3 & -2 & 1 & -1 \\ -2 & 3 & 1 & -1 \\ -2 & -2 & 1 & -1 \\ 3 & -2 & 1 & 4 \end{bmatrix}$$

当 A 的模最大特征值不唯一时，由幂法产生的序列 $\{u^{(k)}\}$ 的收敛性分析将变得非常复杂.
此时 $\{u^{(k)}\}$ 可能有若干个收敛到不同向量的子序列.

- 考虑一个具体的例子:

$$A = X\Lambda X^{-1}$$

$$X = \begin{bmatrix} 1 & & -1 & \\ & 1 & -1 & \\ 1 & 2 & 1 & 1 \\ -1 & & & 1 \end{bmatrix} \qquad \Lambda = \begin{bmatrix} 3 & & & \\ & 2 & & \\ & & 1 & \\ & & & -3 \end{bmatrix} \qquad X^{-1} = \frac{1}{5}\begin{bmatrix} 3 & -2 & 1 & -1 \\ -2 & 3 & 1 & -1 \\ -2 & -2 & 1 & -1 \\ 3 & -2 & 1 & 4 \end{bmatrix}$$

此时 A 有两个模最大特征值 $\lambda_1 = 3$ 和 $\lambda_2 = -3$，不满足**数值线性代数 定理** $6.2.1$ 的条件.

取 $u^{(0)} = [1, 1, 1, 1]^{\mathrm{T}}$，则我们有:

$$A^k u^{(0)} = X\Lambda^k X^{-1} u^{(0)} = \frac{1}{5}\begin{bmatrix} 3^k + 4 \\ 2^k + 4 \\ 3^k + 2^{k+1} - 4 + 6(-3)^k \\ -3^k + 6(-3)^k \end{bmatrix}$$

于是我们有:

$$A^{2m} u^{(0)} = \frac{1}{5}\begin{bmatrix} 3^{2m} + 4 \\ 2^{2m} + 4 \\ 3^{2m} + 2^{2m+1} - 4 + 6(-3)^{2m} \\ -3^{2m} + 6(-3)^{2m} \end{bmatrix} \approx \frac{1}{5}\begin{bmatrix} 3^{2m} \\ 2^{2m} \\ 7 \cdot 3^{2m} + 2^{2m+1} \\ 5 \cdot 3^{2m} \end{bmatrix} = \frac{1}{5}(7 \cdot 3^{2m} + 2^{2m+1})\begin{bmatrix} \frac{1}{7} \\ 0 \\ 1 \\ \frac{5}{7} \end{bmatrix}$$

$$A^{2m-1} u^{(0)} = \frac{1}{5}\begin{bmatrix} 3^{2m-1} + 4 \\ 2^{2m-1} + 4 \\ 3^{2m-1} + 2^{2m} - 4 + 6(-3)^{2m-1} \\ -3^{2m-1} + 6(-3)^{2m-1} \end{bmatrix} \approx \frac{1}{5}\begin{bmatrix} 3^{2m-1} \\ 2^{2m-1} \\ -5 \cdot 3^{2m-1} + 2^{2m} \\ -7 \cdot 3^{2m-1} \end{bmatrix} = \frac{1}{5}(-7 \cdot 3^{2m-1})\begin{bmatrix} -\frac{1}{7} \\ 0 \\ \frac{5}{7} \\ 1 \end{bmatrix}$$

于是 $\{u^{(k)}\}$ 有两个收敛的子序列，分别收敛于向量 $x_1 = [\frac{1}{7}, 0, 1, \frac{5}{7}]^{\mathrm{T}}$ 和 $x_2 = [-\frac{1}{7}, 0, \frac{5}{7}, 1]^{\mathrm{T}}$

可以验证 $x_1 - x_2$ 和 $x_1 + x_2$ 分别是 $\lambda_1 = 3$ 和 $\lambda_2 = -3$ 对应的特征向量.

事实上，我们可以考虑更抽象的形式 **(数值线性代数, 习题** 6.30**)**:

$$A = X\Lambda X^{-1}$$

$$X = [x_1, x_2, \ldots, x_n] \qquad \Lambda = \mathrm{diag}(\lambda_1, \lambda_2, \ldots, \lambda_n) \qquad X^{-1} = \begin{bmatrix} y_1^{\mathrm{H}} \\ y_2^{\mathrm{H}} \\ \vdots \\ y_n^{\mathrm{H}} \end{bmatrix}$$

其中 A 有两个模最大特征值，不妨假设 $|\lambda_1| = |\lambda_2| > |\lambda_3| \geq |\lambda_4| \geq \cdots \geq |\lambda_n|$

设 $\lambda_1 = e^{i\theta}\lambda_2 \ \ (0 < \theta < 2\pi)$

若 $\theta = \frac{p}{q}2\pi$ (其中 p, q 是互质的正整数),

则由幂法产生的序列 $\{u^{(k)}\}$ 具有 q 个收敛子序列，分别收敛到以下向量:

$$e^{ik\theta}(y_1^{\mathrm{H}} u^{(0)})x_1 + (y_2^{\mathrm{H}} u^{(0)})x_2 \ \ (k = 1, \ldots, q)$$

幂法可以求出方阵 A 的一个模最大特征值 λ_1 及其对应的一个特征向量 x_1

如果我们还要求解 A 的第二个模最大特征值 λ_2，那么直接使用幂法进行迭代是行不通的，必须对原矩阵降阶，

即在知道了 λ_1 和 x_1 的前提下，将 A 降一阶，使其只包含 A 的其余特征值 $\lambda_2, \ldots, \lambda_n$

最简单实用的降阶技巧是正交变换.

设 $Ax_1 = \lambda_1 x_1$ (其中 λ_1 和 x_1 已知)

在复数域上计算 **Householder 变换** H 使得 $Hx_1 = \alpha e_1$，其中 e_1 是 $\mathbb{R}^n$ 的第 1 个标准单位基向量.

于是我们有 (注意 H^{H} 代表 H 的共轭转置):

$$\begin{gathered} Ax_1 = \lambda_1 x_1 \\ \Leftrightarrow \\ A \cdot H^{\mathrm{H}} e_1 = \lambda_1 \cdot H^{\mathrm{H}} e_1 \\ \Leftrightarrow \\ HAH^{\mathrm{H}} e_1 = \lambda_1 e_1 \end{gathered}$$

因此 HAH^{H} 具有形状 $HAH^{\mathrm{H}} = \begin{bmatrix} \lambda_1 & * \\ & \widetilde{A} \end{bmatrix}$

其中 $\widetilde{A}$ 是 $n-1$ 阶方阵，且其特征值为 A 的其余特征值 $\lambda_2, \ldots, \lambda_n$

我们对 $\widetilde{A}$ 应用幂法，即可得到 λ_2

最后需要指出的是，幂法的性能依赖于矩阵特征值的分布，因此实际使用 (特别是自动计算) 时相当不便.

只有在矩阵阶数非常高，无法利用其他更有效算法时，我们才使用幂法计算少数几个模最大的特征值及其特征向量.

但幂法的基本思想是重要的，通过它可以引入一些更有效的算法.

子空间迭代:

现假设 $A \in \mathbb{C}^{n \times n}$ 的特征值大小关系为 $|\lambda_1| \geq |\lambda_2| > |\lambda_3| > \cdots > |\lambda_n|$，$A = P\Lambda P^{-1}$

此时为避免 $|\lambda_1| \approx |\lambda_2|$ 造成的问题，我们可以同时迭代这两个特征值的特征向量.

任意给定初始向量 $u^{(0)}, v^{(0)} \in \mathbb{C}^n$，当 $|\lambda_1| = |\lambda_2|$ 时我们有:

$$\mathrm{span}\{A^k u^{(0)}, A^k v^{(0)}\} = \mathrm{span}\{p_1, p_2\}$$

其中 p_1, p_2 分别为 P 的第 $1, 2$ 列，即 λ_1, λ_2 对应的特征向量.
但是当 $|\lambda_1| > |\lambda_2|$ 时，$A^k u^{(0)}, A^k v^{(0)}$ 都会收敛到 p_1
为解决这一问题，我们每步都对 $[A^k u^{(0)}, A^k v^{(0)}]$ 做 QR 分解，
将这组基变为良态的归一化的基，使得 p_2 方向的量不至于掉下去.
于是我们有如下迭代:

$$\begin{gathered}
X_0 := [u^{(0)}, v^{(0)}] \\
\hline
X_k \leftarrow AX_{k-1} \\
[Q_k, R_k] \leftarrow \mathrm{QR}(X_k) \text{ where } \begin{cases} Q_k \in \mathbb{C}^{n\times 2} \\ R_k \in \mathbb{C}^{2\times 2} \end{cases} \\
X_k \leftarrow Q_k
\end{gathered}$$

最终收敛得到的 $X_{\text{converge}} \in \mathbb{C}^{n\times 2}$ 满足 $\mathrm{span}\{X_{\text{converge}}\} \approx \mathrm{span}\{p_1, p_2\}$
比较好的情况是 X_{converge} 的第一列是 p_1，第二列是 p_1, p_2 的线性组合.
比较坏的情况是 X_{converge} 的第一列和第二列都是 p_1, p_2 的线性组合.
无论如何，$\mathrm{span}\{X_{\text{converge}}\}$ 是 A 的一个 2 维不变子空间，我们有方法从中提取出 A 的特征向量 p_1, p_2
上述过程可推广到 k 维的情况，提取 A 的 k 维不变子空间.

若前 k 大特征值互不相同，则我们有:

$$\begin{gathered}
AX_{\text{converge}} = X_{\text{converge}} R \\
\Leftrightarrow \\
R := (X_{\text{converge}})^{\mathrm{H}} A X_{\text{converge}} \text{ is a upper triangular matrix}
\end{gathered}$$

基于上三角阵 R 我们可以很方便地得到前 k 大特征值对应的特征向量，参考 5.2 节末尾.

5.2 反幂法

反幂法，即对 A^{-1} 应用幂法来求 A 的模最小特征值及其特征向量.
因此其迭代格式为:

$$\begin{gathered}
\text{Given initial vector } z^{(0)} \text{ such that } \|z^{(0)}\|_\infty = 1 \\
\hline
\text{solve } Ay^{(k)} = z^{(k-1)} \text{ for } y^{(k)} \\
\rho_k = \|y^{(k)}\|_\infty = y^{(k)}_{j_{\max}} \text{ where } j_{\max} = \arg\max_{1\le j\le n} |y^{(k)}_j| \\
z^{(k)} = \frac{1}{\rho_k} y^{(k)} \\
\lambda^{(k)} = \frac{(z^{(k)})^{\mathrm{T}} A z^{(k)}}{(z^{(k)})^{\mathrm{T}}(z^{(k)})} \text{ (Rayleigh Quotient)} \\
r^{(k)} = Az^{(k)} - z^{(k)}\lambda^{(k)} \text{ (Residual)} \\
\text{Converge if } \|r^{(k)}\|_\infty \le \tau \cdot (\|A\|_\infty \|z^{(k)}\|_\infty + \|z^{(k)}\|_\infty |\lambda^{(k)}|) \text{ where } \tau > 0 \text{ is predefined tolerance}
\end{gathered}$$

其中线性方程组 $Ay^{(k)} = z^{(k-1)}$ 的求解可通过事先使用部分选主元的 Gauss 消元法计算 A 的 LU 分解，
然后通过前代法和回代法求解两个三角方程组来得到 $y^{(k)}$
(于是第一步是 $O(n^3)$，后续步骤都是 $O(n^2)$，因为 LU 分解可以重复利用)

根据幂法的性质可知，若 A 的特征值为 $|\lambda_n| < |\lambda_{n-1}| \le \cdots \le |\lambda_1|$
则 $\{z^{(k)}\}$ 收敛到 A 的对应于 λ_n 的一个特征向量，而 $\{\rho_k\}$ 收敛到 $\frac{1}{\lambda_n}$
收敛速度由 $\frac{|\frac{1}{\lambda_{n-1}}|}{|\frac{1}{\lambda_n}|} = \frac{|\lambda_n|}{|\lambda_{n-1}|}$ 的大小来决定，因此是全局线性收敛.

在实际应用中，反幂法主要用于求解已知特征值对应的特征向量.
假设已经求得 A 的某个特征值 λ 的近似值 $\hat{\lambda}$，
我们对 $A - \hat{\lambda} I$ 应用反幂法，就可以得到 λ 更精确的估计及其特征向量.

带位移 μ 的反幂法的迭代格式如下: (Homework 07 Problem 04)

$$\begin{gathered}
\text{Given initial vector } z^{(0)} \text{ such that } \|z^{(0)}\|_\infty = 1 \\
\hline
\text{solve } (A - \mu I) y^{(k)} = z^{(k-1)} \\
\rho_k = \|y^{(k)}\|_\infty \\
z^{(k)} = \frac{1}{\rho_k} y^{(k)} \\
\lambda^{(k)} = \frac{(z^{(k)})^{\mathrm{T}} A z^{(k)}}{(z^{(k)})^{\mathrm{T}}(z^{(k)})} \text{ (Rayleigh Quotient)} \\
r^{(k)} = Az^{(k)} - z^{(k)}\lambda^{(k)} \text{ (Residual)} \\
\text{Converge if } \|r^{(k)}\|_\infty \le \tau \cdot (\|A\|_\infty \|z^{(k)}\|_\infty + \|z^{(k)}\|_\infty |\lambda^{(k)}|) \text{ where } \tau > 0 \text{ if predefined tolerance}
\end{gathered}$$

其中线性方程组 $(A - \mu I) y^{(k)} = z^{(k-1)}$ 的求解可通过事先使用部分选主元的 Gauss 消元法计算 $A - \mu I$ 的 LU 分解，
然后通过前代法和回代法求解两个三角方程组来得到 $y^{(k)}$

从收敛速度的角度来考虑，μ 应尽可能 A 的某个特征值.
但这会导致 $A - \mu I$ 和一个奇异矩阵很接近，导致线性方程组 $(A - \mu I) y^{(k)} = z^{(k-1)}$ 十分病态.
然而实际计算的经验和理论结果表明:

- $A - \mu I$ 的病态性并不影响其收敛速度
- 当 μ 与 A 的某个特征值足够接近时，往往一次迭代就可以得到相当好的近似特征向量.

