

第八章 矩阵特征值问题计算

这一章我们来介绍矩阵特征值和特征向量的计算方法.大家知道,求一个矩阵的特征值的问题实质上是求一个多项式的根的问题,而数学上已经证明:5阶以上的多项式的根一般不能用有限次运算求得.因此,矩阵特征值的计算方法本质上都是迭代的.目前,已有不少非常成熟的数值方法用于计算矩阵的全部或部分特征值和特征向量.而全面系统地介绍所有这些重要的数值方法,会远远超出我们这门课的范围,因而这里我们仅介绍几类最常用的基本方法.

1 基本概念与性质

为了推导和分析算法方便起见,我们先简要地介绍一些与矩阵特征值和特征向量有关的基本概念和重要结果,作为对初等线性代数有关内容的复习和补充.由于本书篇幅所限,本节下面所给出的所有定理均不作证明,有兴趣的读者可参阅有关的参考书.

设 $A \in \mathbf{C}^{n \times n}$.大家知道,一个复数 λ 称作是 A 的一个**特征值**是指存在非零向量 $x \in \mathbf{C}^n$ 使得 $Ax = \lambda x$; x 称作是 A 之属于 λ 的一个**特征向量**.由此可知, λ 是 A 的一个特征值的充分必要条件是 $\det(\lambda I - A) = 0$. 因而称多项式

$$p_A(\lambda) = \det(\lambda I - A)$$

为 A 的**特征多项式**.由行列式的性质易知 $p_A(\lambda)$ 是一个首项为 λ^n 的 n 次多项式,因而由代数基本定理知 $p_A(\lambda)$ 有 n 个根,即 A 有 n 个特征值.记 A 的特征值的全体为 $\lambda(A)$,通常称之为 A 的**谱集**.现假定 $p_A(\lambda)$ 有如下的分解

$$p_A(\lambda) = (\lambda - \lambda_1)^{n_1} (\lambda - \lambda_2)^{n_2} \cdots (\lambda - \lambda_p)^{n_p},$$

其中 $n_1 + n_2 + \cdots + n_p = n$, $\lambda_i \neq \lambda_j$, $i \neq j$, 则称 n_i 为 λ_i 的**代数重数** (简称**重数**); 而称数

$$m_i = n - \text{rank}(\lambda_i I - A)$$

为 λ_i 的**几何重数**.易知, $m_i \leq n_i$, $i = 1, \dots, p$.如果 $n_i = 1$, 则称 λ_i 是 A 的一个**单特征值**; 否则,称 λ_i 是 A 的一个**重特征值**.对于一个特征值 λ_i , 如果 $n_i = m_i$, 则称其是 A 的一个**半单特征值**.显然,单特征值必是半单特征值.如果 A 的所有特征值都是半单的,则称 A 是**非亏损的**.容易证明, A 是非亏损的充分必要条件是 A 有 n 个线性无关的特征向量(即 A 是可对角化矩阵).

设 $A, B \in \mathbf{C}^{n \times n}$.若存在非奇异阵 $X \in \mathbf{C}^{n \times n}$ 使得

$$B = XAX^{-1},$$

则称 A 与 B 是**相似的**, 而上述变换称作是**相似变换**.若 A 与 B 相似, 则 A 和 B 有相同的特征值,而且 x 是 A 的一个特征向量的充分必要条件是 $y = Xx$ 是 B 的一个特征向量.这样,如果我们能够找到一个适当的变换矩阵 X , 使 B 的特征值和特征向量易于求得, 则我们就可立即得到 A 的特征值和相应的特征向量.很多计算矩阵特征值和特征向量的方法正是基于这一基本思想而得到的.从理论上讲,利用相似变换可以将一个矩阵约化成的最简单形式是Jordan标准型,即有

定理6.1.1 (Jordan分解定理) 设 $A \in \mathbf{C}^{n \times n}$ 有 r 个互不相同的特征值 $\lambda_1, \dots, \lambda_r$, 其重数分别为 $n(\lambda_1), \dots, n(\lambda_r)$, 则必存在一个非奇异矩阵 $P \in \mathbf{C}^{n \times n}$ 使得

$$P^{-1}AP = \begin{bmatrix} J(\lambda_1) & & & \\ & J(\lambda_2) & & \\ & & \ddots & \\ & & & J(\lambda_r) \end{bmatrix},$$

其中

$$J(\lambda_i) = \text{diag}(J_1(\lambda_i), \dots, J_{k_i}(\lambda_i)) \in \mathbf{C}^{n(\lambda_i) \times n(\lambda_i)}, \quad 1 \leq i \leq r,$$

$$J_j(\lambda_i) = \begin{bmatrix} \lambda_i & 1 & & \\ & \lambda_i & \ddots & \\ & & \ddots & 1 \\ & & & \lambda_i \end{bmatrix} \in \mathbf{C}^{n_j(\lambda_i) \times n_j(\lambda_i)}, \quad 1 \leq j \leq k_i,$$

$$n_1(\lambda_i) + \dots + n_{k_i}(\lambda_i) = n(\lambda_i), \quad 1 \leq i \leq r;$$

并且除了 $J_j(\lambda_i)$ 的排列次序可以改变外, J 是唯一确定的.

上述定理中的矩阵 J 称作 A 的 **Jordan标准型**, 其中每个子矩阵 $J_j(\lambda_i)$ 称作 **Jordan块**.

如果限定变换矩阵为酉矩阵, 则有如下著名的Schur分解定理.

定理6.1.2 (Schur分解定理) 设 $A \in \mathbf{C}^{n \times n}$, 则存在酉矩阵 $U \in \mathbf{C}^{n \times n}$ 使得

$$U^*AU = T,$$

其中 T 是上三角矩阵; 而且适当选取 U , 可使 T 的对角元素按任意指定的顺序排列.

这一定理无论在理论上还是在实际应用上都是非常重要的, 著名的QR方法就是基于这一定理而设计的.

下述定理对于估计某些特征值的界限是十分方便而有用的.

定理6.1.3 (Gerschgorin圆盘定理) 设 $A = [a_{ij}] \in \mathbf{C}^{n \times n}$, 令

$$G_i(A) = \left\{ z \in \mathbf{C} : |z - a_{ii}| \leq \sum_{j \neq i} |a_{ij}| \right\}, \quad i = 1, \dots, n,$$

则有

$$\lambda(A) \subset G_1(A) \cup G_2(A) \cup \dots \cup G_n(A).$$

从数值计算的角度来看, 首先应弄清的问题是要计算的特征值和特征向量是否是病态的, 也就是说矩阵的元素有微小的变化, 是否会引起所关心的特征值和特征向量的巨大变化. 对于一般方阵来说, 这一问题是非常复杂的, 限于篇幅, 这里我们只介绍一个简单而又非常重要的结果.

假定 λ 是 A 的一个单特征值, x 是属于它的单位特征向量(即 $Ax = \lambda x$ 且 $\|x\|_2 = 1$). 令 $U = [x, U_2] \in \mathbf{C}^{n \times n}$ 是酉矩阵($U^*U = I$), 即 U 的列向量构成 \mathbf{C}^n 的一组标准正交基, 则有

$$U^*AU = \begin{bmatrix} \lambda & x^*AU_2 \\ 0 & A_2 \end{bmatrix},$$

其中 $A_2 = U_2^*AU_2$ 是 $n-1$ 阶方阵. 由 λ 是 A 的单特征值的假定, 知

$$\delta = \min_{\mu \in \lambda(A_2)} |\lambda - \mu| > 0.$$

于是我们可定义

$$\Sigma^\perp = U_2(\lambda I - A_2)^{-1}U_2^*.$$

此外, 由于 $\det(\lambda I - A^T) = \det(\lambda I - A) = 0$, 故必存在非零向量 $y \in \mathbf{C}^n$ 使 $y^T A = \lambda y^T$. 通常称 y 为 A 之属于 λ 的左特征向量. λ 是单特征值的条件蕴含着 $y^T x \neq 0$. 故可选取 y 使 $y^T x = 1$. 若给矩阵 A 以微小的扰动使其变为 \tilde{A} , 记 $\varepsilon = \|\tilde{A} - A\|_2$, 则存在 \tilde{A} 的一个特征值 $\tilde{\lambda}$ 和对应的特征向量 \tilde{x} , 使得

$$|\tilde{\lambda} - \lambda| \leq \|y\|_2 \varepsilon + O(\varepsilon^2), \quad \|\tilde{x} - x\|_2 \leq \|\Sigma^\perp\|_2 \varepsilon + O(\varepsilon^2).$$

这表明 λ 和 x 的敏感性分别与 $\|y\|_2$ 和 $\|\Sigma^\perp\|_2$ 的大小有关. 因此, 我们分别称 $\|y\|_2$ 和 $\|\Sigma^\perp\|_2$ 为特征值 λ 和特征向量 x 的条件数, 记作

$$\text{cond}(\lambda) = \|y\|_2 \quad \text{和} \quad \text{cond}(x) = \|\Sigma^\perp\|_2.$$

