数值线性代数与算法

第七章 矩阵特征值问题的数值方法




Back
Close

许多工程实际问题的求解, 如振动问题、稳定性问题等, 最终都归结为求某些矩阵的特征值和特征向量的问题. n 阶方阵 $\boldsymbol{A} \in \mathbb{C}^{n\times n}$ 的特征值与特征向量, 是满足如下两个方程的数 $\lambda \in \mathbb{C}$ 和非零向量 $\boldsymbol{x} \in \mathbb{C}^n$:

$$p(\lambda) = \det(\boldsymbol{A} - \lambda \boldsymbol{I}) = 0, \tag{7.1}$$

$$\boldsymbol{A}\boldsymbol{x} = \lambda \boldsymbol{x} \quad \text{或} \quad (\boldsymbol{A} - \lambda \boldsymbol{I})\boldsymbol{x} = \boldsymbol{0}. \tag{7.2}$$

式 (7.1) 称为矩阵 $\boldsymbol{A}$ 的特征方程, $\boldsymbol{I}$ 是 n 阶单位阵, $\det(\boldsymbol{A} - \lambda \boldsymbol{I})$ 表示方阵 $\boldsymbol{A} - \lambda \boldsymbol{I}$ 的行列式, 它是 λ 的 n 次代数多项式, 当 n 较大时其零点难以准确求解. 因此, 从数值计算的观点来看, 用特征多项式来求矩阵特征值的方法并不可取, 必须建立有效的数值方法.

在实际应用中, 求矩阵的特征值和特征向量通常采用迭代法. 其基本思想是, 将特征值和特征向量作为一个无限序列的极限来求得. 舍入误差对这类方法的影响很小, 但通常计算量较大.




Back

Close

根据具体问题的需要, 有些实际问题只要计算模最大的特征值. 当然, 更多的问题则要求计算全部特征值和特征向量. 本章介绍几种目前在计算机上比较常用的矩阵特征值问题的数值方法.

§7.1 矩阵的特征值估计和隔离

在数值计算和其他学科中, 往往需要估计一个矩阵的特征值在复平面上的位置. 例如, 在研究一个迭代算法时, 需要判断迭代矩阵 (或迭代函数的 Jacobi 矩阵) 的特征值是否全部落在单位圆内. 又如, 分析一个非线性系统是否稳定时, 需要知道有关矩阵的特征值是否全部落在复平面的左半部. 本节扼要介绍一些这方面的结果.

首先讨论 Hermite 矩阵的特征值表示问题. 由于 Hermite 矩阵 $\boldsymbol{A} \in \mathbb{C}^{n\times n}$ 的特征值均为实数, 故可约定其 n 个特征值的排列次序

为:

$$\lambda_1 \geqslant \lambda_2 \geqslant \cdots \geqslant \lambda_n. \tag{7.3}$$

定义 7.1 设 $\boldsymbol{A} \in \mathbb{C}^{n\times n}$ 为 Hermite 矩阵, 对任意的非零向量 $\boldsymbol{x} \in \mathbb{C}^n$, 称

$$R(\boldsymbol{x}) = \frac{(\boldsymbol{A}\boldsymbol{x}, \boldsymbol{x})}{(\boldsymbol{x}, \boldsymbol{x})}$$

为 $\boldsymbol{x}$ 的 Rayleigh 商.

下面的定理说明了 Hermite 矩阵的特征值可以用 Rayleigh 商的极大值和极小值来描述.

定理 7.1 设矩阵 $\boldsymbol{A}^{\mathrm{H}} = \boldsymbol{A} \in \mathbb{C}^{n\times n}$, 则

$$\max_{\boldsymbol{0}\neq\boldsymbol{x}\in\mathbb{C}^n} \frac{(\boldsymbol{A}\boldsymbol{x}, \boldsymbol{x})}{(\boldsymbol{x}, \boldsymbol{x})} = \lambda_{\max}(\boldsymbol{A}), \qquad \min_{\boldsymbol{0}\neq\boldsymbol{x}\in\mathbb{C}^n} \frac{(\boldsymbol{A}\boldsymbol{x}, \boldsymbol{x})}{(\boldsymbol{x}, \boldsymbol{x})} = \lambda_{\min}(\boldsymbol{A}). \tag{7.4}$$