为弄清反幂法的特性，我们进行以下分析:

- 出于简化讨论的目的，我们使用 $\|\cdot\|_2$ 范数，实际使用时应替换为 $\|\cdot\|_\infty$ 范数.
  于是迭代格式写为:

$$\frac{\text{Given initial vector } z^{(0)} \text{ such that } \|z^{(0)}\|_\infty = 1}{\text{solve } (A-\mu I)y^{(k)} = z^{(k-1)}}$$
$$z^{(k)} = \frac{y^{(k)}}{\|y^{(k)}\|_2}$$

- 设 λ 是 A 的一个单特征值，x 是 λ 的单位特征向量 (即 $Ax=\lambda x$ 且 $\|x\|_2=1$).
  令 $U\in[x,U_2]\in\mathbb{C}^{n\times n}$ 为酉矩阵 (满足 $U^{\mathrm{H}}U=I$)，则我们有:

$$\begin{aligned}
U^{\mathrm{H}}AU &= \begin{bmatrix} x & U_2\end{bmatrix}^{\mathrm{H}} A \begin{bmatrix} x & U_2\end{bmatrix} \\
&= \begin{bmatrix} x^{\mathrm{H}}Ax & x^{\mathrm{H}}AU_2 \\ U_2^{\mathrm{H}}Ax & U_2^{\mathrm{H}}AU_2\end{bmatrix} \\
&= \begin{bmatrix} \lambda x^{\mathrm{H}}x & x^{\mathrm{H}}AU_2 \\ U_2^{\mathrm{H}}\lambda x & U_2^{\mathrm{H}}AU_2\end{bmatrix} \quad (\text{note that } x^{\mathrm{H}}x=\|x\|_2^2=1 \text{ and } U_2^{\mathrm{H}}x=0_{n-1}) \\
&= \begin{bmatrix} \lambda & x^{\mathrm{H}}AU_2 \\ 0 & U_2^{\mathrm{H}}AU_2\end{bmatrix}
\end{aligned}$$

  我们记 $A_2=U_2^{\mathrm{H}}AU_2\in\mathbb{C}^{(n-1)\times(n-1)}$，则有 $U^{\mathrm{H}}AU=\begin{bmatrix}\lambda & x^{\mathrm{H}}AU_2\\ & A_2\end{bmatrix}$

  由于 λ 是 A 的单特征值，故 A 的其余特征值都不等于 λ
  于是 $A_2=U_2^{\mathrm{H}}AU_2$ 的特征值均不等于 λ，因此 $\lambda I-A_2$ 非奇异.
  我们便可定义 $\Sigma^{\perp}=U_2(\lambda I-A_2)^{-1}U_2^{\mathrm{H}}$

  此外，由于 $\det\left(\lambda I-A^{\mathrm{T}}\right)=\det\left((\lambda I-A)^{\mathrm{T}}\right)=\det(\lambda I-A)=0$，
  故存在非零向量 $y\in\mathbb{C}^n$ 使得 $A^{\mathrm{T}}y=\lambda y$，即有 $y^{\mathrm{T}}A=\lambda y^{\mathrm{T}}$ (我们称 y 为 A 关于 λ 的左特征向量)
  由于 λ 是 A 的单特征值，故 x,y 不正交 (即 $y^{\mathrm{T}}x\neq 0$)，于是可以缩放 y 使得 $y^{\mathrm{T}}x=1$

  若 A 经微小扰动 ΔA 后得到 $\widetilde{A}=A+\Delta A$，
  则可以证明存在 $\widetilde{A}$ 的一个特征值 $\widetilde{\lambda}$ 和对应的特征向量 $\widetilde{x}$ 使得:

$$\begin{aligned}
|\widetilde{\lambda}-\lambda| &\le \|y\|_2\|\Delta A\|_2 + O(\|\Delta A\|_2^2) \\
\|\widetilde{x}-x\|_2 &\le \|\Sigma^{\perp}\|_2\|\Delta A\|_2 + O(\|\Delta A\|_2^2)
\end{aligned}$$

  这表明 λ 和 x 关于扰动 ΔA 的敏感性分别于 $\|y\|_2$ 和 $\|\Sigma^{\perp}\|_2$ 有关.
  因此我们定义**特征值 λ 的条件数**为 $\mathrm{cond}(\lambda)=\|y\|_2$，**单位特征向量 x 的条件数** $\mathrm{cond}(x)=\|\Sigma^{\perp}\|_2$

现假定带位移 μ 的反幂法中，位移因子 μ 和单特征值 λ 非常接近，
且单位特征向量 x 是良态的，即其条件数 $\mathrm{cond}(x)$ 很小.

$$\begin{aligned}
\mathrm{cond}(x) &= \|\Sigma^{\perp}\|_2 \\
&= \|U_2(\lambda I-A_2)^{-1}U_2^{\mathrm{H}}\|_2 \\
&= \sigma_{\max}(U_2(\lambda I-A_2)^{-1}U_2^{\mathrm{H}}) \\
&= \sqrt{\lambda_{\max}[(U_2(\lambda I-A_2)^{-1}U_2^{\mathrm{H}})^{\mathrm{H}}(U_2(\lambda I-A_2)^{-1}U_2^{\mathrm{H}})]} \quad (\text{note that } U_2^{\mathrm{H}}U_2=I_{n-1}) \\
&= \sqrt{\lambda_{\max}[U_2(\lambda I-A_2)^{-H}U_2^{\mathrm{H}}U_2(\lambda I-A_2)^{-1}U_2^{\mathrm{H}}]} \\
&= \sqrt{\lambda_{\max}[U_2(\lambda I-A_2)^{-H}(\lambda I-A_2)^{-1}U_2^{\mathrm{H}}]} \\
&= \sigma_{\max}((\lambda I-A_2)^{-1}U_2^{\mathrm{H}}) \\
&= \|(\lambda I-A_2)^{-1}U_2^{\mathrm{H}}\|_2
\end{aligned}$$

任意给定初始向量 $z^{(0)}$.
使用部分主元的 Gauss 消去法求解线性方程组 $(A-\mu I)y^{(1)}=z^{(0)}$，得到计算解 $\tilde{y}^{(1)}$
根据 Gauss 消去法的误差分析可知计算解 $\tilde{y}^{(1)}$ 满足 $(A-\mu I-\Delta A)\tilde{y}^{(1)}=z^{(0)}$
其中扰动 $\|\Delta A\|_2$ 具有一致的上界，通常是机器精度级别的.

- > **(数值线性代数, 定理 2.4.1)**
  > 若 $A=[a_{ij}]\in\mathbb{R}^{n\times n}$ 非奇异 (即满秩)，
  > 则用部分主元的 Gauss 消去法解线性方程组 $Ax=b$ 得到的计算解 $\widetilde{x}$ 满足 $\begin{cases}(A+\Delta A)\widetilde{x}=b\\ \frac{\|\Delta_A\|_\infty}{\|A\|_\infty}\le n^2\rho\gamma_{3n}\end{cases}$
  > 其中 γ_{3n} 代表 $3n$ 层浮点运算的累积相对误差的绝对值.
  >
  > - 这个定理表明:
  >   部分主元的 Gauss 消去法得到的计算解相当于系数矩阵作某些扰动而得到的扰动方程组的精确解.
  >   一般来说，Δ_A 的元素 (舍入误差) 比起 A 的元素的初始误差 (测量误差、模型误差等) 来是很小的.
  >   从这个意义上来说，部分主元 Gauss 消去法是数值稳定的.
  > - 理论上可以证明部分主元 Gauss 消去法的增长因子 $\rho\le 2^{n-1}$，且上界可以达到.
  >   但在实际问题中，ρ 通常很小 (满足 $\rho\le n$)
  >
  >   此外，上界 $n^2\rho\gamma_{3n}$ 一般比 $\frac{\|\Delta_A\|_\infty}{\|A\|_\infty}$ 大得多.
  >   在实际问题中，通常都有 $\frac{\|\Delta_A\|_\infty}{\|A\|_\infty}\approx\mathrm{eps}$ 成立.

记 $\Delta y^{(1)}=\tilde{y}^{(1)}-y^{(1)}=(A-\mu I)^{-1}(\Delta A\tilde{y}^{(1)}+z^{(0)})-(A-\mu I)^{-1}z^{(0)}=(A-\mu I)^{-1}\Delta A\tilde{y}^{(1)}$
我们可以将 $\Delta y^{(1)}$ 分解为 $\mathrm{span}\{x\}$ 和 $\mathrm{span}\{x\}^{\perp}$ 上的投影:

$$\Delta y^{(1)}=\Delta y_1^{(1)}+\Delta y_2^{(1)} \text{ where } \begin{cases}\Delta y_1^{(1)}\in\mathrm{span}\{x\}\\ \Delta y_2^{(1)}\in\mathrm{span}\{x\}^{\perp}=\mathrm{span}\{U_2\}\end{cases}$$

则存在 $\alpha \in \mathbb{C}$ 和 $c \in \mathbb{C}^{n-1}$，使得 $\Delta y^{(1)} = \alpha x + U_2 c$
注意到 $\begin{cases} \alpha = x^{\mathrm{H}} \Delta y^{(1)} \\ c = U_2^{\mathrm{H}} \Delta y^{(1)} \end{cases}$

- 回忆起 $U^{\mathrm{H}} A U = \begin{bmatrix} \lambda & x^{\mathrm{H}} A U_2 \\ & A_2 \end{bmatrix}$ (其中 $U = [x, U_2]$ 为酉矩阵)
  于是我们有:

$$U^{\mathrm{H}}(A-\mu I)U = U^{\mathrm{H}}AU - \mu I = \begin{bmatrix} \lambda-\mu & x^{\mathrm{H}}AU_2 \\ & A_2-\mu I \end{bmatrix}$$

$$\Leftrightarrow$$

$$U^{\mathrm{H}}(A-\mu I)^{-1}U = (U^{\mathrm{H}}(A-\mu I)U)^{-1} = \begin{bmatrix} \lambda-\mu & x^{\mathrm{H}}AU_2 \\ & A_2-\mu I \end{bmatrix}^{-1} = \begin{bmatrix} \frac{1}{\lambda-\mu} & -\frac{1}{\lambda-\mu}x^{\mathrm{H}}AU_2(A_2-\mu I)^{-1} \\ & (A_2-\mu I)^{-1} \end{bmatrix}$$

$$\Leftrightarrow$$

$$(A-\mu I)^{-1} = [x \quad U_2] \begin{bmatrix} \frac{1}{\lambda-\mu} & -\frac{1}{\lambda-\mu}x^{\mathrm{H}}AU_2(A_2-\mu I)^{-1} \\ & (A_2-\mu I)^{-1} \end{bmatrix} \begin{bmatrix} x^{\mathrm{H}} \\ U_2^{\mathrm{H}} \end{bmatrix} = \frac{xx^{\mathrm{H}}}{\lambda-\mu}[I - AU_2(A_2-\mu I)^{-1}U_2^{\mathrm{H}}] + U_2(A_2-\mu I)^{-1}U_2^{\mathrm{H}}$$

$$\Leftrightarrow$$

$$\Delta y^{(1)} = (A-\mu I)^{-1}\Delta A\tilde{y}^{(1)} = \{\frac{xx^{\mathrm{H}}}{\lambda-\mu}[I - AU_2(A_2-\mu I)^{-1}U_2^{\mathrm{H}}] + U_2(A_2-\mu I)^{-1}U_2^{\mathrm{H}}\}\Delta A\tilde{y}^{(1)}$$

因此我们有:

$$\begin{aligned}
\alpha &= x^{\mathrm{H}}\Delta y^{(1)} \\
&= x^{\mathrm{H}}\{\frac{xx^{\mathrm{H}}}{\lambda-\mu}[I - AU_2(A_2-\mu I)^{-1}U_2^{\mathrm{H}}] + U_2(A_2-\mu I)^{-1}U_2^{\mathrm{H}}\}\Delta A\tilde{y}^{(1)} \\
&= \{\frac{x^{\mathrm{H}}x}{\lambda-\mu}x^{\mathrm{H}}[I - AU_2(A_2-\mu I)^{-1}U_2^{\mathrm{H}}] + x^{\mathrm{H}}U_2(A_2-\mu I)^{-1}U_2^{\mathrm{H}}\}\Delta A\tilde{y}^{(1)} \quad (\text{note that } x^{\mathrm{H}}x = 1 \text{ and } U_2^{\mathrm{H}}x = 0_{n-1}) \\
&= \frac{1}{\lambda-\mu}x^{\mathrm{H}}[I - AU_2(A_2-\mu I)^{-1}U_2^{\mathrm{H}}]\Delta A\tilde{y}^{(1)}
\end{aligned}$$

$$\begin{aligned}
c &= U_2^{\mathrm{H}}\Delta y^{(1)} \\
&= U_2^{\mathrm{H}}\{\frac{xx^{\mathrm{H}}}{\lambda-\mu}[I - AU_2(A_2-\mu I)^{-1}U_2^{\mathrm{H}}] + U_2(A_2-\mu I)^{-1}U_2^{\mathrm{H}}\}\Delta A\tilde{y}^{(1)} \\
&= \{\frac{U_2xx^{\mathrm{H}}}{\lambda-\mu}[I - AU_2(A_2-\mu I)^{-1}U_2^{\mathrm{H}}] + U_2^{\mathrm{H}}U_2(A_2-\mu I)^{-1}U_2^{\mathrm{H}}\}\Delta A\tilde{y}^{(1)} \quad (\text{note that } U_2^{\mathrm{H}}x = 0_{n-1} \text{ and } U_2^{\mathrm{H}}U_2 = I_{n-1}) \\
&= (A_2-\mu I)^{-1}U_2^{\mathrm{H}}\Delta A\tilde{y}^{(1)}
\end{aligned}$$

因此我们有:

$$\begin{cases}
\begin{aligned}
\|\Delta y_1^{(1)}\|_2 &= \|\alpha x\|_2 \\
&= \|\tfrac{1}{\lambda-\mu}x^{\mathrm{H}}[I - AU_2(A_2-\mu I)^{-1}U_2^{\mathrm{H}}]\Delta A\tilde{y}^{(1)}\|_2 \\
&\le \tfrac{1}{|\lambda-\mu|}(\|I\|_2 + \|A\|_2\|U_2(A_2-\mu I)^{-1}U_2^{\mathrm{H}}\|_2)\|\Delta A\tilde{y}^{(1)}\|_2 \\
&= \tfrac{1}{|\lambda-\mu|}(I + \|A\|_2\|(A_2-\mu I)^{-1}U_2^{\mathrm{H}}\|_2)\|\Delta A\tilde{y}^{(1)}\|_2
\end{aligned} \\
\begin{aligned}
\|\Delta y_2^{(1)}\|_2 &= \|U_2 c\|_2 \\
&= \|U_2(A_2-\mu I)^{-1}U_2^{\mathrm{H}}\Delta A\tilde{y}^{(1)}\|_2 \\
&\le \|U_2(A_2-\mu I)^{-1}U_2^{\mathrm{H}}\|_2\|\Delta A\tilde{y}^{(1)}\|_2 \\
&= \|(A_2-\mu I)^{-1}U_2^{\mathrm{H}}\|_2\|\Delta A\tilde{y}^{(1)}\|_2
\end{aligned}
\end{cases}$$

注意到 $(A_2-\mu I)^{-1} = [I + (\lambda-\mu)(A_2-\lambda I)^{-1}]^{-1}(A_2-\lambda I)^{-1}$
因此当 μ 非常接近于 λ 时，$(A_2-\mu I)^{-1}$ 与 $(A_2-\lambda I)^{-1}$ 也非常接近.
于是 $\|(A_2-\mu I)^{-1}U_2^{\mathrm{H}}\|_2 \approx \|(A_2-\lambda I)^{-1}U_2^{\mathrm{H}}\|_2 = \mathrm{cond}(x)$
因此在单位特征向量 x 良态 ($\mathrm{cond}(x)$ 很小) 的条件下，$\|(A_2-\mu I)^{-1}U_2^{\mathrm{H}}\|_2$ 也很小.
这表明 $\|\Delta y_2^{(1)}\|_2$ 也会很小.
而由于 μ 非常接近于 λ，故 $\|\Delta y_1^{(1)}\|_2$ 会很大.

换言之，部分主元 Gauss 消元法求解线性方程组 $(A-\mu I)y^{(1)} = z^{(0)}$ 所引起的误差 $\Delta y^{(1)}$
主要影响计算解 $\tilde{y}^{(1)} = y^{(1)} + \Delta y^{(1)}$ 在特征子空间 $\mathrm{span}\{x\}$ 上的投影长度，
因为当 μ 非常接近于 λ 且单位特征向量 x 良态 ($\mathrm{cond}(x)$ 很小) 时，
误差 $\Delta y^{(1)}$ 在特征子空间 $\mathrm{span}\{x\}$ 上的投影长度 $\|\Delta y_1^{(1)}\|_2$ 会显著大于在 $\mathrm{span}\{x\}^{\perp}$ 上的投影长度 $\|\Delta y_2^{(1)}\|_2$

位移因子 μ 离单特征值 λ 越近，矩阵 $A-\mu I$ 就越病态，
部分主元 Gauss 消元法求解线性方程组 $(A-\mu I)y^{(1)} = z^{(0)}$ 所引起的误差 $\Delta y^{(1)}$ 就越大，
误差 $\Delta y^{(1)}$ 在特征子空间 $\mathrm{span}\{x\}$ 上的投影 $\Delta y_1^{(1)}$ 就会越显著，
计算解 $\tilde{y}^{(1)} = y^{(1)} + \Delta y^{(1)}$ 在特征子空间 $\mathrm{span}\{x\}$ 上的投影就会越显著.
由于我们以 $\tilde{y}^{(1)}$ 的方向作为特征向量的近似方向，故这是十分有利的.

因此当 μ 非常接近于单特征值 λ 且单位特征向量 x 良态 ($\mathrm{cond}(x)$ 很小) 时，
我们只需使用一次反幂法就可以得到 λ 的较好的近似特征向量.