2 幂 法

幂法是计算一个矩阵的模最大特征值和对应的特征向量的一种迭代方法. 为了说明幂法的基本思想, 我们先假定 $A \in \mathbf{C}^{n \times n}$ 是可对角化的, 即 A 有如下分解

$$A = X\Lambda X^{-1}, \tag{2.1}$$

其中 $\Lambda = \text{diag}(\lambda_1, \dots, \lambda_n)$, $X = [x_1, \dots, x_n] \in \mathbf{C}^{n \times n}$ 非奇异, 再假定

$$|\lambda_1| > |\lambda_2| \geq \dots \geq |\lambda_n|. \tag{2.2}$$

现任取一向量 $u_0 \in \mathbf{C}^n$. 由于 X 的列向量构成 \mathbf{C}^n 的一组基, 故 u_0 可表示为

$$u_0 = \alpha_1 x_1 + \alpha_2 x_2 + \dots + \alpha_n x_n, \tag{2.3}$$

这里 $\alpha_i \in \mathbf{C}$. 这样, 我们有

$$\begin{aligned} A^k u_0 &= \sum_{j=1}^n \alpha_j A^k x_j = \sum_{j=1}^n \alpha_j \lambda_j^k x_j \\ &= \lambda_1^k \left(\alpha_1 x_1 + \sum_{j=2}^n \alpha_j \left(\frac{\lambda_j}{\lambda_1} \right)^k x_j \right), \end{aligned} \tag{2.4}$$

由此即知,

$$\lim_{k \rightarrow \infty} \frac{A^k u_0}{\lambda_1^k} = \alpha_1 x_1.$$

这表明, 当 $\alpha_1 \neq 0$ 而且 k 充分大时, 向量

$$u_k = \frac{A^k u_0}{\lambda_1^k} \quad (2.5)$$

就是 A 的一个很好的近似特征向量.

这样, 我们自然想到用(2.5)来求 A 的近似特征向量. 然而, 实际计算时, 这是行不通的. 其原因有二: 一是我们事先并不知道 A 的特征值 λ_1 ; 二是对充分大的 k 计算 A^k 的工作量太大.

仔细观察(2.5), 不难发现(2.5)中的 λ_1^k 仅改变向量 $A^k u_0$ 的长度, 并不影响它的方向, 而我们所感兴趣的只是 $A^k u_0$ 的方向, 并非它的长度, 因此我们不必非用 λ_1^k 来约化 $A^k u_0$ 的长度, 而可用其他方便的常数来进行约化(为了防止溢出, 约化是必要的); 其次计算 $A^k u_0$, 也并不需事先将 A^k 算好之后再来计算, 只需迭代地进行即可. 基于这样的考虑, 我们可设计如下的迭代格式:

$$\begin{aligned} y_k &= A u_{k-1}, \\ \mu_k &= \zeta_j^{(k)}, \quad \zeta_j^{(k)} \text{ 是 } y_k \text{ 的模最大分量}, \\ u_k &= y_k / \mu_k, \end{aligned} \quad (2.6)$$

其中 $u_0 \in \mathbf{C}^n$ 是任意给定的初始向量, 通常要求 $\|u_0\|_\infty = 1$.

这一迭代方法称作**幂法**, 其收敛性定理如下:

定理6.2.1 设 $A \in \mathbf{C}^{n \times n}$ 有 p 个互不相同的特征值满足 $|\lambda_1| > |\lambda_2| \geq \cdots \geq |\lambda_p|$, 并且模最大特征值 λ_1 是半单的(即 λ_1 的几何重数等于它的代数重数). 如果初始向量 u_0 在 λ_1 的特征子空间上的投影不为零, 则由(2.6) 产生的向量序列 $\{u_k\}$ 收敛到 λ_1 的一个特征向量 x_1 , 而且由(2.6)产生的数值序列 $\{\mu_k\}$ 收敛到 λ_1 .

证明 由假定知 A 有如下的Jordan分解

$$A = X \operatorname{diag} (J_1, \dots, J_p) X^{-1}, \quad (2.7)$$

其中 $X \in \mathbf{C}^{n \times n}$ 是非奇异的, $J_i \in \mathbf{C}^{n_i \times n_i}$ 是由属于 λ_i 的Jordan块构成的块上三角矩阵, $n_1 + \cdots + n_p = n$. 而 λ_1 为半单的假定蕴含着 $J_1 = \lambda_1 I_{n_1}$, 这里 I_{n_1} 表示 $n_1 \times n_1$ 单位矩阵. 令 $y = X^{-1} u_0$, 并将 y 与 X 作如下分块

$$y = (y_1^T, y_2^T, \dots, y_p^T)^T, \quad X = [X_1, X_2, \dots, X_p].$$

则由(2.7)得

$$\begin{aligned} A^k u_0 &= X \operatorname{diag} (J_1^k, \dots, J_p^k) X^{-1} u_0 \\ &= X_1 J_1^k y_1 + X_2 J_2^k y_2 + \cdots + X_p J_p^k y_p \\ &= \lambda_1^k X_1 y_1 + X_2 J_2^k y_2 + \cdots + X_p J_p^k y_p \\ &= \lambda_1^k \left(X_1 y_1 + X_2 \left(\frac{J_2}{\lambda_1} \right)^k y_2 + \cdots + X_p \left(\frac{J_p}{\lambda_1} \right)^k y_p \right). \end{aligned}$$

注意到 $\lambda_1^{-1}J_i$ 的谱半径为 $\rho(\lambda_1^{-1}J_i) = |\lambda_i|/|\lambda_1| < 1, i = 2, 3, \dots, p$, 即知上式蕴含着

$$\lim_{k \rightarrow \infty} \frac{1}{\lambda_1^k} A^k u_0 = X_1 y_1. \quad (2.8)$$

而假定 u_0 在 λ_1 的特征子空间上投影不为零蕴含着 $X_1 y_1 \neq 0$. 令 $x_1 = \zeta^{-1} X_1 y_1$, 其中 ζ 为 $X_1 y_1$ 的模最大的分量, 显然 x_1 是属于 λ_1 的一个特征向量. 记 ζ_k 为 $A^k u_0$ 的模最大的分量, 则 $\lambda_1^{-k} A^k u_0$ 之模最大分量即为 $\zeta_k \lambda_1^{-k}$. 由(2.6)知,

$$u_k = \frac{A u_{k-1}}{\mu_k} = \frac{A^k u_0}{\mu_k \mu_{k-1} \dots \mu_1} = \frac{A^k u_0}{\zeta_k} = \frac{A^k u_0}{\lambda_1^k} \bigg/ \frac{\zeta_k}{\lambda_1^k}.$$

再由(2.8)即知 $\{u_k\}$ 收敛而且

$$\lim_{k \rightarrow \infty} u_k = \frac{X_1 y_1}{\zeta} = x_1.$$

由等式 $A u_{k-1} = \mu_k u_k$ 及 $\{u_k\}$ 收敛到属于 λ_1 的一个模最大分量为1的特征向量, 立即可以推知 $\{\mu_k\}$ 收敛到 λ_1 . \square

当定理6.2.1的条件不满足时, 由幂法(2.6)产生的序列的收敛性分析将变得非常复杂, 这时 $\{u_k\}$ 可能有若干个收敛于不同向量的子序列(参见本节习题5). 例如, 假设 $A = X D X^{-1}$, 其中

$$X = \begin{bmatrix} 1 & 0 & -1 & 0 \\ 0 & 1 & -1 & 0 \\ 1 & 2 & 1 & 1 \\ -1 & 0 & 0 & 1 \end{bmatrix}, \quad D = \text{diag}(3, 2, 1, -3).$$

此时 A 有两个模最大的特征值 $\lambda_1 = 3$ 和 $\lambda_2 = -3$. 因此定理6.2.1的条件不满足. 取初始向量为 $u_0 = (1, 1, 1, 1)^T$, 通过简单的计算知,

$$A^k u_0 = X D^k X^{-1} u_0 = \frac{1}{5} \begin{bmatrix} 3^k + 4 \\ 2^k + 4 \\ 3^k + 2^{k+1} - 4 + 6(-3)^k \\ -3^k + 6(-3)^k \end{bmatrix},$$

由此即知, 由幂法(2.6)产生的向量序列 $\{u_k\}$ 有两个收敛的子序列, 分别收敛于向量

$$x_1 = \left(\frac{1}{7}, 0, 1, \frac{5}{7}\right)^T \quad \text{和} \quad x_2 = \left(-\frac{1}{7}, 0, \frac{5}{7}, 1\right)^T.$$

注意此时 $x_1 + x_2$ 和 $x_1 - x_2$ 分别是属于 -3 和 3 的特征向量. 事实上, 适当修改(2.6)可使幂法对于此例所述的情况下亦是收敛的. 请读者作为练习修改(2.6)使其适用于 $\lambda_1 = -\lambda_2$ 而 $|\lambda_2| > |\lambda_3| \geq \dots |\lambda_n|$ 的情形.