证明 设 $\boldsymbol{A}$ 的特征值为 $\lambda_1 \geqslant \lambda_2 \geqslant \cdots \geqslant \lambda_n$, 则存在酉矩阵 $\boldsymbol{Q}$ 使得

$$\boldsymbol{A} = \boldsymbol{Q}\mathrm{diag}(\lambda_1, \lambda_2, \cdots, \lambda_n)\boldsymbol{Q}^{\mathrm{H}},$$

故

$$\frac{(\boldsymbol{A}\boldsymbol{x}, \boldsymbol{x})}{(\boldsymbol{x}, \boldsymbol{x})} = \frac{(\boldsymbol{Q}\mathrm{diag}(\lambda_1, \lambda_2, \cdots, \lambda_n)\boldsymbol{Q}^{\mathrm{H}}\boldsymbol{x}, \boldsymbol{x})}{(\boldsymbol{Q}^{\mathrm{H}}\boldsymbol{x}, \boldsymbol{Q}^{\mathrm{H}}\boldsymbol{x})}$$
$$= \frac{(\mathrm{diag}(\lambda_1, \lambda_2, \cdots, \lambda_n)\boldsymbol{Q}^{\mathrm{H}}\boldsymbol{x}, \boldsymbol{Q}^{\mathrm{H}}\boldsymbol{x})}{(\boldsymbol{Q}^{\mathrm{H}}\boldsymbol{x}, \boldsymbol{Q}^{\mathrm{H}}\boldsymbol{x})}.$$

令

$$\boldsymbol{y} = \boldsymbol{Q}^{\mathrm{H}}\boldsymbol{x} = (y_1, y_2, \cdots, y_n)^{\mathrm{T}},$$

则

$$\frac{(\boldsymbol{A}\boldsymbol{x}, \boldsymbol{x})}{(\boldsymbol{x}, \boldsymbol{x})} = \frac{(\mathrm{diag}(\lambda_1, \lambda_2, \cdots, \lambda_n)\boldsymbol{y}, \boldsymbol{y})}{(\boldsymbol{y}, \boldsymbol{y})}$$
$$= \frac{\lambda_1 y_1^2 + \lambda_2 y_2^2 + \cdots + \lambda_n y_n^2}{y_1^2 + y_2^2 + \cdots + y_n^2}.$$

由于

$$\lambda_n(y_1^2+y_2^2+\cdots+y_n^2) \leqslant \lambda_1 y_1^2+\lambda_2 y_2^2+\cdots+\lambda_n y_n^2 \leqslant \lambda_1(y_1^2+y_2^2+\cdots+y_n^2),$$

故

$$\lambda_n \leqslant \frac{(\boldsymbol{A}\boldsymbol{x}, \boldsymbol{x})}{(\boldsymbol{x}, \boldsymbol{x})} \leqslant \lambda_1, \quad \forall\, \boldsymbol{0} \neq \boldsymbol{x} \in \mathbb{C}^n.$$

取 $\boldsymbol{x}_1$ 和 $\boldsymbol{x}_n$ 分别为对应于 λ_1 和 λ_n 的特征向量, 则

$$\frac{(\boldsymbol{A}\boldsymbol{x}_1, \boldsymbol{x}_1)}{(\boldsymbol{x}_1, \boldsymbol{x}_1)} = \lambda_1, \quad \frac{(\boldsymbol{A}\boldsymbol{x}_n, \boldsymbol{x}_n)}{(\boldsymbol{x}_n, \boldsymbol{x}_n)} = \lambda_n,$$

因此结论成立. 证毕. □

定理 7.2 $\boldsymbol{A}^{\mathrm{H}} = \boldsymbol{A} \in \mathbb{C}^{n\times n}$ 的特征值如式 (7.3), 则对 $1 \leqslant k \leqslant n$, 有

$$\lambda_k = \max_{\mathcal{V}_k} \min_{\boldsymbol{0}\neq\boldsymbol{x}\in\mathcal{V}_k} \frac{(\boldsymbol{A}\boldsymbol{x}, \boldsymbol{x})}{(\boldsymbol{x}, \boldsymbol{x})}, \tag{7.5}$$

式中: $\mathcal{V}_k$ 为 $\mathbb{C}^n$ 的任意一个 k 维子空间.