当单特征值 λ 比较病态 (即 $\mathrm{cond}(\lambda)$ 很大, 前面有定义) 时，
利用反幂法再进行第二次迭代一般不会得到更好的近似特征向量.

最后，我们介绍一下在实际计算中初始向量 $z^{(0)}$ 的两种常用的选取方法.

- 第一种是使用随机数产生随机向量，再依 $\|\cdot\|_\infty$ 范数单位化得到初始向量 $z^{(0)}$
- 第二种是 "半次迭代法"
  在带位移 μ 的反幂法中，首先要使用部分主元的 Gauss 消去法计算 $A-\mu I$ 的 LU 分解 $P(A-\mu I) = LU$.
  那么第一次反幂法迭代为:

$$\begin{cases}(A-\mu I)y^{(1)}=z^{(0)}\\ z^{(1)}=\frac{y^{(1)}}{\|y^{(1)}\|_\infty}\end{cases}\Leftrightarrow\begin{cases}P(A-\mu I)y^{(1)}=LUy^{(1)}=Pz^{(0)}\\ z^{(1)}=\frac{y^{(1)}}{\|y^{(1)}\|_\infty}\end{cases}$$

我们取 $z^{(0)}=P^{\mathrm{T}}L1_n$，则为求解 $y^{(1)}$，只需求解一个三角方程组 $Uy^{(1)}=1_n$
这样，初始向量 $z^{(0)}$ 不需要明确给出，而完成这一次反幂法迭代就只需求解 "半次" 线性方程组.

Rayleigh 商迭代: (Homework 07 Problem 04)

$$\frac{\text{Given initial vector } z^{(0)} \text{ such that } \|z^{(0)}\|_\infty=1, \text{and } \mu_0\in\mathbb{C}}{\begin{gathered}\text{solve } (A-\mu_{k-1}I)y^{(k)}=z^{(k-1)}\\ \rho_k=\|y^{(k)}\|_\infty\\ z^{(k)}=\frac{1}{\rho_k}y^{(k)}\\ \lambda^{(k)}=\frac{(z^{(k)})^{\mathrm{T}}Az^{(k)}}{(z^{(k)})^{\mathrm{T}}(z^{(k)})}\ (\text{Rayleigh Quotient})\\ r^{(k)}=Az^{(k)}-z^{(k)}\lambda^{(k)}\ (\text{Residual})\\ \text{Converge if } \|r^{(k)}\|_\infty\le\tau\cdot(\|A\|_\infty\|z^{(k)}\|_\infty+\|z^{(k)}\|_\infty|\lambda^{(k)}|) \text{ where } \tau>0 \text{ is predefined tolerance}\\ \mu_k=\lambda^{(k)}\end{gathered}}$$

局部二次收敛 (若 A 具有对称性，则有局部三次收敛)
但如果初值 μ_0 选的不好，则全局收敛性可能不会太好.
而且由于每步都需要求解 $A-\mu_kI$ 的 LU 分解，故每步的代价都是 $O(n^3)$

$$\begin{cases}\text{Linear convergence: } \|r^{(k+1)}\|\le\gamma\|r^{(k)}\|\\ \text{Quadratic convergence: } \|r^{(k+1)}\|\le\gamma\|r^{(k)}\|^2\ (0<\gamma<1)\\ \text{Cubic convergence: } \|r^{(k+1)}\|\le\gamma\|r^{(k)}\|^3\end{cases}$$

一个严重的问题是，Rayleigh 商迭代过程中 $A-\mu_kI$ 会出现减法相消.
一个更严重的问题是，$A-\mu_kI$ 会变得越来越病态，越来越解不准 (对扰动敏感)
(特别地，我们所求的特征值还是 $A-\mu_kI$ 的模最小特征值的特征向量)
但幸运的是，虽然特征向量解不准 (大小不准)，但基本方向是准的 (即病态性产生巨大扰动的方向正是我们想要的方向)，
因此对 Rayleigh 商序列 $\{\mu_k\}$ 的收敛性影响不大 (也就是说这个病态性对特征值的精度是不影响的)

> 邵老师提到的一个有趣事实:
>
> $$\begin{cases}\sigma_{\max}=\|A\|_2\ge\rho(A)=|\lambda_{\max}|\\ \sigma_{\min}=\frac{1}{\|A^{-1}\|_2}\le\frac{1}{\rho(A^{-1})}=|\lambda_{\min}|\end{cases}\Rightarrow\sigma_{\max}\ge|\lambda_{\max}|\ge|\lambda_{\min}|\ge\sigma_{\min}$$
>
> 因此如果有小特征值，那么会有更小的奇异值.

设已知 $A\in\mathbb{C}^{n\times n}$ 的 Schur 分解 $A=UTU^{\mathrm{H}}$
则我们有 $(A-\mu I)^{-1}=U(T-\mu I)^{-1}U^{\mathrm{H}}$
因此我们只需对 T 应用反幂法即可.

考虑 $n=4$ 的情况，假设我们想计算 $(2,2)$ 位置的特征值 $\lambda_2=T(2,2)$ 的特征向量，则我们可以这样做:

$$\begin{aligned}&\frac{\text{Solve } (T-\lambda_2I)x=0_n}{x=(T-\lambda_2I)^{-1}0_n}\\ &=\begin{bmatrix}*&*&*&*\\ &0&*&*\\ &&*&*\\ &&&*\end{bmatrix}^{-1}\begin{bmatrix}0\\0\\0\\0\end{bmatrix}\\ &=\begin{bmatrix}\infty\\\infty\\0\\0\end{bmatrix}(\text{normalize})\propto\begin{bmatrix}*\\1\\0\\0\end{bmatrix}\end{aligned}$$

根据这个思路我们可以将右端的零向量的第二个元素置为 1，将 $T(2,2)$ 置为 1，即求解:

$$x=\begin{bmatrix}*&*&*&*\\ &1&*&*\\ &&*&*\\ &&&*\end{bmatrix}^{-1}\begin{bmatrix}0\\1\\0\\0\end{bmatrix}=\begin{bmatrix}*\\1\\0\\0\end{bmatrix}$$

注意到这实际上相当于求解一个更小阶数的上三角方程组，最后在解的后面补 0 得到原三角方程组的解:

$$\widetilde{x}=\begin{bmatrix}*&*\\ &1\end{bmatrix}^{-1}\begin{bmatrix}0\\1\end{bmatrix}=\begin{bmatrix}*\\1\end{bmatrix}$$

$$x=\begin{bmatrix}\widetilde{x}\\0_2\end{bmatrix}=\begin{bmatrix}*\\1\\0\\0\end{bmatrix}$$

上述方法的一般实现可参考 Homework 07 Problem 05.

邵老师的思路: (存疑: 实际使用随机右端向量时解不准，我搞不懂最后的推广有什么意义.)
考虑 $n=4$ 的情况，假设我们想计算 $(2,2)$ 位置的特征值 $\lambda_2=T(2,2)$ 的特征向量，则我们可以这样做:

$$
\begin{aligned}
&\frac{\text{Solve } (T-\lambda_2 I)x = 0_n}{x = (T-\lambda_2 I)^{-1}0_n} \\
&= \begin{bmatrix} * & * & * & * \\ & 0 & * & * \\ & & * & * \\ & & & * \end{bmatrix}^{-1} \begin{bmatrix} 0 \\ 0 \\ 0 \\ 0 \end{bmatrix} \\
&= \begin{bmatrix} \infty \\ \infty \\ 0 \\ 0 \end{bmatrix} (\text{normalize}) \propto \begin{bmatrix} * \\ 1 \\ 0 \\ 0 \end{bmatrix}
\end{aligned}
$$

根据这个思路我们可以将右端的零向量换为任意向量:

$$
\begin{aligned}
&\frac{\text{Solve } (T-\lambda_2 I)x = b \text{ where } b \text{ could be any vector in } \mathbb{C}^n}{x = (T-\lambda_2 I)^{-1}b} \\
&= \begin{bmatrix} * & * & * & * \\ & 0 & * & * \\ & & * & * \\ & & & * \end{bmatrix}^{-1} \begin{bmatrix} * \\ * \\ * \\ * \end{bmatrix} \\
&= \begin{bmatrix} \infty \\ \infty \\ * \\ * \end{bmatrix} (\text{normalize}) \propto \begin{bmatrix} * \\ 1 \\ 0 \\ 0 \end{bmatrix}
\end{aligned}
$$

这是合理的，因为 b/∞ 就近似一个零向量.
在实际使用回代法求解时，
我们解出第 2 个位置的 ∞ (一个非常大的数) 就将其设为 1 (这使得回代法可以继续进行)，再往上求解.

5.3 QR 方法

QR 方法是目前计算一般方阵的全部特征值和特征向量的最有效方法之一.
它利用正交变换将一个给定方阵逐步约化为上三角阵或拟上三角阵 (上 Hessenberg矩阵) 的一种迭代方法.
其基本收敛速度是二次的，当原方阵实对称时，可达到三次收敛.

5.3.1 显式 QR 迭代格式与收敛性

显式 QR 算法的迭代格式如下: **(形式待改进)**

$$
\begin{aligned}
&\frac{\text{Given } A_0 = A \in \mathbb{C}^{n\times n}}{A_{k-1} = Q_k R_k} \\
&A_k = R_k Q_k
\end{aligned}
$$

其中 Q_k 为酉矩阵 ($Q_k^{\mathrm{H}} Q_k = I$)，R_k 为上三角阵 (为方便起见，我们要求 R_k 的对角元均为非负实数)
由迭代格式容易推出 $A_k = R_k Q_k = Q_k^{\mathrm{H}} A_{k-1} Q_k$，因此序列 $\{A_k\}$ 中的每个矩阵都与原矩阵 A 相似.

$$
\begin{aligned}
A_k &= Q_k^{\mathrm{H}} A_{k-1} Q_k \\
&= Q_k^{\mathrm{H}} (Q_{k-1}^{\mathrm{H}} A_{k-2} Q_{k-1}) Q_k \\
&= \cdots \\
&= (Q_1 Q_2 \cdots Q_k)^{\mathrm{H}} A_0 (Q_1 Q_2 \cdots Q_k) \\
&= \widetilde{Q}_k^{\mathrm{H}} A \widetilde{Q}_k \quad (\text{denote } \widetilde{Q}_k = Q_1 Q_2 \cdots Q_k)
\end{aligned}
$$

结合 $A_k = Q_{k+1} R_{k+1}$ 可知:

$$
\begin{gathered}
A_k = \widetilde{Q}_k^{\mathrm{H}} A \widetilde{Q}_k = Q_{k+1} R_{k+1} \\
\Leftrightarrow \\
\widetilde{Q}_k Q_{k+1} R_{k+1} = A \widetilde{Q}_k \\
\Leftrightarrow \\
\widetilde{Q}_k Q_{k+1} R_{k+1} (R_k R_{k-1} \cdots R_1) = A \widetilde{Q}_k (R_k R_{k-1} \cdots R_1) \\
\Leftrightarrow \\
\widetilde{Q}_{k+1} \widetilde{R}_{k+1} = A \widetilde{Q}_k \widetilde{R}_k \\
(\text{denote } \widetilde{R}_k = R_k R_{k-1} \cdots R_1) \\
\Leftrightarrow \\
A^k = \widetilde{Q}_k \widetilde{R}_k
\end{gathered}
$$

利用 $A^k = \widetilde{Q}_k \widetilde{R}_k$ 这一基本关系式，我们可以导出 QR 算法与幂法的联系.
记 $\widetilde{R}_k$ 的元素为 $\tilde{r}_{ij}^{(k)}$，$\widetilde{Q}_k$ 的第一列为 $\tilde{q}_1^{(k)}$，则我们有 $A^k e_1 = \tilde{r}_{11}^{(k)} \tilde{q}_1^{(k)}$
于是 $\tilde{q}_1^{(k)}$ 可以看做对 A 用 e_1 作为初始向量的幂法进行 k 次迭代得到的向量.
若 A 的模最大特征值 λ_1 与其他特征值分离，则 $\{\tilde{q}_1^{(k)}\}$ 收敛到 A 关于 λ_1 的一个特征向量.

事实上，在适当条件下 (下面给出了一个易于验证的情况)，A_k 所有的或大部分的严格下三角元都将趋于零.
(数值线性代数, 定理 6.4.1)
设 A 的 n 个特征值满足 $|\lambda_1| > |\lambda_2| > \cdots > |\lambda_n| > 0$，对应的特征向量构成 $X = [x_1, \ldots, x_n]$
记 $Y = X^{-1}$，其第 i 行为 A 关于 λ_i 的左特征向量.
若 Y 具有 LU 分解 (即 Y 的前 $n-1$ 个顺序主子阵都是非奇异的 **数值线性代数 定理** 1.1.2),
则由 QR 算法产生的矩阵 $A_k = [a_{ij}^{(k)}]$ 的严格下三角元都趋于零，且对角元 $a_{ii}^{(k)}$ 趋向于 λ_i $(i = 1, \ldots, n)$

- **上述定理表明:**
  QR 算法是线性收敛的 (每次迭代的前后误差是成比例的)，收敛速度取决于特征值的分离程度.
- **证明:**
  记 $\Lambda = \mathrm{diag}(\lambda_1, \dots, \lambda_n)$，则有 $A = X\Lambda X^{-1} = X\Lambda Y$
  设 Y 的 LU 分解为 $Y = LU$，其中 L 是单位下三角阵，U 是上三角阵.
  则我们有:

$$\begin{aligned} A^k &= (X\Lambda Y)^k \\ &= X\Lambda^k Y \\ &= X\Lambda^k LU \\ &= X(\Lambda^k L\Lambda^{-k})\Lambda^k U \\ &= X(I + E_k)\Lambda^k U \quad (\text{Let } I + E_k = \Lambda^k L\Lambda^{-k}) \end{aligned}$$

  显然 $I + E_k = \Lambda^k L\Lambda^{-k}$ 仍是一个单位下三角阵，具有 n 重特征值 1.
  因此 E_k 具有 n 重特征值 0，于是 $\rho(E_k) = 0 < 1$，从而 $\lim\limits_{k\to\infty} E_k = O_{n\times n}$

  现令 X 的 QR 分解为 $X = QR$
  由于 X 非奇异，故我们可要求 R 的对角元均为正实数，从而可逆.
  代入 $A^k = X(I + E_k)\Lambda^k U$ 即有:

$$\begin{aligned} A^k &= X(I + E_k)\Lambda^k U \\ &= QR(I + E_k)\Lambda^k U \\ &= Q(I + RE_kR^{-1})R\Lambda^k U \end{aligned}$$

  根据 $\lim\limits_{k\to\infty} E_k = O_{n\times n}$ 可知，当 k 充分大时，方阵 $(I + RE_kR^{-1})$ 非奇异.
  设其 QR 分解为 $I + RE_kR^{-1} = \hat{Q}_k\hat{R}_k$ (其中 $\hat{R}_k$ 的对角元均为正实数)
  根据 $\lim\limits_{k\to\infty} E_k = O_{n\times n}$ 可知 $\lim\limits_{k\to\infty} \hat{Q}_k = \lim\limits_{k\to\infty} \hat{R}_k = I_n$
  代入 $A^k = Q(I + RE_kR^{-1})RA^kU$ 即有:

$$\begin{aligned} A^k &= Q(I + RE_kR^{-1})R\Lambda^k U \\ &= Q(\hat{Q}_k\hat{R}_k)R\Lambda^k U \\ &= (Q\hat{Q}_k)(\hat{R}_kR\Lambda^k U) \end{aligned}$$

  这样我们就得到了 A^k 的一个 QR 分解 $A^k = (Q\hat{Q}_k)(\hat{R}_kR\Lambda^k U)$
  为保证 QR 分解中上三角阵的对角元均为正数，我们定义:

$$\begin{aligned} D_1 &= \mathrm{diag}(\frac{\lambda_1}{|\lambda_1|}, \dots, \frac{\lambda_n}{|\lambda_n|}) \\ D_2 &= \mathrm{diag}(\frac{u_{11}}{|u_{11}|}, \dots, \frac{u_{nn}}{|u_{nn}|}) \end{aligned}$$