此外, 从定理6.2.1的证明亦可看出, 幂法的收敛速度主要取决于 $|\lambda_2|/|\lambda_1|$ 的大小. 在定理6.2.1的条件下, 这个数总是小于1的, 它越小收敛也就越快. 当它接近于1时, 收敛是很慢的. 为了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. 如果适当选取 μ 可使 $A - \mu I$ 之模最大特征值与其他特征值之模的距离更大, 就可起到加速的目的. 例如若在上例中

取 $\mu = 3$ ，即若对上面给出的矩阵 A 应用幂法于 $A - 3I$ ，则此时产生的向量序列 $\{u_k\}$ 将收敛到 A 之属于 -3 的一个特征向量。

用幂法可以求矩阵 A 的一个模最大的特征值 λ_1 及其对应的一个特征向量 x_1 。假如我们还要求第二个模最大的特征值 λ_2 ，直接用(2.6)进行迭代是不行的，必须先对原矩阵降阶才行。降阶就是在知道了 λ_1 和 x_1 的前提下，把矩阵 A 降低一阶，使它只包含 A 的其余特征值 $\lambda_2, \dots, \lambda_n$ 。用来完成这一任务的方法通常称作收缩技巧，最简单实用的收缩技巧是利用正交变换。假设

$$Ax_1 = \lambda_1 x_1, \quad (2.9)$$

现假酉矩阵 P 使

$$Px_1 = \alpha e_1. \quad (2.10)$$

这里 $e_1 = (1, 0, \dots, 0)^T$ 。则将(2.10)代入(2.9)并整理可得

$$PAP^*e_1 = \lambda_1 e_1.$$

即 PAP^* 有如下形状

$$PAP^* = \begin{bmatrix} \lambda_1 & * \\ 0 & B_1 \end{bmatrix},$$

其中 B_1 是 $n-1$ 阶方阵，并且它的特征值是 $\lambda_2, \dots, \lambda_n$ 。因此，要求 λ_2 ，只要对 B_2 应用幂法即可。而变换(2.10)可用复的Householder变换来实现。

作为本节的结束，我们希望指出的是，由于幂法的计算公式依赖于矩阵特征值的分布情况，因此实际使用时很不方便，特别是不适应于自动计算，只是在矩阵阶数非常高、无法利用其他更有效的算法时，才用幂法计算少数几个模最大的特征值和相应的特征向量。然而，幂法的基本思想是重要的，由它可以诱导出一些更有效的算法。

3 反幂法

反幂法又称反迭代法，就是应用幂法于 A^{-1} 上求 A 的模最小特征值和对应的特征向量。因此，其基本迭代格式为：

$$\begin{aligned} Ay_k &= z_{k-1}, \\ \mu_k &= \zeta_i, \quad \zeta_i \text{ 是 } y_k \text{ 的模最大分量}, \\ z_k &= y_k / \mu_k. \end{aligned}$$

由上一节的讨论知，若 A 的特征值为 $|\lambda_n| < |\lambda_{n-1}| \leq \dots \leq |\lambda_1|$ 。则 $\{z_k\}$ 收敛到 A 之对应于 λ_n 的一个特征向量，而 $\{\mu_k\}$ 收敛于 λ_n^{-1} ，其收敛速度由 $|\lambda_n|/|\lambda_{n-1}|$ 的大小来决定。

在实际应用中，反幂法主要是用来求特征向量的，是在用某种方法求得 A 的某个特征值 λ_i 的近似值 $\tilde{\lambda}_i$ 之后，然后应用反幂法于 $A - \tilde{\lambda}_i I$ 上。也就是说在实际计算中常用的是带位移的反幂法。设 μ 是给定的位移。带原点位移 μ 的反幂法的迭代格式如下：

$$\begin{aligned} (A - \mu I)v_k &= z_{k-1}, \\ z_k &= v_k / \|v_k\|_2, \end{aligned} \quad k = 1, 2, \dots \quad (3.1)$$

从(3.1)可以看出, 反幂法每迭代一次就需要解一个线性方程组, 这要比幂法运算量大的多. 但是, 由于方程组的系数矩阵不随 k 的变化而变化, 所以能够事先对它进行列选主元的 LU 分解, 然后每次迭代就只需解两个三角形方程组即可.

另外需要顺便指出的是, 这里只是为了下面的分析方便, 而在(3.1)中采取用 $\|\cdot\|_2$ 进行规范化, 在实际使用时是以 $\|\cdot\|_\infty$ 进行规范化的.

假定我们将 A 的特征值排序为

$$0 < |\lambda_1 - \mu| < |\lambda_2 - \mu| \leq |\lambda_3 - \mu| \leq \cdots \leq |\lambda_n - \mu|,$$

则由前一节对幂法讨论知, (3.1)产生的向量序列 $\{z_k\}$ 将收敛到 λ_1 的一个特征向量, 其收敛速度取决于 $|\lambda_1 - \mu|/|\lambda_2 - \mu|$ 的大小, μ 与 λ_1 越靠近, 其收敛速度就越快.

由此可见, 从收敛速度的角度来考虑, 用(3.1)进行迭代时, μ 取得越靠近 A 的某个特征值越好. 但是当 μ 与 A 的特征值很靠近时, $A - \mu I$ 就与一个奇异矩阵很靠近, 每迭代一步就需解一个非常病态的线性方程组. 然而, 实际计算的经验和理论分析的结果表明: $A - \mu I$ 的病态性, 并不影响其收敛速度, 而且当 μ 与 A 的某个特征值很靠近时, 常常只需迭代一次就可以得到相当好的近似特征向量. 为弄清这点, 我们做如下的简要分析.

假定 λ 是 A 的一个单特征值. x 是属于 λ 的单位特征向量. 并假定(3.1)中的位移 μ 与 λ 十分靠近, 且 x 是良态的, 即 $\text{cond}(x)$ 不是太大. 现取 $U_2 \in \mathbf{C}^{n \times (n-1)}$ 使 $[x, U_2]$ 是酉阵, 即 U_2 的列构成 $\text{span}\{x\}^\perp$ ($\text{span}\{x\}^\perp$ 表示特征子空间 $\text{span}\{x\}$ 的正交补空间)的一组标准正交基. 由条件数的定义知

$$\text{cond}(x) = \|U_2(\lambda I - A_2)^{-1}U_2^*\|_2 = \|(\lambda I - A_2)^{-1}U_2^*\|_2, \quad (3.2)$$

其中 $A_2 = U_2^*AU_2$.

现假定给定 z_0 之后, 我们是用列主元的Gauss消去法求解(3.1)中的线性方程组 $(A - \mu I)v_1 = z_0$ 的. 记计算解为 \hat{v}_1 , 由Gauss消去法的误差分析知, \hat{v}_1 满足

$$(A - \mu I - E)\hat{v}_1 = z_0,$$

其中 E 与 $A - \mu I$ 和 z_0 有关, 但 $\|E\|_2$ 有一致的上界, 通常差不多是机器精度. 记 $e = \hat{v}_1 - v_1 = (A - \mu I)^{-1}E\hat{v}_1$, 并将 e 分解为

$$e = x_1 + x_2,$$

其中 $x_1 \in \text{span}\{x\}$, $x_2 \in \text{span}\{x\}^\perp$. 则存在 $\alpha \in \mathbf{C}$ 和 $y \in \mathbf{C}^{n-1}$ 使得

$$x_1 = \alpha x, \quad x_2 = U_2 y.$$

另一方面, 我们有

$$A - \mu I = [x, U_2] \begin{bmatrix} \lambda - \mu & x^*AU_2 \\ 0 & A_2 - \mu I \end{bmatrix} \begin{bmatrix} x^* \\ U_2^* \end{bmatrix},$$

因而

$$(A - \mu I)^{-1} = [x, U_2] \begin{bmatrix} \frac{1}{\lambda - \mu} & \frac{-1}{\lambda - \mu}x^*AU_2(A_2 - \mu I)^{-1} \\ 0 & (A_2 - \mu I)^{-1} \end{bmatrix} \begin{bmatrix} x^* \\ U_2^* \end{bmatrix}.$$

这样, 我们就有

$$\begin{aligned}\alpha &= x^* e = x^* (A - \mu I)^{-1} E \hat{v}_1 \\ &= \frac{x^*}{\lambda - \mu} \left(I - A U_2 (A_2 - \mu I)^{-1} U_2^* \right) E \hat{v}_1, \\ y &= U_2^* e = (A_2 - \mu I)^{-1} U_2^* E \hat{v}_1.\end{aligned}$$

注意到

$$(A_2 - \mu I)^{-1} = \left(I + (\lambda - \mu)(A_2 - \lambda I)^{-1} \right)^{-1} (A_2 - \lambda I)^{-1},$$

我们就知道, 当 μ 与 λ 十分靠近时,

$$s = \|(A_2 - \mu I)^{-1} U_2^*\|_2 \approx \|(A_2 - \lambda I)^{-1} U_2^*\|_2 = \text{cond}(x).$$

因此, 在 x 良态的条件下, s 也不会太大. 于是

$$\|x_2\|_2 = \|y\|_2 \leq s \|E \hat{v}_1\|_2$$

就是一个不太大的量, 但

$$|\alpha| \leq \frac{1}{|\lambda - \mu|} (1 + \|A\|_2 s) \|E \hat{v}_1\|_2$$

将是一个很大的量. 换句话说, 求解线性方程组所引起的误差, 主要对其解在特征子空间 $\text{span}\{x\}$ 上投影的长度有影响, 误差越大, 其计算解在特征子空间 $\text{span}\{x\}$ 上投影就越大. 这对于我们要计算 λ 的近似特征向量而言, 是十分有利的, 因为我们关心的主要是所得向量的方向而非它的大小.