证明 设 $\boldsymbol{A}$ 的对应于特征值 $\lambda_1, \lambda_2, \cdots, \lambda_n$ 的特征向量依次为 $\boldsymbol{p}_1, \boldsymbol{p}_2, \cdots \boldsymbol{p}_n$, 且标准正交, 构造子空间 $\mathcal{W}_k$=span$\{\boldsymbol{p}_k, \boldsymbol{p}_{k+1}, \cdots, \boldsymbol{p}_n\}$,

那么 $\dim(\mathcal{W}_k) = n - k + 1$. 由于 $\mathcal{V}_k + \mathcal{W}_k \subset \mathbb{C}^n$, 利用子空间的维数公式求得

$$\begin{aligned} n \geqslant \dim(\mathcal{V}_k + \mathcal{W}_k) &= \dim(\mathcal{V}_k) + \dim(\mathcal{W}_k) - \dim(\mathcal{V}_k \cap \mathcal{W}_k) \\ &= n + 1 - \dim(\mathcal{V}_k \cap \mathcal{W}_k), \end{aligned}$$

即 $\dim(\mathcal{V}_k \cap \mathcal{W}_k) \geqslant 1$. 于是存在 $\boldsymbol{x}_0 \in \mathcal{V}_k \cap \mathcal{W}_k \subset \mathcal{W}_k$ 满足 $\|\boldsymbol{x}_0\|_2 = 1$, 且有

$$\boldsymbol{x}_0 = c_k \boldsymbol{p}_k + c_{k+1}\boldsymbol{p}_{k+1} + \cdots + c_n \boldsymbol{p}_n, \quad (c_k^2 + c_{k+1}^2 + \cdots + c_n^2 = 1),$$

$$(\boldsymbol{A}\boldsymbol{x}_0, \boldsymbol{x}_0) = \lambda_k c_k^2 + \lambda_{k+1} c_{k+1}^2 + \cdots + \lambda_n c_n^2 \leqslant \lambda_k.$$

因此

$$\min_{\boldsymbol{0} \neq \boldsymbol{x} \in \mathcal{V}_k} \frac{(\boldsymbol{A}\boldsymbol{x}, \boldsymbol{x})}{(\boldsymbol{x}, \boldsymbol{x})} \leqslant \frac{(\boldsymbol{A}\boldsymbol{x}_0, \boldsymbol{x}_0)}{(\boldsymbol{x}_0, \boldsymbol{x}_0)} \leqslant \lambda_k.$$




Back
Close

由 $\mathcal{V}_k$ 的任意性, 得

$$\max_{\mathcal{V}_k} \min_{\mathbf{0}\neq \boldsymbol{x}\in\mathcal{V}_k} \frac{(\boldsymbol{A}\boldsymbol{x},\boldsymbol{x})}{(\boldsymbol{x},\boldsymbol{x})} \leqslant \lambda_k.$$

另外, 取 k 维子空间 $\mathcal{V}_k = \mathrm{span}\{\boldsymbol{p}_1, \boldsymbol{p}_2, \cdots, \boldsymbol{p}_k\}$, 当 $\boldsymbol{x} \in \mathcal{V}_k$ 满足 $\|\boldsymbol{x}\|_2 = 1$ 时, 有

$$\boldsymbol{x} = c_1\boldsymbol{p}_1 + c_2\boldsymbol{p}_2 + \cdots + c_k\boldsymbol{p}_k, \quad (c_1^2 + c_2^2 + \cdots + c_k^2 = 1),$$

$$(\boldsymbol{A}\boldsymbol{x},\boldsymbol{x}) = \lambda_1 c_1^2 + \lambda_2 c_2^2 + \cdots + \lambda_k c_k^2 \geqslant \lambda_k.$$