  其中 u_{ii} 是 U 的第 i 个对角元，于是我们有:

$$\begin{aligned} A^k &= (Q\hat{Q}_k)(\hat{R}_kR\Lambda^k U) \\ &= (Q\hat{Q}_kD_1^kD_2)(D_2^{-1}D_1^{-k}\hat{R}_kR\Lambda^k U) \end{aligned}$$

  回忆起 QR 算法中 $A^k = \widetilde{Q}_k\widetilde{R}_k$
  根据非奇异矩阵的 (限制上三角阵的对角元为正实数的) QR 分解的存在唯一性可知:

$$\begin{gathered} A^k = \widetilde{Q}_k\widetilde{R}_k = (Q\hat{Q}_kD_1^kD_2)(D_2^{-1}D_1^{-k}\hat{R}_kR\Lambda^k U) \\ \Leftrightarrow \\ \begin{cases} \widetilde{Q}_k = Q\hat{Q}_kD_1^kD_2 \\ \widetilde{R}_k = D_2^{-1}D_1^{-k}\hat{R}_kR\Lambda^k U \end{cases} \end{gathered}$$

  代入 $A_k = \widetilde{Q}_k^{\mathrm{H}}A\widetilde{Q}_k$ (前文结论) 可知:

$$\begin{aligned} A_k &= \widetilde{Q}_k^{\mathrm{H}}A\widetilde{Q}_k \\ &= (Q\hat{Q}_kD_1^kD_2)^{\mathrm{H}}A(Q\hat{Q}_kD_1^kD_2) \\ &= (D_2^{\mathrm{H}}(D_1^{\mathrm{H}})^k\hat{Q}_k^{\mathrm{H}}Q^{\mathrm{H}}) \cdot A \cdot (Q\hat{Q}_kD_1^kD_2) \quad (\text{note that } A = X\Lambda X^{-1} = (QR)\Lambda(QR)^{-1} = QR\Lambda R^{-1}Q^{\mathrm{H}}) \\ &= (D_2^{\mathrm{H}}(D_1^{\mathrm{H}})^k\hat{Q}_k^{\mathrm{H}}Q^{\mathrm{H}})(QR\Lambda R^{-1}Q^{\mathrm{H}})(Q\hat{Q}_kD_1^kD_2) \\ &= D_2^{\mathrm{H}}(D_1^{\mathrm{H}})^k\hat{Q}_k^{\mathrm{H}} \cdot R\Lambda R^{-1} \cdot \hat{Q}_kD_1^kD_2 \end{aligned}$$

  显然 $R\Lambda R^{-1}$ 是一个对角元为 $\lambda_1, \dots, \lambda_n$ 的上三角阵.
  根据 $\lim\limits_{k\to\infty} \hat{Q}_k = I_n$ 可知 $\hat{Q}_k^{\mathrm{H}} \cdot R\Lambda R^{-1} \cdot \hat{Q}_k \to R\Lambda R^{-1}$ $(k \to \infty)$
  而对于任意 k，$D_2^{\mathrm{H}}(D_1^{\mathrm{H}})^k \cdot R\Lambda R^{-1} \cdot D_1^kD_2$ 都仍是对角元为 $\lambda_1, \dots, \lambda_n$ 的上三角阵.
  因此 $A_k = [a_{ij}^{(k)}]$ 的严格下三角元都趋于零，且对角元 $a_{ii}^{(k)}$ 趋向于 λ_i $(i = 1, \dots, n)$
  ($\{A_k\}$ 的严格上三角元不一定收敛，因为 $\{D_1^k\}$ 的极限一般不存在)
  命题得证.

5.3.2 实 Schur 标准型

在实际应用中，大量的特征值问题都是关于实方阵的，
我们自然希望设计只涉及实数运算的 QR 算法:

$$\begin{gathered} \text{Given } A_0 = A \in \mathbb{R}^{n\times n} \\ \hline A_{k-1} = Q_kR_k \\ A_k = R_kQ_k \end{gathered}$$

其中 $Q_k \in \mathbb{R}^{n\times n}$ 是正交矩阵 $Q_k^{\mathrm{T}}Q_k = I$，$R_k \in \mathbb{R}^{n\times n}$ 是具有非负对角元的上三角阵.

然而，此时由于复共轭特征值的存在，我们不再期望序列 $\{A_k\}$ 仍然逼近一个上三角阵.
那么 $\{A_k\}$ 会趋向于什么呢?

(实 Schur 分解, Matrix Analysis 定理 2.3.4)
任意给定一个实方阵 $A \in \mathbb{R}^{n\times n}$

- ① 存在一个实的非奇异阵 $S \in \mathbb{R}^{n\times n}$ 使得 $S^{-1}AS$ 是实的拟上三角阵 (对角块都是 1×1 或 2×2 的):

$$S^{-1}AS = \begin{bmatrix} A_1 & * & \cdots & * \\ & A_2 & \ddots & \vdots \\ & & \ddots & * \\ & & & A_p \end{bmatrix}$$

  - 它的 1×1 对角块给出 A 的实特征值
  - 它的 2×2 对角块 $\begin{bmatrix} \alpha & -\beta \\ \beta & \alpha \end{bmatrix}$ 给出 A 的一对共轭的复特征值 $\alpha \pm i\beta$
  - 它的对角块由 A 的特征值完全确定，并且可以按照任意预先指定的次序出现.
- ② 存在一个实正交阵 $Q \in \mathbb{R}^{n\times n}$ 使得 Q^TAQ 是实的拟上三角阵 (对角块都是 1×1 或 2×2 的):

$$Q^{\mathrm{T}}AQ = \begin{bmatrix} A_1 & * & \cdots & * \\ & A_2 & \ddots & \vdots \\ & & \ddots & * \\ & & & A_p \end{bmatrix}$$

  - 它的 1×1 对角块给出 A 的实特征值
  - 它的 2×2 对角块不一定有特殊的形式 (但和形如 $\begin{bmatrix} \alpha & -\beta \\ \beta & \alpha \end{bmatrix}$ 的矩阵相似) 给出 A 的一对共轭的复特征值 $\alpha \pm i\beta$
  - 它的对角块由 A 的特征值完全确定，并且可以按照任意预先指定的次序出现.

我们不难想到，只涉及实数运算的 QR 算法产生的序列 $\{A_k\}$ 应当逼近实方阵 A 的实 Schur 标准型.
在下面的讨论中，如无特别说明，我们默认给定的方阵 A 是实方阵.

此外，上述迭代格式对于实际应用是没有竞争力的:
一是每次迭代的运算量太大，二是收敛速度太慢.
我们将针对这两点对其进行改进，使之成为一种高效的方法.

5.3.3 上 Hessenberg 分解

为减少每次迭代的运算量，我们事先将实方阵 A 经相似变换约化为**拟上三角阵 (上 Hessenberg 阵)**
再对约化后的矩阵进行只涉及实数运算的 QR 迭代.

设 $A \in \mathbb{R}^{n\times n}$，我们希望得到一个非奇异矩阵 $S \in \mathbb{R}^{n\times n}$ 使得相似变换后的矩阵 $S^{-1}AS$ 具有某种特殊形式.
一个自然的想法是让 $S^{-1}AS$ 的零元素越多越好.
本节我们考虑用 Householder 变换 (形如 $H := I - 2ww^{\mathrm{T}} \in \mathbb{R}^{n\times n}$ 的对称正交阵) 对 A 进行约化.

第一步，我们自然选取 Householder 变换 H_1，使得 H_1A 的第 1 列有尽可能多的零元素 (至多 $n-1$ 个).
为保证对 A 进行 (正交) 相似变换，我们需对行、列进行相同的变换，相似变换后的矩阵为 $H_1^{\mathrm{T}}AH_1 = H_1AH_1$
为避免 H_1A 右乘 H_1 时第 1 列已构建的零元素被破坏，H_1 应形如 $H_1 = \begin{bmatrix} 1 & \\ & \hat{H}_1 \end{bmatrix}$

因此 H_1 只能保证 H_1A 第 1 列的第 3 个至第 n 个元素为零，
相应地，H_1AH_1 的第 1 列的第 3 个至第 n 个元素为零，即形如:

$$H_1A = \begin{bmatrix} 1 & \\ & \hat{H}_1 \end{bmatrix}\begin{bmatrix} a_{11} & a_2^{\mathrm{T}} \\ a_1 & A_{22} \end{bmatrix} = \begin{bmatrix} a_{11} & a_2^{\mathrm{T}} \\ \hat{H}_1a_1 & \hat{H}_1A_{22} \end{bmatrix} = \left[\begin{array}{c|cccc} * & * & * & * & \cdots & * \\ \hline * & * & * & * & \cdots & * \\ 0 & * & * & * & \cdots & * \\ 0 & * & * & * & \cdots & * \\ \vdots & \vdots & \vdots & \vdots & \vdots & \vdots \\ 0 & * & * & * & \cdots & * \end{array}\right]$$

$$H_1AH_1 = (H_1A)H_1 = \begin{bmatrix} a_{11} & a_2^{\mathrm{T}} \\ \hat{H}_1a_1 & \hat{H}_1A_{22} \end{bmatrix}\begin{bmatrix} 1 & \\ & \hat{H}_1 \end{bmatrix} = \begin{bmatrix} a_{11} & a_2^{\mathrm{T}}\hat{H}_1 \\ \hat{H}_1a_1 & \hat{H}_1A_{22}\hat{H}_1 \end{bmatrix} = \left[\begin{array}{c|cccc} * & * & * & * & \cdots & * \\ \hline * & * & * & * & \cdots & * \\ 0 & * & * & * & \cdots & * \\ 0 & * & * & * & \cdots & * \\ \vdots & \vdots & \vdots & \vdots & \vdots & \vdots \\ 0 & * & * & * & \cdots & * \end{array}\right]$$

我们对 $a_1 = \begin{bmatrix} a_{21} \\ a_{31} \\ \vdots \\ a_{n1} \end{bmatrix}$ 计算 Householder 变换 $\widetilde{H}_1$，使得 $\widetilde{H}_1a_1 = \alpha\begin{bmatrix} 1 \\ 0 \\ \vdots \\ 0 \end{bmatrix} = \alpha e_1$ (其中 e_1 是 $\mathbb{R}^{n-1}$ 的第 1 个标准单位基向量)

> 回忆 Householder 变换的构造方法:
> **(数值线性代数, 定理 3.2.2)**
> 若 $x \neq 0_n \in \mathbb{R}^n$，则可构造 $\begin{cases} \alpha = \pm\|x\|_2 \\ w = \frac{x-\alpha e_1}{\|x-\alpha e_1\|_2} \\ H = I - 2ww^{\mathrm{T}} \end{cases}$ 使得 $Hx = \alpha e_1$ (其中 e_1 是 $\mathbb{R}^n$ 的第 1 个标准单位基向量)

对 $\widetilde{A}_{22} = \widetilde{H}_1 A_{22} \widetilde{H}_1 \in \mathbb{R}^{(n-1)\times(n-1)}$ 进行相同的考虑，
又可找到 Householder 变换 $\widetilde{H}_2 = \begin{bmatrix} 1 & \\ & \hat{H}_2 \end{bmatrix} \in \mathbb{R}^{(n-1)\times(n-1)}$ 使得:

$$\widetilde{H}_2 \widetilde{A}_{22} \widetilde{H}_2 = \widetilde{H}_2 (\widetilde{H}_1 A_{22} \widetilde{H}_1) \widetilde{H}_2 = \left[\begin{array}{c|cccc} * & * & * & \cdots & * \\ \hline * & * & * & \cdots & * \\ 0 & * & * & \cdots & * \\ \vdots & \vdots & \vdots & \vdots & \vdots \\ 0 & * & * & \cdots & * \end{array}\right]$$

令 $H_2 = \begin{bmatrix} 1 & \\ & \widetilde{H}_2 \end{bmatrix} = \left[\begin{array}{c|c} 1 & \\ \hline & \hat{H}_2 \end{array}\right] = \begin{bmatrix} I_2 & \\ & \hat{H}_2 \end{bmatrix} \in \mathbb{R}^{n\times n}$，则我们有:

$$H_2 (H_1 A H_1) H_2 = \left[\begin{array}{c|c|ccc} * & * & * & * & \cdots & * \\ \hline * & * & * & * & \cdots & * \\ \hline 0 & * & * & * & \cdots & * \\ 0 & 0 & * & * & \cdots & * \\ \vdots & \vdots & \vdots & \vdots & \vdots & \vdots \\ 0 & 0 & * & * & \cdots & * \end{array}\right]$$

如此进行 $n-2$ 步，就可找到 $n-2$ 个 Householder 变换 $H_1, \ldots, H_{n-2} \in \mathbb{R}^{n\times n}$ 使得:

$$Q^{\mathrm{T}} A Q = (H_{n-2} \cdots H_1) A (H_1 \cdots H_{n-2}) = \begin{bmatrix} h_{11} & h_{12} & h_{13} & \cdots & h_{1,n} \\ h_{21} & h_{22} & h_{23} & \cdots & h_{2,n} \\ & h_{32} & h_{33} & \cdots & h_{3,n} \\ & & \ddots & \ddots & \vdots \\ & & & h_{n,n-1} & h_{nn} \end{bmatrix} \overset{\Delta}{=} H$$

其中 $Q = H_1 \cdots H_{n-2} \in \mathbb{R}^{n\times n}$ 是正交阵.
我们称 $Q^{\mathrm{T}} A Q = H$ (即 $A = QHQ^{\mathrm{H}}$) 为实方阵 A 的**上 Hessenberg 分解**
(Householder 变换法计算上 Hessenberg 分解, 数值线性代数, 算法 6.4.1)

Given $A \in \mathbb{R}^{n\times n}$
for $k = 1 : n-2$
  $[v, \beta] = \mathrm{Householder}(A(k+1:n, k))$
  $A(k+1:n, k:n) = (I_{n-k} - \beta v v^{\mathrm{T}}) A(k+1:n, k:n) = A(k+1:n, k:n) - (\beta v)(v^{\mathrm{T}} A(k+1:n, k:n))$
  $A(1:n, k+1:n) = A(1:n, k+1:n)(I_{n-k} - \beta v v^{\mathrm{T}}) = A(1:n, k+1:n) - (A(1:n, k+1:n)v)(\beta v)^{\mathrm{T}}$
end
$H = A$

这一算法计算出的上 Hessenberg 矩阵就存放在 A 所对应的存储单元内，运算量为 $\frac{10}{3}n^3$
(注意其中将一次矩阵-矩阵乘法替换为一次矩阵-向量乘法和一次向量-向量乘法的技巧)
如果需要计算 $Q_0 = H_1 \cdots H_{n-2}$，则还需增加 $\frac{4}{3}n^3$ 的运算量 **(下面的算法实现存疑)**:

Given $A \in \mathbb{R}^{n\times n}$
$Q = I_n$
for $k = 1 : n-2$
  $[v, \beta] = \mathrm{Householder}(A(k+1:n, k))$
  $A(k+1:n, k:n) = (I_{n-k} - \beta v v^{\mathrm{T}}) A(k+1:n, k:n) = A(k+1:n, k:n) - (\beta v)(v^{\mathrm{T}} A(k+1:n, k:n))$
  $A(1:n, k+1:n) = A(1:n, k+1:n)(I_{n-k} - \beta v v^{\mathrm{T}}) = A(1:n, k+1:n) - (A(1:n, k+1:n)v)(\beta v)^{\mathrm{T}}$
  $Q(1:n, k+1:n) = Q(1:n, k+1:n)(I_{n-k} - \beta v v^{\mathrm{T}}) = Q(1:n, k+1:n) - (Q(1:n, k+1:n)v)(\beta v)^{\mathrm{T}}$
end
$H = A$

误差分析:
上述算法的计算解 $\hat{H}$ 满足 $\begin{cases} \hat{H} = Q^{\mathrm{T}}(A + \Delta A)Q \\ \|\Delta A\|_{\mathrm{F}} \leq cn^2 \|A\|_{\mathrm{F}} \cdot \mathrm{eps} \end{cases}$ (其中 Q 是正交阵，c 是常数，eps 是机器精度)
计算 Hessenberg 型的算法向前不稳定，但向后稳定.