上面的分析实质上亦表明, 在 μ 十分靠近 λ 且 x 良态的条件下, 我们只需使用一次反幂法就可得到 λ 之较好的近似特征向量. 为了说明这一点, 我们先介绍一个基本概念.

假设机器精度为 \mathbf{u} . 对于一个给定的 $\mu \in \mathbf{C}$, 如果存在 $E \in \mathbf{C}^{n \times n}$ 使得

$$\det(A + E - \mu I) = 0 \quad \text{且} \quad \|E\|_2 = O(\mathbf{u}),$$

则我们就说 μ 是 A 的一个**达到机器精度的近似特征值**. 同样, 对于一个给定的 $x \in \mathbf{C}^n$, 如果存在 $F \in \mathbf{C}^{n \times n}$ 满足 $\|F\|_2 = O(\mathbf{u})$ 使得 x 是 $A + F$ 的特征向量, 则我们就说 x 是 A 的一个**达到机器精度的近似特征向量**.

若 μ 是 A 的一个达到机器精度的近似特征值, 则存在 $E \in \mathbf{C}^{n \times n}$ 满足 $\|E\|_2 = O(\mathbf{u})$ 使得 $(A + E - \mu I)y = 0$ 有非零解. 设 $y \in \mathbf{C}^n$ 满足

$$(A + E - \mu I)y = 0 \quad \text{和} \quad \|y\|_2 = 1,$$

那么我们有

$$(A + E)y = \mu y \quad \text{且} \quad \|E\|_2 = O(\mathbf{u}),$$

即 y 是 A 的一个达到机器精度的近似特征向量. 换句话说, 若我们在(3.1)中取 $z_0 = (A - \mu I)y$, 那么在精确计算的前提下, 只需用(3.1)迭代一次就可得到 A 的达到机器精度的特征向量. 当然, 在实际计算时我们不会按照这种方式来选取初始向量 z_0 的, 这里只是说明一个道理, 即在适当的选取初始向量之后, 反幂法具有“一次迭代”性.

这里还需指出的一点是, 在 λ 比较病态时, 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. 请看下例.

设

$$A = \begin{bmatrix} 1 & 1 \\ 10^{-10} & 1 \end{bmatrix}.$$

它有特征值 $\lambda_1 = 0.99999$ 和 $\lambda_2 = 1.00001$, 以及对应的特征向量 $x_1 = (1, -10^{-5})^T$ 和 $x_2 = (1, 10^{-5})^T$. 这两个特征值的条件数都是 10^5 数量级的. 取 $\mu = 1$, $z_0 = (0, 1)^T$, 应用反幂法迭代一次(在10位10进制的浮点数系下进行), 则得 $z_1 = (1, 0)^T$, 并且有 $\|Az_1 - \mu z_1\|_2 = 10^{-10}$. 这说明 z_1 已是 A 的一个达到机器精度的近似特征向量. 但若再迭代一次将产生 $z_2 = (0, 1)^T$, 则有 $\|Az_2 - \mu z_2\|_2 = 1$.

最后, 我们来介绍一下在实际计算时初始向量 z_0 的两种常用的选取方法. 第一种是先利用随机数发生器程序随机地选取向量, 然后再将它规范化后作为初始向量 z_0 . 第二种方法是所谓的“半次迭代法”. 如前所述, 用(3.1)进行迭代时, 首先要作好 $A - \mu I$ 的LU分解:

$$A - \mu I = LU.$$

这里为了叙述简略起见, 我们略去了选主元的排列方阵. 那么第一次迭代为

$$LUv_1 = z_0.$$

我们选 $z_0 = Le$, 其中 e 为分量全为1的向量, 则为了求 v_1 , 只要求解一个三角形方程组

$$Uv_1 = e.$$

这样一来, 初始向量 z_0 不需要明确给出, 而完成这一次迭代只需求解“半次”线性方程组.

4 QR方法

在这一节, 我们来介绍著名的QR方法. QR方法是自电子计算机问世以来矩阵计算的重大进展之一, 也是目前计算一般矩阵的全部特征值和特征向量的最有效方法之一. QR方法是利用正交相似变换将一个给定的矩阵逐步约化为上三角矩阵或拟上三角矩阵的一种迭代方法, 其基本收敛速度是二次的, 当原矩阵实对称时, 则可达到三次收敛.

4.1 基本迭代与收敛性

对给定的 $A_0 = A \in \mathbb{C}^{n \times n}$, QR算法的基本迭代格式如下:

$$\begin{aligned} A_{m-1} &= Q_m R_m, \\ A_m &= R_m Q_m, \end{aligned} \quad m = 1, 2, \dots, \quad (4.1)$$

其中 Q_m 为酉矩阵, R_m 为上三角矩阵. 为了下面的理论分析方便起见, 我们这里暂且要求 R_m 的对角元都是非负的. 由(4.1)容易推出

$$A_m = Q_m^* A_{m-1} Q_m, \quad (4.2)$$

即矩阵序列 $\{A_m\}$ 中的每一个矩阵都与原矩阵 A 相似. 反复运用(4.2) 可得

$$A_m = \tilde{Q}_m^* A \tilde{Q}_m, \quad (4.3)$$

其中 $\tilde{Q}_m = Q_1 Q_2 \cdots Q_m$. 将 $A_m = Q_{m+1} R_{m+1}$ 代入上式即有

$$\tilde{Q}_m Q_{m+1} R_{m+1} = A \tilde{Q}_m,$$

从而有

$$\tilde{Q}_m Q_{m+1} R_{m+1} R_m \cdots R_1 = A \tilde{Q}_m R_m \cdots R_1,$$

即

$$\tilde{Q}_{m+1} \tilde{R}_{m+1} = A \tilde{Q}_m \tilde{R}_m,$$

其中 $\tilde{R}_k = R_k R_{k-1} \cdots R_1$, $k = m, m+1$. 由此即知

$$A^m = \tilde{Q}_m \tilde{R}_m. \quad (4.4)$$

利用(4.4)这一基本关系式, 我们可以导出QR算法与幂法的关系. 记 \tilde{R}_{m+1} 的元素为 γ_{ij} , \tilde{Q}_{m+1} 的第一列为 $q_1^{(m)}$, 则由(4.4)可得

$$A^m e_1 = \gamma_{11} q_1^{(m)}.$$

所以 $q_1^{(m)}$ 可以看作是对 A 用 e_1 作初始向量的幂法所得到的向量. 若 A 的模最大特征值 λ_1 与其他特征值分离, 则 $q_1^{(m)}$ 将收敛到 A 的一个属于 λ_1 的特征向量.

事实上, 在适当条件下, A_m 的所有的或大部分的对角线以下的元素都将趋向于零. 下面的定理仅叙述发生这种情况的一个易于验证的条件.

定理6.4.1 设 A 的 n 个特征值满足 $|\lambda_1| > |\lambda_2| > \cdots > |\lambda_n| > 0$, 并设 Y 的第 i 行是 A 对应于 λ_i 的左特征向量. 如果 Y 有LU分解, 则由(4.1)产生的矩阵 $A_m = [\alpha_{ij}^{(m)}]$ 的对角线以下的元素趋向于零, 同时对角元素 $\alpha_{ii}^{(m)}$ 趋向于 λ_i , $i = 1, 2, \dots, n$.

证明 令

$$X = Y^{-1}, \quad \Lambda = \text{diag}(\lambda_1, \dots, \lambda_n).$$

则有 $A = X \Lambda Y$. 假定 Y 的LU分解为 $Y = LU$, 其中 L 是单位下三角矩阵, U 是上三角矩阵. 这样, 我们有

$$\begin{aligned} A^m &= X \Lambda^m Y = X \Lambda^m L U = X (\Lambda^m L \Lambda^{-m}) \Lambda^m U \\ &= X (I + E_m) \Lambda^m U, \end{aligned} \quad (4.5)$$

其中 $I + E_m = \Lambda^m L \Lambda^{-m}$. 由于 L 是单位下三角矩阵, 而 $|\lambda_i| < |\lambda_j| (i > j)$, 故必有

$$\lim_{m \rightarrow \infty} E_m = 0. \quad (4.6)$$

现在令 X 的QR分解为: $X = QR$. 由 X 非奇异, 我们可要求 R 的对角元素均为正数. 将这一分解代入(4.5)可得

$$A^m = QR(I + E_m) \Lambda^m U = Q(I + R E_m R^{-1}) R \Lambda^m U. \quad (4.7)$$

当 m 充分大时, $I + RE_m R^{-1}$ 是非奇异的, 故它有如下的QR分解

$$I + RE_m R^{-1} = \hat{Q}_m \hat{R}_m, \quad (4.8)$$

其中 \hat{R}_m 的对角元素均为正数. 从(4.6)和(4.8)不难推出

$$\lim_{m \rightarrow \infty} \hat{Q}_m = \lim_{m \rightarrow \infty} \hat{R}_m = I. \quad (4.9)$$

将(4.8)代入(4.7), 得

$$A^m = (Q\hat{Q}_m)(\hat{R}_m R \Lambda^m U),$$

即我们已找到了 A^m 的一个QR分解. 为了保证这一分解中的上三角矩阵的对角元素均为正数, 我们定义

$$D_1 = \text{diag} \left(\frac{\lambda_1}{|\lambda_1|}, \dots, \frac{\lambda_n}{|\lambda_n|} \right),$$

$$D_2 = \text{diag} \left(\frac{u_{11}}{|u_{11}|}, \dots, \frac{u_{nn}}{|u_{nn}|} \right),$$

其中 u_{ii} 是 U 的第 i 个对角元素. 于是我们有

$$A^m = (Q\hat{Q}_m D_1^m D_2)(D_2^{-1} D_1^{-m} \hat{R}_m R \Lambda^m U).$$

将上式与(4.4)比较, 并注意到QR分解的唯一性, 就有

$$\tilde{Q}_m = Q\hat{Q}_m D_1^m D_2, \quad \tilde{R}_m = D_2^{-1} D_1^{-m} \hat{R}_m R \Lambda^m U.$$

将上式代入(4.3)即有

$$A_m = D_2^* (D_1^*)^m \hat{Q}_m^* Q^* A Q \hat{Q}_m D_1^m D_2.$$

再注意到

$$A = X \Lambda Y = X \Lambda X^{-1} = Q R \Lambda R^{-1} Q^*,$$

我们就知

$$A_m = D_2^* (D_1^*)^m \hat{Q}_m^* R \Lambda R^{-1} \hat{Q}_m D_1^m D_2.$$

由此便可立即推出定理的结论成立. □

注6.4.1 A_m 的上三角部分的元素并不一定收敛, 这是因为 D_1^m 的极限一般并不存在.

4.2 实Schur标准形

由于实际应用中所遇到的大量的特征值问题都是关于实矩阵的, 因此我们自然希望设计只涉及实数运算的QR迭代, 即给定 $A \in \mathbf{R}^{n \times n}$, 令 $A_1 = A$, 构造迭代:

$$A_k = Q_k R_k, \quad k = 1, 2, \dots,$$

$$A_{k+1} = R_k Q_k, \quad (4.10)$$

其中 Q_k 是正交矩阵, R_k 是上三角矩阵.

然而, 此时由于复共轭特征值的存在, 我们自然不能再期望(4.10) 产生的 A_k 仍然逼近一个上三角矩阵. 那么, A_k 将会趋向于什么呢? 这就涉及到一个实矩阵在正交相似变换下的标准形问题. 事实上, 我们有

定理6.4.2 (实Schur分解) 设 $A \in \mathbf{R}^{n \times n}$, 则存在正交矩阵 $Q \in \mathbf{R}^{n \times n}$, 使得

$$Q^T A Q = \begin{bmatrix} R_{11} & R_{12} & \cdots & R_{1m} \\ & R_{22} & \cdots & R_{2m} \\ & & \ddots & \vdots \\ & & & R_{mm} \end{bmatrix}, \quad (4.11)$$

其中 R_{ii} 或者是一个实数, 或者是一个具有一对复共轭特征值的2阶方阵.

通常称分解(4.11)为矩阵 A 的**实Schur分解**, 而称其右边的拟上三角矩阵为 A 的**实Schur标准形**. 显然, 只要求得一个实矩阵的实Schur标准形, 我们就可很容易求得它的全部特征值.

由定理6.3.1, 我们不难想到迭代(4.10)产生的 A_k 将应该逼近于 A 的实Schur标准形.

此外, (4.10)作为一种实用的迭代法是没有竞争力的, 其原因有二: 一是每次迭代的运算量太大; 二是收敛速度太慢. 因此, 要想使其成为一种高效的方法, 我们必须减少其每次迭代所需的运算量, 提高其收敛速度. 这正是本节下面所要讨论的主要内容.

在下面的讨论中, 如无特别说明, 我们总假定给定的矩阵是实的.

4.3 上Hessenberg化

实际计算时, 为了减少每次迭代所需的运算量, 总是先将原矩阵 A 经相似变换约化为一个准上三角矩阵, 然后再对约化后的矩阵进行QR迭代.

设 $A \in \mathbf{R}^{n \times n}$. 我们希望计算一个非奇异矩阵 Q 使得

$$\tilde{A} = Q A Q^{-1}$$

具有某种特殊形式. 很自然, 我们希望 \tilde{A} 的零元素越多越好. 首先我们来看利用Householder变换可以得到什么样的 \tilde{A} .

对于给定的 $A = [\alpha_{ij}] \in \mathbf{R}^{n \times n}$, 第一步, 我们自然应该选取Householder 变换 H_1 使得 $H_1 A$ 的第一列有尽可能多的零元素(至多只能有 $n-1$ 个零元素). 然而, 为了保证对 A 进行相似变换, 在对 A 进行了行变换之后, 必须亦对 A 进行同样的列变换, 即应将 H_1 亦右乘于 $H_1 A$ 上变为

$$H_1 A H_1.$$

这样, 为了保证已在 $H_1 A$ 的第一列所出现的零元素不致于在左乘 H_1 时被破坏掉, 我们应该选取 H_1 具有如下形状

$$H_1 = \begin{bmatrix} 1 & 0 \\ 0 & \tilde{H}_1 \end{bmatrix} \begin{matrix} 1 \\ n-1 \end{matrix}. \quad (4.12)$$

利用形如(4.12)的Householder变换对 A 进行相似变换即有

$$H_1 A H_1 = \begin{bmatrix} \alpha_{11} & a_2^T \tilde{H}_1 \\ \tilde{H}_1 a_1 & \tilde{H}_1 A_{22} \tilde{H}_1 \end{bmatrix}, \quad (4.13)$$

其中 $a_1^T = (\alpha_{21}, \alpha_{31}, \dots, \alpha_{n1})$, $a_2^T = (\alpha_{12}, \alpha_{13}, \dots, \alpha_{1n})$, A_{22} 是 A 的右下角的 $n-1$ 阶主子阵. 由(4.13)易知, Householder变换 \tilde{H}_1 的最佳选择应该使得

$$\tilde{H}_1 a_1 = p e_1, \quad (4.14)$$

其中 $p \in \mathbf{R}$, e_1 是 $n-1$ 阶单位矩阵的第一列. 这样一来, 就可选取形如(4.12)的Householder变换 H_1 使得(4.13)第一列有 $n-2$ 个零元素.

然后, 再对 $\tilde{A}_{22} = \tilde{H}_1 A_{22} \tilde{H}_1$ 进行同样的考虑, 又可找到Householder变换

$$\tilde{H}_2 = \begin{bmatrix} 1 & 0 \\ 0 & \hat{H}_2 \end{bmatrix}$$

使得

$$(\tilde{H}_2 \tilde{A}_{22} \tilde{H}_2) e_1 = \begin{bmatrix} * \\ * \\ 0 \\ \vdots \\ 0 \end{bmatrix}.$$

从而令

$$H_2 = \begin{bmatrix} 1 & 0 \\ 0 & \tilde{H}_2 \end{bmatrix},$$

即有

$$H_2 H_1 A H_1 H_2 = \left[\begin{array}{cc|c} h_{11} & h_{12} & \\ h_{21} & h_{22} & * \\ 0 & h_{32} & \\ \hline 0 & & * \end{array} \right].$$

如此进行 $n-2$ 步, 就可找到 $n-2$ 个Householder变换 H_1, \dots, H_{n-2} , 使得

$$H_{n-2} \cdots H_1 A H_1 \cdots H_{n-2} = H,$$

其中 $H = [h_{ij}]$ 满足

$$h_{ij} = 0, \quad i > j + 1,$$

即

$$H = \begin{bmatrix} h_{11} & h_{12} & h_{13} & \cdots & h_{1,n-1} & h_{1n} \\ h_{21} & h_{22} & h_{23} & \cdots & h_{2,n-1} & h_{2n} \\ 0 & h_{32} & h_{33} & \cdots & h_{3,n-1} & h_{3n} \\ \vdots & \vdots & \vdots & & \vdots & \vdots \\ 0 & 0 & 0 & \cdots & h_{n,n-1} & h_{nn} \end{bmatrix}. \quad (4.15)$$

通常称形如(4.15)的矩阵为上Hessenberg矩阵.

现在令

$$Q_0 = H_1 H_2 \cdots H_{n-2},$$

则有

$$Q_0^T A Q_0 = H. \quad (4.16)$$

通常称分解式(4.16)为 A 的上Hessenberg分解.

总结上面利用Householder变换约化一个矩阵为上Hessenberg矩阵的方法可得如下的实用算法.

算法6.4.1 (计算上Hessenberg分解: Householder变换法)

```

for  $k = 1 : n - 2$ 
     $[v, \beta] = \mathbf{house}(A(k+1:n, k))$ 
     $A(k+1:n, k:n) = (I - \beta v v^T) A(k+1:n, k:n)$ 
     $A(1:n, k+1:n) = A(1:n, k+1:n) (I - \beta v v^T)$ 
end

```

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 $10n^3/3$; 如果需要累积 $Q_0 = H_1 \cdots H_{n-2}$, 则还需要再增加运算量 $4n^3/3$.

此外, 上述算法计算得到的上Hessenberg矩阵 \hat{H} 满足

$$\hat{H} = Q^T (A + E) Q,$$

其中 Q 是正交矩阵, $\|E\|_F \leq cn^2 \|A\|_F \mathbf{u}$, 这里 c 是一常数, \mathbf{u} 是机器精度.