> 回忆起计算 Householder 变换的算法:
> **(计算 Householder 变换, 数值线性代数, 算法 3.2.1)**

$$
\begin{aligned}
&\text{function: } [v,\beta]=\text{Householder}(x)\\
&\quad n=\text{length}(x)\\
&\quad x=\frac{x}{\|x\|_\infty}\\
&\quad v(2:n)=x(2:n)\\
&\quad (\text{下面确定 } x_1 \text{ 和 } \beta)\\
&\quad \sigma=x(2:n)^{\mathrm{T}}x(2:n)\\
&\quad \text{if } \sigma=0\\
&\qquad \beta=0\\
&\quad \text{else}\\
&\qquad \alpha=\sqrt{x(1)^2+\sigma}\\
&\qquad \text{if } x(1)>0 \quad (\text{规避相消})\\
&\qquad\quad v(1)=-\frac{\sigma}{x(1)+\alpha}\\
&\qquad \text{else} \quad (x(1)\le 0 \text{ 时无需规避相消})\\
&\qquad\quad v(1)=x(1)-\alpha\\
&\qquad \text{end}\\
&\qquad \beta=\frac{2v(1)^2}{v(1)^2+\sigma}\\
&\qquad v=\frac{v}{v(1)}\\
&\quad \text{end}
\end{aligned}
$$

我们也可以用 Givens 变换将 A 约化为上 Hessenberg 阵，运算量通常是 Householder 方法的两倍.
但如果 A 有较多零元素，则通过适当安排 Givens 变换的次序，可使运算量大为减少.

为减少计算量，我们也可以使用 Gauss 变换将 A 约化为上 Hessenberg 阵，代价是数值稳定性降低.

一般来说，上 Hessenberg 分解是不唯一的，但我们有如下结论:
(数值线性代数, 定理 $6.4.3$)
设 $A\in\mathbb{R}^{n\times n}$ 有两个上 Hessenberg 分解 $\begin{cases}(Q^{(1)})^{\mathrm{T}}AQ^{(1)}=H^{(1)}\\(Q^{(2)})^{\mathrm{T}}AQ^{(2)}=H^{(2)}\end{cases}$
其中 $Q^{(1)}=[q_1^{(1)},\dots,q_n^{(1)}]$ 和 $Q^{(2)}=[q_1^{(2)},\dots,q_n^{(2)}]$ 是 n 阶正交矩阵，
而 $H^{(1)}=[h_{ij}^{(1)}]$ 和 $H^{(2)}=[h_{ij}^{(2)}]$ 是上 Hessenberg 矩阵.

若 $q_1^{(1)}=q_1^{(2)}$ 且 $H^{(1)}$ 是**不可约的** (即其次对角元 $h_{i+1,i}^{(1)}$ 均不为零)，
则存在对角元均属于 $\{+1,-1\}$ 的对角阵 D 使得 $\begin{cases}Q^{(1)}=Q^{(2)}D\\H^{(1)}=DH^{(2)}D\end{cases}$ (即仅在正负号上有区别)

- 上述定理表明:
  若 $Q^{\mathrm{T}}AQ=H$ 是不可约的上 Hessenberg 矩阵，其中 Q 是正交矩阵，
  则在不考虑正负号变动的意义下，Q 和 H 完全由 Q 的第一列 q_1 确定.
- 这是合理的，因为 $A[q_1,q_2,\dots,q_n]=[q_1,q_2,\dots,q_n]H$ (类似于 Arnoldi 过程，只不过没有秩一项了)
  无论如何，q_2 正交于 q_1 和 Aq_1，因此方向基本上确定了 (除了在正负号上有区别)
  以此类推，可知 $Q=[q_1,q_2,\dots,q_n]$ 完全由 q_1 确定 (除了在正负号上有区别)

5.3.4 上 Hessenberg 阵的显式 QR 迭代

设 $H\in\mathbb{R}^{n\times n}$ 是上 Hessenberg 矩阵，考虑对 H 进行一次 QR 迭代.

第一步是计算 H 的 QR 分解，这步可由 $n-1$ 个 Givens 变换来完成.
为让讨论更加直观，考虑 $n=5$ 的情形:

$$H=\begin{bmatrix}\underline{*} & * & * & * & *\\ \underline{*} & * & * & * & *\\ 0 & * & * & * & *\\ 0 & 0 & * & * & *\\ 0 & 0 & 0 & * & *\end{bmatrix}$$

$$G_{1,2}H=\begin{bmatrix}\underline{*} & * & * & * & *\\ \underline{0} & * & * & * & *\\ 0 & * & * & * & *\\ 0 & 0 & * & * & *\\ 0 & 0 & 0 & * & *\end{bmatrix}=\begin{bmatrix}* & * & * & * & *\\ 0 & \underline{*} & * & * & *\\ 0 & \underline{*} & * & * & *\\ 0 & 0 & * & * & *\\ 0 & 0 & 0 & * & *\end{bmatrix}$$

$$G_{2,3}(G_{1,2}H)=\begin{bmatrix}* & * & * & * & *\\ 0 & \underline{*} & * & * & *\\ 0 & \underline{0} & * & * & *\\ 0 & 0 & * & * & *\\ 0 & 0 & 0 & * & *\end{bmatrix}=\begin{bmatrix}* & * & * & * & *\\ 0 & * & * & * & *\\ 0 & 0 & \underline{*} & * & *\\ 0 & 0 & \underline{*} & * & *\\ 0 & 0 & 0 & * & *\end{bmatrix}$$

$$G_{3,4}(G_{2,3}G_{1,2}H)=\begin{bmatrix}* & * & * & * & *\\ 0 & * & * & * & *\\ 0 & 0 & \underline{*} & * & *\\ 0 & 0 & \underline{0} & * & *\\ 0 & 0 & 0 & * & *\end{bmatrix}=\begin{bmatrix}* & * & * & * & *\\ 0 & * & * & * & *\\ 0 & 0 & * & * & *\\ 0 & 0 & 0 & \underline{*} & *\\ 0 & 0 & 0 & \underline{*} & *\end{bmatrix}$$

$$G_{4,5}(G_{3,4}G_{2,3}G_{1,2}H)=\begin{bmatrix}* & * & * & * & *\\ 0 & * & * & * & *\\ 0 & 0 & * & * & *\\ 0 & 0 & 0 & \underline{*} & *\\ 0 & 0 & 0 & \underline{0} & *\end{bmatrix}\overset{\Delta}{=}R$$

不难看出，对于一般的 n 阶上 Hessenberg 矩阵 H，
我们可以确定 $n-1$ 个 Givens 变换 $G_{1,2},\dots,G_{n-1,n}$ 使得 $G_{n-1,n}\cdots G_{1,2}H=R$
令正交阵 $Q=(G_{n-1,n}\cdots G_{1,2})^{-1}=(G_{n-1,n}\cdots G_{1,2})^{\mathrm{T}}=G_{1,2}^{\mathrm{T}}\cdots G_{n-1,n}^{\mathrm{T}}$
即得到 H 的 QR 分解 $H=QR$
具体算法如下:

Given Hessenberg matrix $H\in\mathbb{R}^{n\times n}$

$Q=I_n$
for $k=1:n-1$
  $[c,s]=\mathrm{Givens}(H(k,k),H(k+1,k))$
  $H(k:k+1,k:n)=\begin{bmatrix}c & s\\ -s & c\end{bmatrix}H(k:k+1,k:n)\quad\left(\begin{cases}H(k,k:n)=cH(k,k:n)+sH(k+1,k:n)\\ H(k+1,k:n)=-sH(k,k:n)+cH(k+1,k:n)\end{cases}\right)$
  $Q(1:n,k:k+1)=Q(1:n,k:k+1)\begin{bmatrix}c & s\\ -s & c\end{bmatrix}^{\mathrm{T}}\quad\left(\begin{cases}Q(1:n,k)=cQ(1:n,k)+sQ(1:n,k+1)\\ Q(1:n,k+1)=-sQ(1:n,k)+cQ(1:n,k+1)\end{cases}\right)$
end
$R=H$

> **(计算 Givens 变换, 数值线性代数, 算法 3.2.2)**
>
> function: $[c,s]=\mathrm{Givens}(a,b)$
> if $b=0$
> $c=1;\ s=0$
> else
> if $|b|>|a|$
> $t=\dfrac{a}{b};\ s=\dfrac{1}{\sqrt{1+t^2}};\ c=st$
> else
> $t=\dfrac{b}{a};\ c=\dfrac{1}{\sqrt{1+t^2}};\ s=ct$
> end
> end

得到 H 的 QR 分解 $H=QR$ 后，第二步便是计算 $\widetilde{H}=RQ$

$$\begin{cases}R=G_{n-1,n}\cdots G_{1,2}H\\ Q=G_{1,2}^{\mathrm{T}}\cdots G_{n-1,n}^{\mathrm{T}}\end{cases}\Rightarrow\ \widetilde{H}=RQ=RG_{1,2}^{\mathrm{T}}\cdots G_{n-1,n}^{\mathrm{T}}$$

可以验证 $\widetilde{H}$ 是一个新的上 Hessenberg 矩阵.
为让讨论更加直观，考虑 $n=5$ 的情形:

$$R=\begin{bmatrix}*&*&*&*&*\\ \underline{0}&\underline{*}&*&*&*\\ 0&0&*&*&*\\ 0&0&0&*&*\\ 0&0&0&0&*\end{bmatrix}$$

$$RG_{1,2}^{\mathrm{T}}=\begin{bmatrix}*&*&*&*&*\\ \underline{*}&\underline{*}&*&*&*\\ 0&0&*&*&*\\ 0&0&0&*&*\\ 0&0&0&0&*\end{bmatrix}=\begin{bmatrix}*&*&*&*&*\\ *&*&*&*&*\\ 0&\underline{0}&\underline{*}&*&*\\ 0&0&0&*&*\\ 0&0&0&0&*\end{bmatrix}$$

$$(RG_{1,2}^{\mathrm{T}})G_{2,3}^{\mathrm{T}}=\begin{bmatrix}*&*&*&*&*\\ *&*&*&*&*\\ 0&\underline{*}&\underline{*}&*&*\\ 0&0&0&*&*\\ 0&0&0&0&*\end{bmatrix}=\begin{bmatrix}*&*&*&*&*\\ *&*&*&*&*\\ 0&*&*&*&*\\ 0&0&\underline{0}&\underline{*}&*\\ 0&0&0&0&*\end{bmatrix}$$

$$(RG_{1,2}^{\mathrm{T}}G_{2,3}^{\mathrm{T}})G_{3,4}^{\mathrm{T}}=\begin{bmatrix}*&*&*&*&*\\ *&*&*&*&*\\ 0&*&*&*&*\\ 0&0&\underline{*}&\underline{*}&*\\ 0&0&0&0&*\end{bmatrix}=\begin{bmatrix}*&*&*&*&*\\ *&*&*&*&*\\ 0&*&*&*&*\\ 0&0&*&*&*\\ 0&0&0&\underline{0}&\underline{*}\end{bmatrix}$$

$$(RG_{1,2}^{\mathrm{T}}G_{2,3}^{\mathrm{T}}G_{3,4}^{\mathrm{T}})G_{4,5}^{\mathrm{T}}=\begin{bmatrix}*&*&*&*&*\\ *&*&*&*&*\\ 0&*&*&*&*\\ 0&0&*&*&*\\ 0&0&0&\underline{*}&\underline{*}\end{bmatrix}\overset{\Delta}{=}\widetilde{H}$$

将上述两步合并，即为上 Hessenberg 矩阵的一次 QR 迭代.

$$R=G_{n-1,n}\cdots G_{1,2}H\overset{\Delta}{=}Q^{\mathrm{T}}H$$
$$\widetilde{H}=RQ=RG_{1,2}^{\mathrm{T}}\cdots G_{n-1,n}^{\mathrm{T}}$$

具体算法如下 **(两个循环是否可以合并? 存疑)**:
(似乎并不能合并，这是显式 QR 迭代，计算 QR 分解完成后才能计算 RQ 乘积得到新的上 Hessenberg 矩阵)

Given Hessenberg matrix $H\in\mathbb{R}^{n\times n}$

$Q=I_n$
$G=\mathrm{Zeros}(n-1,2)$
for $k=1:n-1$
  $[c,s]=\mathrm{Givens}(H(k,k),H(k,k+1))$
  $G(k,1:2)=[c,s]$
  $H(k:k+1,k:n)=\begin{bmatrix}c&s\\-s&c\end{bmatrix}H(k:k+1,k:n)$
  $Q(1:n,k:k+1)=Q(1:n,k:k+1)\begin{bmatrix}c&s\\-s&c\end{bmatrix}^{\mathrm{T}}$
end
$R=H$

---

for $k=1:n-1$
  $[c,s]=G(k,1:2)$
  $H(1:k+1,k:k+1)=H(1:k+1,k:k+1)\begin{bmatrix}c&s\\-s&c\end{bmatrix}^{\mathrm{T}}$
end
$\widetilde{H}=H$

上 Hessenberg 矩阵的一次 QR 迭代的运算量为 $O(n^2)$
而对于一般的方阵，一次 QR 迭代的运算量为 $O(n^3)$

5.3.5 带位移的 QR 迭代

QR 算法是线性收敛的 (每次迭代的前后误差是成比例的)，收敛速度取决于特征值的分离程度.
为加速收敛 (受反幂法的启发)，我们可引入位移.
设第 k 步迭代的位移为 μ_k，则带位移的 QR 迭代格式为:

$$\frac{\text{Given Hessenberg matrix } H_0=H\in\mathbb{R}^{n\times n}}{\begin{aligned}H_k-\mu_kI&=Q_kR_k\\ H_{k+1}&=R_kQ_k+\mu_kI\end{aligned}}$$

根据反幂法的经验，我们可选取位移 $\mu_k=h_{n,n}^{(k)}$ (即 H_k 的第 n 个对角元)
可以证明:
当第 $n-1$ 个次对角元 $h_{n,n-1}^{(k)}$ 很小时，经一次位移 $\mu_k=h_{n,n}^{(k)}$ 的 QR 迭代后，有 $h_{n,n-1}^{(k+1)}=O((h_{n,n-1}^{(k)})^2)$ 成立.

- **证明:**
  取位移因子 $\mu_k = h_{n,n}^{(k)}$
  假定 $H_k - \mu_k I = H_k - h_{n,n}^{(k)} I$ 已经过 $n-2$ 次 Givens 变换，还差 1 次 Givens 变换就能约化为上三角阵.
  考虑此时右下角 2×2 的子矩阵 $\begin{bmatrix} \widetilde{h}_{n-1,n-1}^{(k)} & \widetilde{h}_{n-1,n}^{(k)} \\ h_{n,n-1}^{(k)} & h_{n,n}^{(k)} - h_{n,n}^{(k)} \end{bmatrix} = \begin{bmatrix} \widetilde{h}_{n-1,n-1}^{(k)} & \widetilde{h}_{n-1,n}^{(k)} \\ h_{n,n-1}^{(k)} & 0 \end{bmatrix}$
  (注意前 $n-2$ 次 Givens 变换会改变 $H_k - h_{n,n}^{(k)} I$ 的第 $n-1$ 行，但不会影响第 n 行)
  第 $n-1$ 次 Givens 变换需要消去 $h_{n,n-1}^{(k)}$
  即需确定 $c = \cos(\theta)$ 和 $s = \sin(\theta)$ 使得 $\begin{bmatrix} c & s \\ -s & c \end{bmatrix} \begin{bmatrix} \widetilde{h}_{n-1,n-1}^{(k)} \\ h_{n,n-1}^{(k)} \end{bmatrix} = \begin{bmatrix} \rho \\ 0 \end{bmatrix}$
  解得:

$$c = \frac{h_{n-1,n-1}^{(k)}}{\rho} \qquad s = \frac{h_{n,n-1}^{(k)}}{\rho} \qquad \rho = \sqrt{(\widetilde{h}_{n-1,n-1}^{(k)})^2 + (h_{n,n-1}^{(k)})^2}$$

  根据 QR 迭代的过程可知:
  新的上 Hessenberg 矩阵 H_{k+1} 的第 $n-1$ 个次对角元 $h_{n,n-1}^{(k+1)}$
  就是矩阵乘积 $\begin{bmatrix} c & s \\ -s & c \end{bmatrix} \begin{bmatrix} \widetilde{h}_{n-1,n-1}^{(k)} & \widetilde{h}_{n-1,n}^{(k)} \\ h_{n,n-1}^{(k)} & 0 \end{bmatrix} \begin{bmatrix} c & s \\ -s & c \end{bmatrix}^{\mathrm{T}}$ 的 $(2,2)$ 位置上的元素，即有:

$$\begin{aligned}
h_{n,n-1}^{(k+1)} &= (-s \cdot \widetilde{h}_{n-1,n-1}^{(k)} + c h_{n,n-1}^{(k)}) \cdot c + (-s \cdot \widetilde{h}_{n-1,n}^{(k)}) \cdot s \\
&\quad \left(\text{note that } \begin{bmatrix} c & s \\ -s & c \end{bmatrix} \begin{bmatrix} \widetilde{h}_{n-1,n-1}^{(k)} \\ h_{n,n-1}^{(k)} \end{bmatrix} = \begin{bmatrix} \rho \\ 0 \end{bmatrix} \Rightarrow \ -s \cdot \widetilde{h}_{n-1,n-1}^{(k)} + c h_{n,n-1}^{(k)} = 0\right) \\
&= 0 \cdot c - s \cdot \widetilde{h}_{n-1,n}^{(k)} \cdot s \\
&= -s^2 \widetilde{h}_{n-1,n}^{(k)} \\
&= -\frac{(h_{n,n-1}^{(k)})^2}{(\widetilde{h}_{n-1,n-1}^{(k)})^2 + (h_{n,n-1}^{(k)})^2} \widetilde{h}_{n-1,n}^{(k)} \quad (\text{由于 } h_{n,n-1}^{(k)} \text{趋于零，故可假设 } |h_{n,n-1}^{(k)}| \text{ 远小于 } |\widetilde{h}_{n-1,n-1}^{(k)}| \text{ 和 } |\widetilde{h}_{n-1,n}^{(k)}|) \\
&\approx -\frac{\widetilde{h}_{n-1,n}^{(k)}}{(\widetilde{h}_{n-1,n-1}^{(k)})^2} (h_{n,n-1}^{(k)})^2 \\
&= O((h_{n,n-1}^{(k)})^2))
\end{aligned}$$

这表明通过取位移因子 $\mu_k = h_{n,n}^{(k)}$，可让特征值的渐近收敛速度从线性收敛加速为二次收敛.

邵老师提供的观点:
假设对 $H_k - \mu_k I$ 进行 QR 分解得到的上三角阵为:

$$H_k - \mu_k I = Q_k R_k$$

$$R_k = \begin{bmatrix} * & * & * & * \\ 0 & * & * & * \\ & 0 & * & * \\ & & 0 & \varepsilon \end{bmatrix}$$

其中 ε 是一个模很小的小量.
最后得到 $H_{k+1} = R_k Q_k + \mu_k I$:

$$H_{k+1} = R_k Q_k + \mu_k I = \begin{bmatrix} * & * & * & * \\ + & * & * & * \\ & + & * & * \\ & & \varepsilon_1 & \varepsilon_2 + \mu_k \end{bmatrix}$$

其中 $\varepsilon_1, \varepsilon_2$ 都是模很小的小量.
根据 Gershgorin 圆盘定理可知，
会有一个特征值落在以 $\varepsilon_2 + \mu_k$ 为中心，以 $|\varepsilon_1|$ 为半径的圆盘中 (如果该圆盘不与其他圆盘连通的话)
因此将 $\varepsilon_2 + \mu_k$ 当作特征值是合理的.

特征向量的迭代:
若取 $\mu_k = h_{n,n}^{(k)}$，则相当于 Rayleigh 商迭代:

$$\mu_k = h_{n,n}^{(k)} = [0, \ldots, 0, 1] H_k \begin{bmatrix} 0 \\ \vdots \\ 0 \\ 1 \end{bmatrix} = e_n^{\mathrm{T}} H_k e_n$$

近似特征向量可以一直视为 $e_n = [0, \ldots, 0, 1]^{\mathrm{T}}$
它近似 $h_{n,n}^{(k+1)}$ 的特征向量的程度要好于近似 $h_{n,n}^{(k)}$ 的特征向量的程度 (二次收敛)

设 H 是上 Hessenberg 矩阵，设位移 $\mu = h_{n,n}$
若视 $h_{n,n-1} \approx 0$，则我们有:

$$(H-\mu I)^{\mathrm{T}}e_n \approx e_n \cdot 0$$

$$\begin{bmatrix} * & * & & \\ * & * & * & \\ * & * & * & * \\ * & * & * & 0 \end{bmatrix}\begin{bmatrix} 0 \\ 0 \\ 0 \\ 1 \end{bmatrix} \approx \begin{bmatrix} 0 \\ 0 \\ 0 \\ 1 \end{bmatrix} \cdot 0$$

这表明 e_n 是近似特征值 $\mu = h_{n,n}$ 的近似左特征向量，即 $e_n^{\mathrm{T}}H \approx \mu e_n^{\mathrm{T}}$
我们现在想要将近似做得更好 (对 $H-\mu I$ 做 QR 分解来解线性方程组)

$$\begin{aligned} x^{\mathrm{T}} &= e_n^{\mathrm{T}}(H-\mu I)^{-1} \\ &= e_n^{\mathrm{T}}(QR)^{-1} \\ &= e_n^{\mathrm{T}}R^{-1}Q^{\mathrm{T}} \quad (\text{note that } e_n^{\mathrm{T}}R^{-1} = \alpha e_n^{\mathrm{T}} \text{ for some } \alpha) \\ &= \alpha e_n^{\mathrm{T}}Q^{\mathrm{T}} \end{aligned}$$

因此 $e_n^{\mathrm{T}}Q^{\mathrm{T}}$ 也是 H 的一个近似左特征向量，满足 $e_n^{\mathrm{T}}Q^{\mathrm{T}}H \approx \mu e_n^{\mathrm{T}}Q^{\mathrm{T}}$
现在记 $\hat{H} = Q^{\mathrm{T}}HQ$，则我们有:

$$\begin{aligned} e_n^{\mathrm{T}}\hat{H} &= e_n^{\mathrm{T}}Q^{\mathrm{T}}HQ \\ &\approx \mu e_n^{\mathrm{T}}Q^{\mathrm{T}}Q \\ &= \mu e_n^{\mathrm{T}} \end{aligned}$$

这说明 e_n^{T} 也是 $\hat{H}$ 的一个近似左特征向量，但由于 $\hat{h}_{n,n-1}$ 会比 $h_{n,n-1}$ 更小，故近似效果会更好.

5.3.6 双位移的隐式 QR 迭代

单位移的 QR 迭代存在严重的缺点:
若 A 具有复共轭特征值，则实位移一般不能起到加速作用.
为克服这一缺点，我们来介绍双位移的 QR 迭代，
其基本思想是将两步单位移的 QR 迭代合并为一步，以避免复数运算.

考察单位移的 QR 迭代格式:

$$\frac{\text{Given Hessenberg matrix } H_0 = H \in \mathbb{R}^{n\times n}}{\begin{aligned} H_k - \mu_k I &= Q_k R_k \\ H_{k+1} &= R_k Q_k + \mu_k I \end{aligned}}$$

- 假定出现的所有上 Hessenberg 阵都是不可约的.
  若不然，例如 $H_k = \begin{bmatrix} H_{11}^{(k)} & * \\ & H_{22}^{(k)} \end{bmatrix}$，则我们可以分别对 $H_{11}^{(k)}$ 和 $H_{22}^{(k)}$ 进行 QR 迭代.

实矩阵可以有复特征值.
如果 H_k 的右下角的 2×2 子矩阵 $\hat{H}_k = \begin{bmatrix} h_{n-1,n-1}^{(k)} & h_{n-1,n}^{(k)} \\ h_{n,n-1}^{(k)} & h_{n,n}^{(k)} \end{bmatrix}$ 有一对复共轭特征值 $\mu_1^{(k)}, \mu_2^{(k)}$
我们就不能期望 $h_{n,n}^{(k)}$ 最终收敛到 A 的某个特征值，
此时取位移 $\mu_k = h_{n,n}^{(k)}$ 就完全起不到加速收敛的作用.

我们自然想到取 $\mu_1^{(k)}$ 或 $\mu_2^{(k)}$ 为位移来加速收敛，但这样一来就必须涉及复数运算.
为规避复数运算，我们用 $\mu_1^{(k)}$ 和 $\mu_2^{(k)}$ 连续作两次位移，迭代格式为:
(为简化记号，我们省去标记 (k)，记 H_k 为 H，记 $\mu_1^{(k)}, \mu_2^{(k)}$ 为 μ_1, μ_2)

$$\begin{aligned} H - \mu_1 I &= U_1 R_1 \\ H_1 &= R_1 U_1 + \mu_1 I \\ H_1 - \mu_2 I &= U_2 R_2 \\ H_2 &= R_2 U_2 + \mu_2 I \end{aligned}$$

其中 $U_1, U_2 \in \mathbb{C}^{n\times n}$ 均是酉矩阵，并要求 R_1, R_2 的对角元均为正实数.
我们有:

$$\begin{cases} H_1 = R_1U_1 + \mu_1 I = U_1^{\mathrm{H}}(U_1R_1 + \mu_1 I)U_1 = U_1^{\mathrm{H}}HU_1 \\ H_2 = R_2U_2 + \mu_2 I = U_2^{\mathrm{H}}(U_2R_2 + \mu_2 I)U_2 = U_2^{\mathrm{H}}H_1U_2 \end{cases} \Rightarrow\ H_2 = U_2^{\mathrm{H}}H_1U_2 = U_2^{\mathrm{H}}(U_1^{\mathrm{H}}HU_1)U_2 = (U_1U_2)^{\mathrm{H}}H(U_1U_2)$$

$$\begin{aligned} (H-\mu_1 I)(H-\mu_2 I) &= U_1R_1(U_1R_1 + (\mu_1-\mu_2)I) \\ &= U_1(R_1U_1 + (\mu_1-\mu_2)I)R_1 \\ &= U_1(H_1 - \mu_1 I + (\mu_1-\mu_2)I)R_1 \\ &= U_1(H_1 - \mu_2 I)R_1 \\ &= U_1(U_2R_2)R_1 \end{aligned}$$

我们记 $\begin{cases} Q = U_1U_2 \\ R = R_2R_1 \\ M = (H-\mu_1 I)(H-\mu_2 I) \end{cases}$ 则有:

$$\begin{aligned} H_2 &= (U_1U_2)^{\mathrm{H}}H(U_1U_2) = Q^{\mathrm{H}}HQ \\ M &= (H-\mu_1 I)(H-\mu_2 I) = U_1U_2R_2R_1 = QR \end{aligned}$$

注意到 μ_1 和 μ_2 作为 H_k 的右下角的 2×2 子矩阵 $\hat{H}_k=\begin{bmatrix}h_{n-1,n-1}^{(k)} & h_{n-1,n}^{(k)}\\ h_{n,n-1}^{(k)} & h_{n,n}^{(k)}\end{bmatrix}$ 的复共轭特征值.

我们记 $\begin{cases}t=\operatorname{tr}(\hat{H}_k)=h_{n-1,n-1}^{(k)}+h_{n,n}^{(k)}=\mu_1+\mu_2\in\mathbb{R}\\ s=\det(\hat{H}_k)=h_{n-1,n-1}^{(k)}h_{n,n}^{(k)}-h_{n-1,n}^{(k)}h_{n,n-1}^{(k)}=\mu_1\mu_2\in\mathbb{R}\end{cases}$

于是 $M=(H-\mu_1 I)(H-\mu_2 I)=H^2-(\mu_1+\mu_2)H+\mu_1\mu_2 I=H^2-tH+sI$ 是实方阵.

由于 R_1,R_2 的对角元均为正实数，故 $R=R_2R_1$ 的对角元也都是正实数.
若 μ_1,μ_2 不是 H 的特征值 (这保证了实方阵 M 非奇异)，则由 $M=QR$ 可知 Q 是实正交阵.
于是根据 $H_2=Q^{\mathrm{H}}HQ=Q^{\mathrm{T}}HQ$ 可知 H_2 也是实方阵.
也就是说，在不考虑误差的情况下，用 μ_1,μ_2 连续作两次位移进行 QR 迭代得到的 H_2 也是实的上 Hessenberg 阵.

但实际计算中，由于舍入误差的影响，这样得到的 H_2 不一定是实的.
为确保计算得到的 H_2 仍是实的，我们自然想到按如下步骤计算 H_2:

$$t=\operatorname{tr}\left(\begin{bmatrix}h_{n-1,n-1}^{(k)} & h_{n-1,n}^{(k)}\\ h_{n,n-1}^{(k)} & h_{n,n}^{(k)}\end{bmatrix}\right)=h_{n-1,n-1}^{(k)}+h_{n,n}^{(k)}$$

$$s=\det\left(\begin{bmatrix}h_{n-1,n-1}^{(k)} & h_{n-1,n}^{(k)}\\ h_{n,n-1}^{(k)} & h_{n,n}^{(k)}\end{bmatrix}\right)=h_{n-1,n-1}^{(k)}h_{n,n}^{(k)}-h_{n-1,n}^{(k)}h_{n,n-1}^{(k)}$$

$$M=H^2-tH+sI$$

$$M=QR$$

$$H_2=Q^{\mathrm{T}}HQ$$

然而第三步计算 M 的运算量就是 $O(n^3)$ (这是我们不希望看到的)，
而且 M 也不再是上 Hessenberg 矩阵 (它比上 Hessenberg 矩阵还要多一条非零的次对角线)，
尽管使用 Givens 变换计算其 QR 分解 $M=QR$ 并得到 $H_2=Q^{\mathrm{T}}HQ$ 的运算量仍是 $O(n^2)$ **(存疑)**

幸运的是，回忆起**数值线性代数 定理** $6.4.3$ 的结论:

> 设 $A\in\mathbb{R}^{n\times n}$ 有两个上 Hessenberg 分解 $\begin{cases}(Q^{(1)})^{\mathrm{T}}AQ^{(1)}=H^{(1)}\\ (Q^{(2)})^{\mathrm{T}}AQ^{(2)}=H^{(2)}\end{cases}$
> 其中 $Q^{(1)}=[q_1^{(1)},\dots,q_n^{(1)}]$ 和 $Q^{(2)}=[q_1^{(2)},\dots,q_n^{(2)}]$ 是 n 阶正交矩阵，
> 而 $H^{(1)}=[h_{ij}^{(1)}]$ 和 $H^{(2)}=[h_{ij}^{(2)}]$ 是上 Hessenberg 矩阵.
>
> 若 $q_1^{(1)}=q_1^{(2)}$ 且 $H^{(1)}$ 是**不可约的** (即其次对角元 $h_{i+1,i}^{(1)}$ 均不为零)，
> 则存在对角元均属于 $\{+1,-1\}$ 的对角阵 D 使得 $\begin{cases}Q^{(1)}=Q^{(2)}D\\ H^{(1)}=DH^{(2)}D\end{cases}$ (即仅在正负号上有区别)

上述定理表明:
若 $Q^{\mathrm{T}}AQ=H$ 是不可约的上 Hessenberg 矩阵，其中 Q 是正交矩阵，
则在不考虑正负号变动的意义下，Q 和 H 完全由 Q 的第一列 q_1 确定.