当然, 我们亦可用Givens变换将 A 约化为上Hessenberg形, 一般所需要的运算量大约是算法6.4.1的二倍. 但是, 如果 A 有较多的零元素, 则适当安排Givens变换的次序, 可使运算量大为减少. 另外, 为了节省运算量, 也可采用列主元的Gauss消去法将 A 约化为上Hessenberg矩阵. 不过这样做, 虽然运算量少, 但数值稳定性较差.

尽管一般来讲上Hessenberg分解是不唯一的, 然而我们可以证明

定理6.4.3 设 $A \in \mathbf{R}^{n \times n}$ 有如下两个上Hessenberg分解:

$$U^T A U = H, \quad V^T A V = G, \quad (4.17)$$

其中 $U = [u_1, u_2, \dots, u_n]$ 和 $V = [v_1, v_2, \dots, v_n]$ 是 n 阶正交矩阵, $H = [h_{ij}]$ 和 $G = [g_{ij}]$ 是上Hessenberg矩阵. 若 $u_1 = v_1$, 而且 H 的次对角元素 $h_{i+1,i}$ 均不为零, 则存在对角元素均为1或-1的对角矩阵 D 使得

$$U = V D, \quad H = D G D. \quad (4.18)$$

证明 假定对某个 m ($1 \leq m < n$)已证

$$u_j = \varepsilon_j v_j, \quad j = 1, 2, \dots, m, \quad (4.19)$$

其中 $\varepsilon_1 = 1$, $\varepsilon_j = 1$ 或 -1 . 下面来证存在 $\varepsilon_{m+1} = 1$ 或 -1 使得

$$u_{m+1} = \varepsilon_{m+1} v_{m+1}.$$

从(4.17)可得

$$A U = U H, \quad A V = V G.$$

分别比较上面两个矩阵等式的第 m 列, 可得

$$Au_m = h_{1m}u_1 + \cdots + h_{mm}u_m + h_{m+1,m}u_{m+1}, \quad (4.20)$$

$$Av_m = g_{1m}v_1 + \cdots + g_{mm}v_m + g_{m+1,m}v_{m+1}. \quad (4.21)$$

分别在(4.20)和(4.21)两边左乘 u_i^T 和 v_i^T , 可得

$$h_{im} = u_i^T Au_m, \quad g_{im} = v_i^T Av_m, \quad i = 1, 2, \dots, m.$$

再利用(4.19)就有

$$h_{im} = \varepsilon_i \varepsilon_m g_{im}, \quad i = 1, 2, \dots, m. \quad (4.22)$$

将(4.22)代入(4.20), 并利用(4.19)和(4.21), 可得

$$\begin{aligned} h_{m+1,m}u_{m+1} &= \varepsilon_m (Av_m - \varepsilon_1^2 g_{1m}v_1 - \cdots - \varepsilon_m^2 g_{mm}v_m) \\ &= \varepsilon_m (Av_m - g_{1m}v_1 - \cdots - g_{mm}v_m) \\ &= \varepsilon_m g_{m+1,m}v_{m+1}. \end{aligned} \quad (4.23)$$

由此即知

$$|h_{m+1,m}| = |g_{m+1,m}|.$$

而 $h_{m+1,m} \neq 0$, 故(4.23)蕴含着

$$u_{m+1} = \varepsilon_{m+1}v_{m+1},$$

其中 $\varepsilon_{m+1} = 1$ 或 -1 .

因此, 由归纳法原理即知定理得证. \square

一个上Hessenberg矩阵 $H = [h_{ij}]$, 如果其次对角元素均不为零, 即 $h_{i+1,i} \neq 0$, $i = 1, 2, \dots, n-1$, 则称它是**不可约**的. 上述定理表明: 如果 $Q^T A Q = H$ 为不可约的上Hessenberg矩阵, 其中 Q 为正交矩阵, 则 Q 和 H 完全由 Q 的第一列确定(这里是在相差一个正负号的意义下的唯一).

现在假定 $H \in \mathbf{R}^{n \times n}$ 是上Hessenberg矩阵, 我们来考虑对 H 进行一次QR迭代的具体实现问题. 第一步是计算 H 的QR分解. 由于 H 的特殊性, 这一步可用 $n-1$ 个平面旋转变换来完成, 其具体计算细节可从下面5阶矩阵的例子中明白. 设 $n = 5$, 并假定我们已经确定了两个平面旋转变换 P_{12} 和 P_{23} 使得 $P_{23}P_{12}H$ 有如下形状:

$$P_{23}P_{12}H = \begin{bmatrix} \times & \times & \times & \times & \times \\ & \times & \times & \times & \times \\ & & h_{33} & \times & \times \\ & & h_{43} & \times & \times \\ & & & \times & \times \end{bmatrix}.$$

然后在(3, 4)坐标平面内选择平面旋转变换 P_{34} 使 $P_{34}P_{23}P_{12}H$ 的(4, 3)位置上的元素为零. 即确定 $P_{34} = G(3, 4, \theta_3)$ 使得旋转角 θ_3 满足

$$\begin{bmatrix} \cos \theta_3 & \sin \theta_3 \\ -\sin \theta_3 & \cos \theta_3 \end{bmatrix} \begin{bmatrix} h_{33} \\ h_{43} \end{bmatrix} = \begin{bmatrix} \times \\ 0 \end{bmatrix}.$$

这样, $P_{34}P_{23}P_{12}H$ 就有如下形状:

$$P_{34}P_{23}P_{12}H = \begin{bmatrix} \times & \times & \times & \times & \times \\ & \times & \times & \times & \times \\ & & \times & \times & \times \\ & & & \times & \times \\ & & & & \times & \times \end{bmatrix}.$$

由此不难看出, 对一般的 n 阶上Hessenberg矩阵 H , 我们可以确定 $n-1$ 个平面旋转变换 $P_{12}, P_{23}, \dots, P_{n-1,n}$ 使得

$$P_{n-1,n}P_{n-2,n-1} \cdots P_{12}H = R$$

是上三角矩阵. 令 $Q = (P_{n-1,n} \cdots P_{12})^T$, 则 $H = QR$, 即这样就已完成了 H 的QR分解. 要完成一次QR迭代, 我们还须计算

$$\tilde{H} = RQ = RP_{12}^T P_{23}^T \cdots P_{n-1,n}^T.$$

由于 P_{12} 是(1,2)坐标平面内的旋转变换, 因此 RP_{12}^T 仅有前两列与 R 不同, 而 RP_{12}^T 的前两列由 R 的前两列的线性组合构成, R 又是上三角矩阵, 故 RP_{12}^T 必有如下形状($n=5$ 的情形):

$$RP_{12}^T = \begin{bmatrix} \times & \times & \times & \times & \times \\ \times & \times & \times & \times & \times \\ & & \times & \times & \times \\ & & & \times & \times \\ & & & & \times \end{bmatrix}.$$

同样, P_{23} 是(2,3)坐标平面内的旋转变换, $RP_{12}^T P_{23}^T$ 仅有第二列和第三列与 RP_{12}^T 不同, 它们是 RP_{12}^T 的第二和第三列的线性组合, 故 $RP_{12}^T P_{23}^T$ 有如下形状($n=5$ 的情形):

$$P_{23}P_{12}H = \begin{bmatrix} \times & \times & \times & \times & \times \\ \times & \times & \times & \times & \times \\ & \times & \times & \times & \times \\ & & & \times & \times \\ & & & & \times \end{bmatrix}.$$

如此进行下去, 最后我们得到的 \tilde{H} 仍是一个上Hessenberg矩阵. 而且不难算出, 这样进行的一次QR迭代的运算量是 $O(n^2)$. 注意, 对一般方阵进行的一次QR迭代的运算量是 $O(n^3)$.

4.4 带原点位移的QR迭代

从定理6.4.1已经知道, 基本的QR算法是线性收敛的, 其收敛速度取决于特征值之间的分离程度. 为了加速其收敛速度, 类似于反幂法, 可引进原点位移. 设第 m 步迭代的位移为 μ_m , 则带原点位移的QR迭代如下:

$$\begin{aligned} H_m - \mu_m I &= Q_m R_m, \\ H_{m+1} &= R_m Q_m + \mu_m I, \end{aligned}$$

这里 $H_0 = H \in \mathbf{R}^{n \times n}$ 是给定的上Hessenberg阵.