也就是说，无论采用什么方法去求正交矩阵 $\widetilde{Q}$ 使得 $\widetilde{H}_2=\widetilde{Q}^{\mathrm{T}}HQ$ 为上 Hessenberg 矩阵，
只要 $\widetilde{Q}$ 的第一列与 Q 的第一列相同，
$\widetilde{H}_2$ 就与 $H_2=Q^{\mathrm{T}}HQ$ 本质上相同 (所有元素绝对值都对应相等，只是正负号可能不同)
当然，这需要 H_2 是不可约的上 Hessenberg 矩阵.
换言之，只要 H_2 是不可约的上 Hessenberg 矩阵，
我们就有很大的自由度去寻找更有效的方法来实现由 H 到 H_2 的变换.

下面的定理给出了 H_2 不可约的条件.
(数值线性代数, 定理 $6.4.4$**)**
若 H 是不可约的上 Hessenberg 矩阵，且 μ_1,μ_2 均不是 H 的特征值，
则上述迭代产生的 H_2 也是不可约的上 Hessenberg 矩阵.

我们可以通过其他途径来实现从 H 到 H_2 的变换.

首先根据 $M=QR$ 可知 Q 的第一列 Qe_1 就是 M 的第一列 Me_1 单位化得到的.
而根据 $M=H^2-tH+sI$ 可知 M 的第一列元素如下:
(我们不用计算整个 M，只需计算其第一列的前三个元素 m_{11},m_{21},m_{31} 即可)

$$Me_1=\begin{bmatrix}m_{11}\\ m_{21}\\ m_{31}\\ 0\\ \vdots\\ 0\end{bmatrix}=\begin{bmatrix}(h_{11}^{(k)})^2+h_{12}^{(k)}h_{21}^{(k)}-th_{11}^{(k)}+s\\ h_{21}^{(k)}(h_{11}^{(k)}+h_{22}^{(k)}-t)\\ h_{21}^{(k)}h_{32}^{(k)}\\ 0\\ \vdots\\ 0\end{bmatrix}$$

其次，我们计算 Householder 变换 P_0，将 M 的第一列 Me_1 的第 2 个和第 3 个分量化为零，
即使得 $P_0Me_1=\alpha e_1$ (其中 $\alpha=\|Me_1\|_2=\sqrt{m_{11}^2+m_{21}^2+m_{31}^2}$):

$$[v_0,\beta_0]=\text{Householder}(M(1:3,1))=\text{Householder}\left(\begin{bmatrix}m_{11}\\m_{21}\\m_{31}\end{bmatrix}\right)$$

$$\widetilde{P}_0=I_3-\beta_0v_0v_0^{\mathrm{T}}$$

$$P_0=\begin{bmatrix}\widetilde{P}_0 & \\ & I_{n-3}\end{bmatrix}$$

从而 P_0 的第一行的转置 (即第一列 P_0e_1) 就是 Me_1 单位化得到的向量,
因此待求正交矩阵 Q 的第一列即为 P_0 的第一列，即 $Qe_1=P_0e_1$

现令 $B=P_0HP_0^{\mathrm{T}}=P_0HP_0$
由于这个正交相似变换只改变了 H 的前三行和前三列，故 B 的形状如下:

$$B=P_0HP_0=\begin{bmatrix}\widetilde{P}_0 & \\ & I_{n-3}\end{bmatrix}\begin{bmatrix}*&*&*&*&\cdots&*&*\\*&*&*&*&\cdots&*&*\\&*&*&*&\cdots&*&*\\&&*&*&\cdots&*&*\\&&&\ddots&\ddots&\vdots&\vdots\\&&&&*&*&*\\&&&&&*&*\end{bmatrix}\begin{bmatrix}\widetilde{P}_0 & \\ & I_{n-3}\end{bmatrix}=\begin{bmatrix}\underline{*}&\underline{*}&\underline{*}&\underline{*}&\cdots&\underline{*}&\underline{*}\\\underline{*}&\underline{*}&\underline{*}&\underline{*}&\cdots&\underline{*}&\underline{*}\\+&\underline{*}&\underline{*}&\underline{*}&\cdots&\underline{*}&\underline{*}\\+&+&\underline{*}&*&\cdots&*&*\\&&&\ddots&\ddots&\vdots&\vdots\\&&&&*&*&*\\&&&&&*&*\end{bmatrix}$$

接下来我们只要能够找到第一列为 e_1 的正交矩阵 $\widetilde{Q}$ 使得 $\widetilde{H}_2=\widetilde{Q}^{\mathrm{T}}B\widetilde{Q}$ 为上 Hessenberg 矩阵，
那么 $\widetilde{H}_2$ 就是我们希望得到的 H_2

根据 5.3.3 节的上 Hessenberg 分解算法可知，这是容易办到的，
例如可以确定 $n-2$ 个 Householder 变换 $P_1,\dots,P_{n-2}$ 使得 $P_{n-2}\cdots P_1BP_1\cdots P_{n-2}=\widetilde{H}_2$
则正交矩阵 $\widetilde{Q}=P_1\cdots P_{n-2}$ 的第一行和第一列只有 $(1,1)$ 位置上的元素为 1，其余元素非零
(这是因为 $P_1,\dots,P_{n-2}$ 的第一行和第一列都只有 $(1,1)$ 位置上的元素为 1，其余元素非零，参考 5.3.3 节的内容)
因此 $\widetilde{Q}$ 的第一列自然为 e_1
由于 B 矩阵结构的稀疏性，实现这一上 Hessenberg 分解的运算量仅为 $O(n^2)$

综上所述，我们得到了著名的 **Francis 双位移的 QR 迭代算法**:
(双位移的 QR 迭代算法, 数值线性代数, 算法 $6.4.2$)

Given Hessenberg matrix $H\in\mathbb{R}^{n\times n}$

$$t=\operatorname{tr}\left(\begin{bmatrix}h_{n-1,n-1}&h_{n-1,n}\\h_{n,n-1}&h_{n,n}\end{bmatrix}\right)=h_{n-1,n-1}+h_{n,n}$$

$$s=\det\left(\begin{bmatrix}h_{n-1,n-1}&h_{n-1,n}\\h_{n,n-1}&h_{n,n}\end{bmatrix}\right)=h_{n-1,n-1}h_{n,n}-h_{n-1,n}h_{n,n-1}$$

$$\begin{cases}m_{11}=h_{11}^2+h_{12}h_{21}-th_{11}+s\\m_{21}=h_{21}(h_{11}+h_{22}-t)\\m_{31}=h_{21}h_{32}\end{cases}\quad(\text{note that } Me_1=\begin{bmatrix}m_{11}\\m_{21}\\m_{31}\\0\\\vdots\\0\end{bmatrix}=\begin{bmatrix}h_{11}^2+h_{12}h_{21}-th_{11}+s\\h_{21}(h_{11}+h_{22}-t)\\h_{21}h_{32}\\0\\\vdots\\0\end{bmatrix}\text{ where } M=H^2-tH+sI)$$

$$[v,\beta]=\text{Householder}\left(\begin{bmatrix}m_{11}\\m_{21}\\m_{31}\end{bmatrix}\right)\quad(\text{case of } k=0)$$

$$H(1:3,1:n)=(I_3-\beta vv^{\mathrm{T}})H(1:3,1:n)=H(1:3,1:n)-(\beta v)(v^{\mathrm{T}}H(1:3,1:n))$$
$$H(1:4,1:3)=H(1:4,1:3)(I_3-\beta vv^{\mathrm{T}})=H(1:4,1:3)-(H(1:4,1:3)v)(\beta v)^{\mathrm{T}}$$

---

for $k=1:n-4$
    $[v,\beta]=\text{Householder}(H(k+1:k+3,k))$
    $H(k+1:k+3,k:n)=(I_3-\beta vv^{\mathrm{T}})H(k+1:k+3,k:n)=H(k+1:k+3,k:n)-(\beta v)(v^{\mathrm{T}}H(k+1:k+3,k:n))$
    $H(1:k+4,k+1:k+3)=H(1:k+4,k+1:k+3)(I_3-\beta vv^{\mathrm{T}})=H(1:k+4,k+1:k+3)-(H(1:k+4,k+1:k+3)v)(\beta v)^{\mathrm{T}}$
end

$[v,\beta]=\text{Householder}(H(n-2:n,n-3))\quad(\text{case of } k=n-3)$
$H(n-2:n,n-3:n)=(I_3-\beta vv^{\mathrm{T}})H(n-2:n,n-3:n)=H(n-2:n,n-3:n)-(\beta v)(v^{\mathrm{T}}H(n-2:n,n-3:n))$
$H(1:n,n-2:n)=H(1:n,n-2:n)(I_3-\beta vv^{\mathrm{T}})=H(1:n,n-2:n)-(H(1:n,n-2:n)v)(\beta v)^{\mathrm{T}}$

$[v,\beta]=\text{Householder}(H(n-1:n,n-2))\quad(\text{case of } k=n-2)$
$H(n-1:n,n-2:n)=(I_2-\beta vv^{\mathrm{T}})H(n-1:n,n-2:n)=H(n-1:n,n-2:n)-(\beta v)(v^{\mathrm{T}}H(n-1:n,n-2:n))$
$H(1:n,n-1:n)=H(1:n,n-1:n)(I_2-\beta vv^{\mathrm{T}})=H(1:n,n-1:n)-(H(1:n,n-1:n)v)(\beta v)^{\mathrm{T}}$

上述算法的运算量为 $10n^2$
若需要累积正交变换，则还需再增加运算量 $10n^2$

5.3.7 隐式 QR 算法

QR 算法作为一种实用的算法，还需给出有效的收敛判定准则.
当 $|h_{i+1,i}^{(k)}|\le(|h_{i,i}^{(k)}|+|h_{i+1,i+1}^{(k)}|)\text{eps}$ 时，
我们就认为上 Hessenberg 矩阵 H_k 的次对角元 $h_{i+1,i}^{(k)}$ 为零.
(因为在最初对 $A\in\mathbb{R}^{n\times n}$ 进行上 Hessenberg 分解得到 $H_0\in\mathbb{R}^{n\times n}$ 的过程中引入了 $\|A\|_{\mathrm{F}}\cdot\text{eps}$ 量级的误差)

结合之前的讨论，我们得到隐式 QR 算法.
该算法用于计算给定实方阵 $A \in \mathbb{R}^{n\times n}$ 的实 Schur 分解 $Q^{\mathrm{T}}AQ = H$
其中 $Q \in \mathbb{R}^{n\times n}$ 是正交矩阵
而 H 是拟上三角阵 (即上 Hessenberg 矩阵)，具体来说是对角块为 1×1 或 2×2 方阵的块上三角阵，
每个 2×2 的对角块对应一对复共轭特征值.

(隐式 QR 算法计算实方阵的实 Schur 分解, 数值线性代数, 算法 $6.4.3$)

- **(1) 将给定方阵 $A \in \mathbb{R}^{n\times n}$ 上 Hessenberg 化:**

  $\mathrm{Given}\ A \in \mathbb{R}^{n\times n}$

  $Q = I_n$
  $\mathrm{for}\ k = 1 : n-2$
  $\quad [v, \beta] = \mathrm{Householder}(A(k+1:n, k))$
  $\quad A(k+1:n, k:n) = (I_{n-k} - \beta vv^{\mathrm{T}})A(k+1:n, k:n) = A(k+1:n, k:n) - (\beta v)(v^{\mathrm{T}}A(k+1:n, k:n))$
  $\quad A(1:n, k+1:n) = A(1:n, k+1:n)(I_{n-k} - \beta vv^{\mathrm{T}}) = A(1:n, k+1:n) - (A(1:n, k+1:n)v)(\beta v)^{\mathrm{T}}$
  $\quad Q(1:n, k+1:n) = Q(1:n, k+1:n)(I_{n-k} - \beta vv^{\mathrm{T}}) = Q(1:n, k+1:n) - (Q(1:n, k+1:n)v)(\beta v)^{\mathrm{T}}$
  end
  $H = A$

  得到上 Hessenberg 矩阵 $H = Q^{\mathrm{T}}AQ$

- **(2) 收敛性判定:**
  - 将所有满足条件 $|h_{i+1,i}| \le (|h_{i,i}| + |h_{i+1,i+1}|)\mathrm{eps}$ 的次对角元 $h_{i+1,i}\ (i = 1, \ldots, n-1)$ 置为零.
    (本来只需将 $\le \|A\|_2\mathrm{eps}$ 的那些次对角元置为零即可，但现实中会提这样严格的要求，而且效果相当不错)
    (这样操作是向后稳定的，因为在正交变换下，对 H 的扰动向后传递时 2 范数不变)
  - 将 H 划分为 $H = \begin{bmatrix} H_{11} & H_{12} & H_{13} \\ & H_{22} & H_{23} \\ & & H_{33} \end{bmatrix}$ (其中 $\begin{cases} H_{11} \in \mathbb{R}^{l\times l} \\ H_{22} \in \mathbb{R}^{(u-l)\times(u-l)} \\ H_{33} \in \mathbb{R}^{(n-u)\times(n-u)} \end{cases}$)
    最小化 u 使得 $H_{33} \in R^{(n-u)\times(n-u)}$ 为对角块为 1×1 或 2×2 方阵的块上三角阵
    最小化 l 使得 $H_{22} \in \mathbb{R}^{(u-l)\times(u-l)}$ 为不可约的上 Hessenberg 矩阵

    若 $u = 0$，则迭代终止; 否则进行下一步.

- **(3) 双位移的隐式 QR 迭代:**
  对 H 的 H_{22} 分块进行双位移的隐式 QR 迭代
  (但对 H_{22} 的正交变换应用于整个矩阵 H，即 H_{23} 应享受到 H_{22} 同样的行变换，而 H_{12} 应享受到 H_{22} 同样的列变换)
  注意到 H_{22} 分块由 H 的第 $l+1$ 至 u 行和第 $l+1$ 至 u 列的元素构成，
  即 $H_{22} = \begin{bmatrix} h_{l+1,l+1} & h_{l+1,l+2} & \cdots & h_{l+1,u} \\ h_{l+2,l+1} & h_{l+2,l+2} & \cdots & h_{l+2,u} \\ & \ddots & \ddots & \vdots \\ & & h_{u,u-1} & h_{uu} \end{bmatrix}$
  具体算法如下:

Given Hessenberg matrix $H \in \mathbb{R}^{n\times n}$, orthogonal matrix $Q \in \mathbb{R}^{n\times n}$ and $0 \le l < u \le n$

$t = \operatorname{tr}\left(\begin{bmatrix} h_{u-1,u-1} & h_{u-1,u} \\ h_{u,u-1} & h_{u,u} \end{bmatrix}\right) = h_{u-1,u-1} + h_{u,u}$

$s = \det\left(\begin{bmatrix} h_{u-1,u-1} & h_{u-1,u} \\ h_{u,u-1} & h_{u,u} \end{bmatrix}\right) = h_{u-1,u-1}h_{u,u} - h_{u-1,u}h_{u,u-1}$

$\begin{cases} m_{11} = h_{l+1,l+1}^2 + h_{l+1,l+2}h_{l+2,l+1} - th_{l+1,l+1} + s \\ m_{21} = h_{l+2,l+1}(h_{l+1,l+1} + h_{l+2,l+2} - t) \\ m_{31} = h_{l+2,l+1}h_{l+3,l+2} \end{cases}$

$$(\text{note that } Me_1 = \begin{bmatrix} m_{11} \\ m_{21} \\ m_{31} \\ 0 \\ \vdots \\ 0 \end{bmatrix} = \begin{bmatrix} h_{l+1,l+1}^2 + h_{l+1,l+2}h_{l+2,l+1} - th_{l+1,l+1} + s \\ h_{l+2,l+1}(h_{l+1,l+1} + h_{l+2,l+2} - t) \\ h_{l+2,l+1}h_{l+3,l+2} \\ 0 \\ \vdots \\ 0 \end{bmatrix} \text{ where } M = H_{22}^2 - tH_{22} + sI_{u-l})$$

$[v, \beta] = \text{Householder}\left(\begin{bmatrix} m_{11} \\ m_{21} \\ m_{31} \end{bmatrix}\right) \quad (\text{case of } k = 0)$

$H(l+1:l+3, l+1:n) = H(l+1:l+3, l+1:n) - (\beta v)(v^{\mathrm{T}}H(l+1:l+3, l+1:n))$

if $u - l = 3$

$\quad H(1:l+3, l+1:l+3) = H(1:l+3, l+1:l+3) - (H(1:l+3, l+1:l+3))v)(\beta v)^{\mathrm{T}}$

else

$\quad H(1:l+4, l+1:l+3) = H(1:l+4, l+1:l+3) - (H(1:l+4, l+1:l+3))v)(\beta v)^{\mathrm{T}}$

end

$Q(1:n, l+1:l+3) = Q(1:n, l+1:l+3) - (Q(1:n, l+1:l+3)v)(\beta v)^{\mathrm{T}}$

---

for $k = 1 : u - l - 4$

$\quad [v, \beta] = \text{Householder}(H(l+k+1:l+k+3, l+k))$

$\quad H(l+k+1:l+k+3, l+k:n) = H(l+k+1:l+k+3, l+k:n) - (\beta v)(v^{\mathrm{T}}H(l+k+1:l+k+3, l+k:n))$

$\quad H(1:l+k+4, l+k+1:l+k+3) = H(1:l+k+4, l+k+1:l+k+3) - (H(1:l+k+4, l+k+1:l+k+3)v)(\beta v)^{\mathrm{T}}$

$\quad Q(1:n, l+k+1:l+k+3) = Q(1:n, l+k+1:l+k+3) - (Q(1:n, l+k+1:l+k+3)v)(\beta v)^{\mathrm{T}}$

end

---

if $u - l \neq 3$

$\quad H(u-2:u, u-3:n) = H(u-2:u, u-3:n) - (\beta v)(v^{\mathrm{T}}H(u-2:u, u-3:n))$

$\quad H(1:u, u-2:u) = H(1:u, u-2:u) - (H(1:u, u-2:u)v)(\beta v)^{\mathrm{T}}$

$\quad Q(1:n, u-2:u) = Q(1:n, u-2:u) - (Q(1:n, u-2:u)v)(\beta v)^{\mathrm{T}}$

end

$[v, \beta] = \text{Householder}(H(u-1:u, u-2)) \quad (\text{case of } k = u - l - 2)$

$H(u-1:u, u-2:n) = H(u-1:u, u-2:n) - (\beta v)(v^{\mathrm{T}}H(u-1:u, u-2:n))$

$H(1:u, u-1:u) = H(1:u, u-1:u) - (H(1:u, u-1:u)v)(\beta v)^{\mathrm{T}}$

$Q(1:n, u-1:u) = Q(1:n, u-1:u) - (Q(1:n, u-1:u)v)(\beta v)^{\mathrm{T}}$

这样就完成了一次双位移的隐式 QR 迭代，上 Hessenberg 矩阵 H 和正交矩阵 Q 均得到更新.
然后转到第 (2) 步.

实际计算的经验表明:
隐式 QR 算法每分离出一个 1×1 或 2×2 子矩阵平均需要 2 次迭代，因此总运算量约为 $25n^3$
若只计算特征值 (即不累积正交矩阵 Q)，则总运算量约为 $10n^3$

最后我们需要验证:

- ① 正交性损失 $\|Q^{\mathrm{T}}Q - I\|_{\mathrm{F}}$ 机器精度级别
- ② 误差 $\frac{\|Q^{\mathrm{T}}AQ - T\|_{\mathrm{F}}}{\|A\|_{\mathrm{F}}}$ 机器精度级别
- ③ 矩阵 T 是实 Schur 型的形式 (即分块上三角阵，对角分块要么是 1×1 的，要么是 2×2 的)

5.4 实 Schur 型的后处理

5.4.1 对角块排序

冒泡排序调整特征值收敛次序:
先考虑 2×2 上三角阵的换序: (Homeork 08 Problem 04)

$$Q^{\mathrm{T}}\begin{bmatrix} a & d \\ & b \end{bmatrix}Q = \begin{bmatrix} b & \pm d \\ & a \end{bmatrix}$$

其中变换后的矩阵的 $(1,2)$ 元素之所以是 $\pm d$ 是显然的: 酉相似变换不改变矩阵的 Frobenius 范数.
我们设 Q 的形式为:

$$\begin{bmatrix} c & -s \\ s & c \end{bmatrix}^{\mathrm{T}}\begin{bmatrix} a & d \\ & b \end{bmatrix}\begin{bmatrix} c & -s \\ s & c \end{bmatrix} = \begin{bmatrix} b & d \\ & a \end{bmatrix} \ (\text{where } c^2 + s^2 = 1)$$

可以证明向前向后都是稳定的.
但是推广到 2×2 块需要解 Sylvester 方程 (为解得准需要进行全主元 Gauss 消去法)
其舍入误差分析至今都没有解决.

5.4.2 求解 Sylvester 方程

给定 $A \in \mathbb{C}^{m\times m}, B \in \mathbb{C}^{n\times n}, C \in \mathbb{C}^{m\times n}$
Sylvester 方程组 $AX - XB = C$ 可以等价表示为 $\mathrm{vec}(X) = (I_n \otimes A - B^{\mathrm{T}} \otimes I_m)^{-1}\mathrm{vec}(C)$

$$AX - XB = C$$
$$\Leftrightarrow$$
$$(I_n \otimes A - B^{\mathrm{T}} \otimes I_m)\mathrm{vec}(X) = \mathrm{vec}(C)$$

(Sylvester 定理, Matrix Analysis 定理 2.4.4.1)
Sylvester 方程组有唯一解，当且仅当系数矩阵 $(I_n \otimes A - B^{\mathrm{T}} \otimes I_m)$ 非奇异，即当且仅当 A, B 没有公共特征值.

假设 $A \in \mathbb{C}^{m,m}$ 和 $B \in \mathbb{C}^{n\times n}$ 没有公共特征值.
直接求解 $(I_n \otimes A - B^{\mathrm{T}} \otimes I_m)\mathrm{vec}(X) = \mathrm{vec}(C)$ 的代价是 $O(m^3n^3)$，难以接受.
我们可以对 A 做 Schur 分解 $A = U_1T_1U_1^{\mathrm{H}}$ (代价为 $O(m^3)$)，得到:

$$AX - XB = U_1T_1U_1^{\mathrm{H}}X - XB = C$$
$$\Leftrightarrow$$
$$T_1(U_1^{\mathrm{H}}X) - (U_1^{\mathrm{H}}X)B = U_1^{\mathrm{H}}C$$
$$\Leftrightarrow$$
$$T_1Y - YB = \widetilde{C} \text{ where } \begin{cases} Y = U_1^{\mathrm{H}}X \in \mathbb{C}^{m\times n} \\ \widetilde{C} = U_1^{\mathrm{H}}C \in \mathbb{C}^{m\times n} \end{cases}$$

我们也可以对 B 做 Schur 分解 $B = U_2T_2U_2^{\mathrm{H}}$ (代价为 $O(n^3)$)，得到:

$$AX - XB = AX - XU_2T_2U_2^{\mathrm{H}} = C$$
$$\Leftrightarrow$$
$$A(XU_2) - (XU_2)T_2 = CU_2$$
$$\Leftrightarrow$$
$$AY - YT_2 = \widetilde{C} \text{ where } \begin{cases} Y = XU_2 \in \mathbb{C}^{m\times n} \\ \widetilde{C} = CU_2 \in \mathbb{C}^{m\times n} \end{cases}$$

因此 Schur 分解预处理代价是 $O(\min(m^3, n^3))$
我们可以比较 A, B 的阶数，哪个阶数小就对哪个做 Schur 分解.

不失一般性，假设 $m > n$，对 $B \in \mathbb{C}^{n\times n}$ 做 Schur 分解 $B = UTU^{\mathrm{H}}$:

$$AX - XB = AX - XUTU^{\mathrm{H}} = C$$
$$\Leftrightarrow$$
$$A(XU) - (XU)T = CU$$
$$\Leftrightarrow$$
$$AY - YT = \widetilde{C} \text{ where } \begin{cases} Y = XU \in \mathbb{C}^{m\times n} \\ \widetilde{C} = CU \in \mathbb{C}^{m\times n} \end{cases}$$

我们可以首先解出 Y 的第 1 列 Ye_1:
(记 $T = [t_{ij}]_{i,j=1}^n$，则 $Te_1 = t_{11}e_1$)

$$AYe_1 - YTe_1 = \widetilde{C}e_1$$
$$\Leftrightarrow$$
$$A(Ye_1) - t_{11}Ye_1 = \widetilde{C}e_1$$

这相当于求解一个线性方程组，得到 Y 的第 1 列 (Ye_1)
然后利用回代法的思想求解 Y 的剩余各列即可.
上述算法称为 **Bartels-Stewart 算法**

(Bartels-Stewart 算法, Matrix Computation 算法 7.6.2)
给定矩阵 $A \in \mathbb{C}^{m\times m}, C \in \mathbb{C}^{m\times n}$ 和上三角阵 $T \in \mathbb{C}^{n\times n}$ (其中 $m < n$，且 A, T 没有公共特征值)
本算法用方程 $AY - YT = C$ 的解覆盖 C:

$$\begin{aligned}
&\text{function: } Y = \text{Bartels-Stewart}(A, T, C) \\
&\quad \text{for } k = 1 : n \\
&\qquad C(1:m, k) = C(1:m, k) + C(1:p, 1:k-1)T(1:k-1, k) \\
&\qquad \text{solve } (A - T(k,k)I_m)y = C(1:m, k) \text{ to get } y \\
&\qquad C(1:m, k) = y \\
&\quad \text{end} \\
&\quad Y = C \\
&\text{end}
\end{aligned}$$

此算法需要 $mn(m+n)$ 的计算量.

5.4.3 块对角化

设 Schur 分解 $A = UTU^{\mathrm{H}}$ 的划分如下:

$$T=\begin{bmatrix} T_{11} & T_{12} & \cdots & T_{1d} \\ & T_{22} & \cdots & T_{2d} \\ & & \ddots & \vdots \\ & & & T_{dd} \end{bmatrix}$$

设 $T_{ii}\in\mathbb{C}^{n_i\times n_i}\ (i=1,\dots,d)$ 的特征值互不相交.

(Matrix Analysis 定理 2.4.6.1)
设 $A\in\mathbb{C}^{n\times n}$ 的不同特征值 $\lambda_1,\dots,\lambda_d$ 的代数重数分别为 $n_1,\dots,n_d$
Schur 分解定理保证了 A 酉相似于一个 $d\times d$ 分块上三角阵 $T=[T_{ij}]_{i,j=1}^{d}$
其中 $T_{ij}\in\mathbb{R}^{n_i\times n_j}$，且每一个对角分块 T_{ii} 分别是对角元全为 λ_i 的上三角阵.
因此 A 就相似于 $T_{11}\oplus\cdots\oplus T_{dd}$ (因为二者具有完全一致的特征值)
总之，存在酉矩阵 $U\in\mathbb{C}^{n\times n}$ 和非奇异矩阵 $S\in\mathbb{C}^{n\times n}$ 使得:

$$S^{-1}(U^{\mathrm{H}}AU)S=S^{-1}TS=\begin{bmatrix} T_{11} & & \\ & \ddots & \\ & & T_{dd} \end{bmatrix}$$

邵老师提供的证明:
考虑 2×2 分块的情况，我们假设存在 X 使得:

$$\begin{aligned}\begin{bmatrix} T_{11} & T_{12} \\ & T_{22} \end{bmatrix}&=\begin{bmatrix} I & X \\ & I \end{bmatrix}\begin{bmatrix} T_{11} & \\ & T_{22} \end{bmatrix}\begin{bmatrix} I & -X \\ & I \end{bmatrix}\quad \left(\text{note that }\begin{bmatrix} I & X \\ & I \end{bmatrix}^{-1}=\begin{bmatrix} I & -X \\ & I \end{bmatrix}\right)\\ &=\begin{bmatrix} T_{11} & XT_{22} \\ & T_{22} \end{bmatrix}\begin{bmatrix} I & -X \\ & I \end{bmatrix}\\ &=\begin{bmatrix} T_{11} & XT_{22}-T_{11}X \\ & T_{22} \end{bmatrix}\end{aligned}$$

要使上式成立，就等价于 X 是 Sylvester 方程 $XT_{22}-T_{11}X=0$ 的解.
由于 T_{11},T_{22} 没有公共特征值，故根据 Sylvester 定理可知上述 Sylvester 方程具有唯一解.

> **(Sylvester 定理, Matrix Analysis 定理 2.4.4.1)**
> 设 $A\in\mathbb{C}^{m\times m},B\in\mathbb{C}^{n\times n},C\in\mathbb{C}^{m\times n}$
> 当且仅当 A,B 没有公共特征值 (即 $\mathrm{eig}(A)\cap\mathrm{eig}(B)=\emptyset$) 时，Sylvester 方程 $AX-XB=C$ 有唯一解 $X\in\mathbb{C}^{m\times n}$

根据数学归纳法容易将上述结论推广到 $d\times d$ 分块的情况，我们只需递归地对其进行 2×2 划分即可.
命题得证.

下面我们给出块对角化的具体算法.
将 I_n 与 T 共形地分块为:

$$I_n=[E_1,E_2,\dots,E_d]=\begin{bmatrix} E_{11} & E_{12} & \cdots & E_{1d} \\ E_{21} & E_{22} & \cdots & E_{2d} \\ \vdots & \vdots & & \vdots \\ E_{d1} & E_{d2} & \cdots & E_{dd} \end{bmatrix}$$

定义 $X_{ij}=I_n+E_iY_{ij}E_j^{\mathrm{T}}\in\mathbb{C}^{n\times n}(i<j)$ (其中 $Y_{ij}\in\mathbb{C}^{n_i\times n_j}$)
换言之，在单位矩阵 I_n 的 (i,j) 分块处填入 Y_{ij} 就得到了 X_{ij}

注意到 $X_{ij}^{-1}=I_n-E_iY_{ij}E_j^{\mathrm{T}}$
若记 $\widetilde{T}=X_{ij}^{-1}TX_{ij}$，则 $T,\widetilde{T}$ 只有以下分块不同:

$$\begin{aligned}\widetilde{T}_{ij}&=T_{ii}Y_{ij}-Y_{ij}T_{jj}+T_{ij}\\ \widetilde{T}_{ik}&=T_{ik}-Y_{ij}T_{jk}\quad(k=j+1:d)\\ \widetilde{T}_{kj}&=T_{kj}+T_{ki}Y_{ij}\quad(k=1:i-1)\end{aligned}$$

我们希望找到 Y_{ij} 使得 $\widetilde{T}_{ij}=T_{ii}Y_{ij}-Y_{ij}T_{jj}+T_{ij}=0_{n_i\times n_j}$
也就只需使用 Bartels-Stewart 算法求解 Sylvester 方程 $T_{ii}Y-YT_{jj}=-T_{ij}$ 即可.
于是我们得到如下算法:

(拟上三角阵的块对角化, Matrix Computation 算法 7.6.3)
给定酉矩阵 $U\in\mathbb{C}^{n\times n}$ 和一个拟上三角阵 $T=U^{\mathrm{H}}AU$，且 T 具有分块形式:

$$T=\begin{bmatrix} T_{11} & T_{12} & \cdots & T_{1d} \\ & T_{22} & \cdots & T_{2d} \\ & & \ddots & \vdots \\ & & & T_{dd} \end{bmatrix}$$

其中 $T_{ii}\in\mathbb{C}^{n_i\times n_i}\ (i=1,\dots,d)$ 的特征值互不相交.
下面的算法用 US 覆盖 U (假设 U 与 T 共形地划分)
其中非奇异阵 $S\in\mathbb{C}^{n\times n}$ 使得 $S^{-1}TS=T_{11}\oplus\cdots\oplus T_{dd}$:

function: $U = \text{Block\_Diagonalization}(T, U)$
    for $j = 2 : d$
        for $i = 1 : j - 1$
            $Y = \text{Bartels-Stewart}(T_{ii}, T_{jj}, -T_{ij})$ (Solve $T_{ii}Y - YT_{jj} = -T_{ij}$)
            for $k = j + 1 : d$
                $T_{ik} = T_{ik} - YT_{jk}$
            end
            for $k = 1 : d$
                $U_{kj} = U_{kj} + U_{ki}Y$
            end
        end
    end
end

The End