现在来讨论位移的选取. 由于 H_m 为上Hessenberg矩阵, 故其最后一行仅有两个非零元素 $h_{n,n-1}^{(m)}$ 和 $h_{nn}^{(m)}$. 若QR算法收敛, 则当 m 充分大时, $h_{n,n-1}^{(m)}$ 就很小, 因而 $h_{nn}^{(m)}$ 就接近于 H 的一个特征值. 这样, 根据从反幂法所得到的经验, 我们可选取位移为 $\mu_m = h_{nn}^{(m)}$. 事实上, 对于这样选取的位移, 可以证明, 若 $h_{n,n-1}^{(m)} = \varepsilon$ 很小的话, 则经一次带原点位移QR迭代后, 成立

$$h_{n,n-1}^{(m+1)} = O(\varepsilon^2) \quad (4.24)$$

显然, 这只需考察 $H_m - h_{nn}^{(m)}I$ 的右下角的 2×2 子矩阵 H_{m2} 的变化即可. 由前面的讨论可知, 将 $H_m - h_{nn}^{(m)}I$ 约化成上三角阵有 $n-1$ 步, 现假定前面 $n-2$ 步已经完成, 此时的 H_{m2} 变为

$$\tilde{H}_{m2} = \begin{bmatrix} \alpha & \beta \\ \varepsilon & 0 \end{bmatrix},$$

这是因为前 $n-2$ 步不改变 $H_m - h_{nn}^{(m)}I$ 的最后一行. 约化的第 $n-1$ 步就是要消去 ε , 既确定 $c = \cos \theta$ 和 $s = \sin \theta$ 使得

$$\begin{bmatrix} c & s \\ -s & c \end{bmatrix} \begin{bmatrix} \alpha \\ \varepsilon \end{bmatrix} = \begin{bmatrix} \sigma \\ 0 \end{bmatrix}.$$

从平面旋转变换的确定方法易知, 此处

$$c = \frac{\alpha}{\sigma}, \quad s = \frac{\varepsilon}{\sigma}, \quad \sigma = \sqrt{\alpha^2 + \varepsilon^2}.$$

这样, 通过简单的计算可知

$$h_{n,n-1}^{(m+1)} = -s^2 \beta = -\frac{\beta}{\sigma^2} \varepsilon^2,$$

即(4.24)成立. 我们看到通过原点位移, 特征值的渐进收敛速度从线性收敛加速而变成二次收敛.

4.5 双重步位移的QR迭代

上面所讨论的带原点位移的QR迭代, 存在严重的缺点: 若 A 具有复共轭特征值, 则实位移一般并不能起到加速的作用. 为了克服这一缺点, 我们下面来介绍双重步位移的QR迭代, 其基本思想是将两步带原点位移的QR迭代合并为一步, 以避免复数运算.

设 $A \in \mathbf{R}^{n \times n}$, 考察如下的迭代:

$$\begin{cases} H_1 = Q_0^T A Q_0, & (\text{上Hessenberg 分解}) \\ H_k - \mu_k I = Q_k R_k, & (\text{QR 分解}) \\ H_{k+1} = R_k Q_k + \mu_k I, & k = 1, 2, \dots \end{cases} \quad (4.25)$$

不失一般性, 我们可以假定迭代(4.25)中出现的上Hessenberg矩阵都是不可约的. 因若不然, 在迭代的某一步, 已有

$$H_k = \begin{bmatrix} H_{11}^{(k)} & * \\ 0 & H_{22}^{(k)} \end{bmatrix},$$

则我们可以分别对 $H_{11}^{(k)}$ 和 $H_{22}^{(k)}$ 进行QR迭代即可.

大家已经知道, 在一定条件下取位移 $\mu_k = h_{nn}^{(k)}$ 可起到加速收敛的作用; 然而, 大家也熟知实矩阵可以有复特征值. 这样, 假如 H_k 的尾部 2×2 子矩阵

$$G_k = \begin{bmatrix} h_{mm}^{(k)} & h_{mn}^{(k)} \\ h_{nm}^{(k)} & h_{nn}^{(k)} \end{bmatrix}, \quad m = n-1,$$

有一对复共轭特征值 μ_1 和 μ_2 时, 我们就不能期望 $h_{nn}^{(k)}$ 最终收敛于 A 的某个特征值, 因而此种情形再取位移为 $\mu_k = h_{nn}^{(k)}$ 就完全起不到加速收敛的作用. 为了加速收敛, 此时我们自然应该取 μ_1 或 μ_2 作位移. 但这样一来就必须涉及复数运算, 而这又是我们所不希望的. 为了避免复运算的出现, 人们想到用 μ_1 和 μ_2 连续作两次位移, 即进行

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

这里我们记 $H = H_k$. 对上面迭代所产生的矩阵进行一些简单的推算, 可得

$$M = QR, \tag{4.26}$$

$$H_2 = Q^* H Q, \tag{4.27}$$

其中

$$M = (H - \mu_1 I)(H - \mu_2 I), \tag{4.28}$$

$$Q = U_1 U_2, \quad R = R_2 R_1. \tag{4.29}$$

由(4.28)可得

$$M = H^2 - sH + tI, \tag{4.30}$$

其中

$$s = \mu_1 + \mu_2 = h_{mm}^{(k)} + h_{nn}^{(k)} \in \mathbf{R},$$

$$t = \mu_1 \mu_2 = \det G_k \in \mathbf{R}.$$

因此 M 是一个实矩阵; 而且如果 μ_1 和 μ_2 均不是 H 的特征值, 并假定在迭代过程中选取 R_1 和 R_2 的对角元素均为正数, 则由(4.26)可推知, Q 亦是实的; 从而由(4.27)知 H_2 也是实的. 这也就是说, 在没有误差的情况下, 用 μ_1 或 μ_2 连续作两次位移进行QR迭代产生的 H_2 仍是实的上Hessenberg矩阵. 但是, 实际计算时, 由于舍入误差的影响, 如此得到的 H_2 一般并不一定是实的. 这样为了确保计算得到的 H_2 仍是实的, 根据(4.26)和(4.27), 我们自然想到按如下的步骤来计算 H_2 :

- (1) 计算 $M = H^2 - sH + tI$;
- (2) 计算 M 的QR分解: $M = QR$;
- (3) 计算 $H_2 = Q^T H Q$.

然而, 如此计算的第一步形成 M 的运算量就是 $O(n^3)$. 当然, 这是我们所不希望的. 幸运的是, 定理6.4.3告诉我们: 不论采用什么样的方法去求正交矩阵 \tilde{Q} 使 $\tilde{Q}^T H \tilde{Q} = \tilde{H}_2$ 是上Hessenberg矩阵, 只要保证 \tilde{Q} 的第一列与 Q 的第一列一样, 则 \tilde{H}_2 就与 H_2 本质上是一样的(所有元素的绝对值都相等).

当然, 这需要 H_2 是不可约来加以保证的. 因此, 只要 H_2 是不可约的, 则我们就可有很大的自由度去寻求更有效的方法来实现由 H 到 H_2 的变换. 下面的定理给出 H_2 不可约的条件.

定理6.4.4 若 H 是不可约上Hessenberg矩阵, 且 μ_1 和 μ_2 均非 H 的特征值, 则 H_2 也是不可约上Hessenberg矩阵.

证明

用反证法. 记 $H_2 = [\tilde{h}_{ij}]$, 并假定存在 r ($1 \leq r \leq n-1$)使得 $\tilde{h}_{r+1,r} = 0$, 而 $\tilde{h}_{i+1,i} \neq 0$, $i = 1, \dots, r-1$. 比较等式 $HQ = QH_2$ 的两边矩阵的前 r 列, 得

$$\begin{aligned} Hq_j &= \tilde{h}_{1j}q_1 + \dots + \tilde{h}_{jj}q_j + \tilde{h}_{j+1,j}q_{j+1}, \quad j = 1, \dots, r-1, \\ Hq_r &= \tilde{h}_{1r}q_1 + \tilde{h}_{2r}q_2 + \dots + \tilde{h}_{rr}q_r. \end{aligned}$$

由此可得

$$(\alpha_0 I + \alpha_1 H + \dots + \alpha_r H^r)q_1 = 0, \quad (4.31)$$

其中

$$\alpha_r = (\tilde{h}_{21}\tilde{h}_{32}\dots\tilde{h}_{r,r-1})^{-1} \neq 0.$$

由 $M = QR$ 得

$$q_1 = r_{11}^{-1}Me_1.$$

将其代入(4.31), 并注意到 M 也是 H 的多项式, 就有

$$My = 0, \quad (4.32)$$

其中

$$y = (\alpha_0 I + \alpha_1 H + \dots + \alpha_r H^r)e_1.$$

记 $H = [h_{ij}]$, 并注意到 H 是不可约的上Hessenberg矩阵, 直接计算可知 y 的第 $r+1$ 个分量为

$$\alpha_r h_{21} h_{32} \dots h_{r+1,r} \neq 0,$$

这也就是说(4.32)有非零解, 而这与 μ_1 和 μ_2 均非 H 的特征值蕴含着 M 非奇异矛盾. \square

基于定理6.4.1和6.4.4, 我们可以从另外的途径来实现 H 到 H_2 的变换. 首先, 我们从(4.26)知, Q 的第一列与 M 的第一列共线(其实 Q 的第一列就相当于由 M 的第一列单位化而得到的). 而由(4.30)容易算出

$$Me_1 = (\xi_1, \xi_2, \xi_3, 0, \dots, 0)^T,$$

其中

$$\begin{aligned} \xi_1 &= (h_{11}^{(k)})^2 + h_{12}^{(k)} h_{21}^{(k)} - s h_{11}^{(k)} + t, \\ \xi_2 &= h_{21}^{(k)} (h_{11}^{(k)} + h_{22}^{(k)} - s), \\ \xi_3 &= h_{21}^{(k)} h_{32}^{(k)}. \end{aligned}$$

其次, 如果Householder变换 P_0 将 Me_1 变为 αe_1 , $\alpha \in \mathbf{R}$, 则 P_0 的第一列就与 Me_1 共线, 从而 P_0 的第一列就可作为 Q 的第一列, 即 $P_0 e_1 = Q e_1$. 而由关于Householder变换的理论知, P_0 可以按如下方式确定:

$$P_0 = \text{diag}(\tilde{P}_0, I_{n-3}),$$

其中

$$\begin{aligned} \tilde{P}_0 &= I_3 - \beta v v^T, \quad v = \begin{bmatrix} \xi_1 - \alpha \\ \xi_2 \\ \xi_3 \end{bmatrix}, \\ \alpha &= (\xi_1^2 + \xi_2^2 + \xi_3^2)^{\frac{1}{2}}, \quad \beta = 2/v^T v, \end{aligned}$$

现今

$$B = P_0 H P_0,$$

则我们只要能够找到第一列为 e_1 的正交矩阵 \tilde{Q} 使 $\tilde{Q}^T H \tilde{Q} = \tilde{H}_2$ 为上Hessenberg矩阵, 那么 \tilde{H}_2 就是我们希望得到的 H_2 . 由本节所介绍的约化一个矩阵为上Hessenberg矩阵的方法可知, 这是容易办到的. 这只需确定 $n-1$ 个Householder变换 P_1, P_2, \dots, P_{n-1} 使

$$P_{n-1} \cdots P_1 B P_1 \cdots P_{n-1} = \tilde{H}$$

为上Hessenberg矩阵, 则 $\tilde{Q} = P_1 \cdots P_{n-1}$ 的第一列就为 e_1 . 而且由于 B 所具有的特殊性, 实现这一约化过程所需的运算量仅为 $O(n^2)$.

事实上, 由于用 P_0 将 H 相似变换为 B 只改变了 H 的前三行前三列, 故 B 有如下形状

$$B = P_0 H P_0 = \begin{bmatrix} \times & \times & \times & \times & \cdots & \times & \times \\ \times & \times & \times & \times & \cdots & \times & \times \\ + & \times & \times & \times & \cdots & \times & \times \\ + & + & \times & \times & \cdots & \times & \times \\ & & & \times & \cdots & \times & \times \\ & & 0 & & \ddots & \vdots & \vdots \\ & & & & & \times & \times \end{bmatrix},$$

仅比上Hessenberg形多三个可能的非零元素“+”. 由 B 的这种特殊性易知, 用来约化 B 为上Hessenberg形的第一个Householder变换 P_1 具有如下形状

$$P_1 = \text{diag}(1, \tilde{P}_1, I_{n-4}),$$

其中 \tilde{P}_1 为3阶Householder变换, 而且 $P_1 B P_1$ 具有如下形状

$$B = P_1 H P_1 = \begin{bmatrix} \times & \times & \times & \times & \times & \cdots & \times & \times \\ \times & \times & \times & \times & \times & \cdots & \times & \times \\ 0 & \times & \times & \times & \times & \cdots & \times & \times \\ 0 & + & \times & \times & \times & \cdots & \times & \times \\ 0 & + & + & \times & \times & \cdots & \times & \times \\ & & & & \times & \cdots & \times & \times \\ & & & 0 & & \ddots & \vdots & \vdots \\ & & & & & & \times & \times \end{bmatrix}.$$

一般地, 第 k 次约化所用的Householder变换 P_k 具有如下形状

$$P_k = \text{diag}(I_k, \tilde{P}_k, I_{n-k-3}),$$

其中 \tilde{P}_k 为3阶Householder变换, $k = 2, 3, \dots, n-3$, 而且 $P_{n-3} \cdots P_1 B P_1 \cdots P_{n-3}$ 具有如下形状

$$P_{n-3} \cdots P_1 B P_1 \cdots P_{n-3} = \begin{bmatrix} \times & \cdots & \times & \times & \times \\ \times & \cdots & \times & \times & \times \\ & & \ddots & \vdots & \vdots \\ & & & \times & \times & \times \\ 0 & & & + & \times & \times \end{bmatrix}.$$

因此, 最后一次约化所用的Householder变换 P_{n-2} 具有如下形状

$$P_{n-2} = \text{diag}(I_{n-2}, \tilde{P}_{n-2}),$$

其中 \tilde{P}_k 为2阶Householder变换.

综述上面的讨论, 就得到了著名的Francis双重步位移的QR迭代算法:

算法6.4.2 (双重步位移QR迭代)

$m = n - 1$

$s = H(m, m) + H(n, n)$

$t = H(m, m)H(n, n) - H(m, n)H(n, m)$

$x = H(1, 1)H(1, 1) + H(1, 2)H(2, 1) - sH(1, 1) + t$

$y = H(2, 1)(H(1, 1) + H(2, 2) - s)$

$z = H(2, 1)H(3, 2)$

for $k = 0 : n - 3$

$[v, \beta] = \mathbf{house}([x, y, z]^T)$

$q = \max\{1, k\}$

$H(k+1 : k+3, q : n) = (I - \beta v v^T)H(k+1 : k+3, q : n)$

$r = \min\{k+4, n\}$

$H(1 : r, k+1 : k+3) = H(1 : r, k+1 : k+3)(I - \beta v v^T)$

$x = H(k+2, k+1)$

```

    y = H(k + 3, k + 1)
    if k < n - 3
        z = H(k + 4, k + 1)
    end
end
end
[v, β] = house([x, y]T)
H(n - 1 : n, n - 2 : n) = (I - βvvT)H(n - 1 : n, n - 2 : n)
H(1 : n, n - 1 : n) = H(1 : n, n - 1 : n)(I - βvvT)

```

该算法的运算量为 $10n^2$; 如果需要累积正交变换, 则还需再增加运算量 $10n^2$.

4.6 隐式QR算法

前面的讨论已经解决了用QR方法求一个给定的实矩阵的实Schur标准形的几个关键的问题. 然而, 作为一种实用的算法, 还需给出一种有效的判定准则, 来判定迭代过程中所产生的上Hessenberg矩阵的次对角元何时可以忽略不计. 一种简单而实用的准则是, 当

$$|h_{i+1,i}| \leq (|h_{ii}| + |h_{i+1,i+1}|)\mathbf{u}$$

时, 就将 $h_{i+1,i}$ 看作零. 这样做的理由是, 在前面约化 A 为上Hessenberg矩阵时就已经引进了量级为 $\|A\|_F\mathbf{u}$ 的误差.

综合上面的讨论, 就得到如下的隐式QR算法. 该算法是计算一个给定的 n 阶实矩阵 A 的实Schur分解: $Q^T A Q = T$, 其中 Q 是正交矩阵, T 为拟上三角矩阵, 即对角块为 1×1 或 2×2 方阵的块上三角矩阵, 而且每个 2×2 的对角块必有一对复共轭特征值.

算法6.4.3 (计算实矩阵的实Schur分解: 隐式QR算法)

- (1) 输入 A .
- (2) 上Hessenberg化: 用算法6.4.1计算 A 的上Hessenberg分解, 得 $H = U_0^T A U_0$; $Q = U_0$.
- (3) 收敛性判定:
 - (i) 把所有满足条件

$$|h_{i,i-1}| \leq (|h_{ii}| + |h_{i-1,i-1}|)\mathbf{u}$$

的 $h_{i,i-1}$ 置零;

- (ii) 确定最大的非负整数 m 和最小的非负整数 l , 使

$$H = \begin{bmatrix} H_{11} & H_{12} & H_{13} \\ 0 & H_{22} & H_{23} \\ 0 & 0 & H_{33} \end{bmatrix} \begin{matrix} l \\ n-l-m \\ m \end{matrix},$$

其中 H_{33} 为拟上三角形, 而 H_{22} 为不可约的上Hessenberg形;

- (iii) 如果 $m = n$, 则输出有关信息, 结束; 否则进行下一步.
 - (4) QR迭代: 对 H_{22} 用算法6.4.2迭代一次得

$$H_{22} = P^T H_{22} P, \quad P = P_0 P_1 \cdots P_{n-m-l-2}.$$

(5) 计算

$$Q = Q \text{diag}(I_l, P, I_m), \quad H_{12} = H_{12}P, \quad H_{23} = P^T H_{23},$$

然后转步(3).

实际计算的统计表明, 这一算法每分离出一个 1×1 或 2×2 子矩阵平均约需2次QR迭代. 因此, 如果只计算特征值, 则运算量平均约为 $10n^3$; 如果 Q 和 T 都需要, 则运算量平均约为 $25n^3$.

例6.4.1 应用算法6.4.3于矩阵

$$H = \begin{bmatrix} 2 & 3 & 4 & 5 & 6 \\ 4 & 4 & 5 & 6 & 7 \\ 0 & 3 & 6 & 7 & 8 \\ 0 & 0 & 2 & 8 & 9 \\ 0 & 0 & 0 & 1 & 10 \end{bmatrix},$$

其次对角元收敛情况如下

迭代次数	$O(h_{21})$	$O(h_{32})$	$O(h_{43})$	$O(h_{54})$
1	10^0	10^0	10^0	10^0
2	10^0	10^0	10^0	10^0
3	10^0	10^0	10^{-1}	10^0
4	10^0	10^0	10^{-3}	10^{-3}
5	10^0	10^0	10^{-6}	10^{-5}
6	10^{-1}	10^0	10^{-13}	10^{-13}
7	10^{-1}	10^0	10^{-28}	10^{-13}
8	10^{-4}	10^0	收敛	收敛
9	10^{-8}	10^0		
10	10^{-8}	10^0		
11	10^{-16}	10^0		
12	10^{-32}	10^0		
13	收敛	收敛		