由 $\boldsymbol{x} \in \mathcal{V}_k$ 的任意性, 得

$$\min_{\mathbf{0}\neq \boldsymbol{x}\in\mathcal{V}_k} \frac{(\boldsymbol{A}\boldsymbol{x},\boldsymbol{x})}{(\boldsymbol{x},\boldsymbol{x})} \geqslant \lambda_k.$$

从而有

$$\max_{\mathcal{V}_k} \min_{\mathbf{0}\neq \boldsymbol{x}\in\mathcal{V}_k} \frac{(\boldsymbol{A}\boldsymbol{x},\boldsymbol{x})}{(\boldsymbol{x},\boldsymbol{x})} \geqslant \lambda_k.$$

综上所述, 式 (7.5) 成立. 证毕. □

定理 7.3 设 n 阶 Hermite 矩阵 $\boldsymbol{A}$ 和 $\boldsymbol{B}=\boldsymbol{A}+\boldsymbol{E}$ 的特征值依次为

$$\lambda_1 \geqslant \lambda_2 \geqslant \cdots \geqslant \lambda_n, \quad \mu_1 \geqslant \mu_2 \geqslant \cdots \geqslant \mu_n.$$

则有

$$\varepsilon_n \leqslant \mu_k - \lambda_k \leqslant \varepsilon_1, \tag{7.6}$$

式中: ε_1 和 ε_n 分别为 Hermite 矩阵 $\boldsymbol{E}$ 的最大特征值和最小特征值.

证明 下面约定向量 $\boldsymbol{x}$ 满足 $\|\boldsymbol{x}\|_2=1$. 在式 (7.5) 中固定 $\mathcal{V}_k$, 利用式 (7.4), 得

$$\begin{aligned}\mu_k &\geqslant \min_{\boldsymbol{x}\in\mathcal{V}_k}(\boldsymbol{B}\boldsymbol{x},\boldsymbol{x}) \geqslant \min_{\boldsymbol{x}\in\mathcal{V}_k}(\boldsymbol{A}\boldsymbol{x},\boldsymbol{x}) + \min_{\boldsymbol{x}\in\mathcal{V}_k}(\boldsymbol{E}\boldsymbol{x},\boldsymbol{x}) \\ &\geqslant \min_{\boldsymbol{x}\in\mathcal{V}_k}(\boldsymbol{A}\boldsymbol{x},\boldsymbol{x}) + \min_{\boldsymbol{x}\in\mathbb{C}^n}(\boldsymbol{E}\boldsymbol{x},\boldsymbol{x}) = \min_{\boldsymbol{x}\in\mathcal{V}_k}(\boldsymbol{A}\boldsymbol{x},\boldsymbol{x}) + \varepsilon_n,\end{aligned}$$

$$\lambda_k \geqslant \min_{\boldsymbol{x}\in\mathcal{V}_k}(\boldsymbol{Ax},\boldsymbol{x}) \geq \min_{\boldsymbol{x}\in\mathcal{V}_k}(\boldsymbol{Bx},\boldsymbol{x}) + \min_{\boldsymbol{x}\in\mathcal{V}_k}(-\boldsymbol{Ex},\boldsymbol{x})$$

$$\geqslant \min_{\boldsymbol{x}\in\mathcal{V}_k}(\boldsymbol{Bx},\boldsymbol{x}) + \min_{\boldsymbol{x}\in\mathbb{C}^n}(-\boldsymbol{Ex},\boldsymbol{x}) = \min_{\boldsymbol{x}\in\mathcal{V}_k}(\boldsymbol{Bx},\boldsymbol{x}) + (-\varepsilon_1).$$

由于 $\mathcal{V}_k \subset \mathbb{R}^n$ 任意, 所以

$$\mu_k \geqslant \max_{\mathcal{V}_k}\min_{\boldsymbol{x}\in\mathcal{V}_k}(\boldsymbol{Ax},\boldsymbol{x}) + \varepsilon_n = \lambda_k + \varepsilon_n,$$

$$\lambda_k \geqslant \max_{\mathcal{V}_k}\min_{\boldsymbol{x}\in\mathcal{V}_k}(\boldsymbol{Bx},\boldsymbol{x}) - \varepsilon_1 = \mu_k - \varepsilon_1,$$

即式 (7.6) 成立. 证毕. □

注 7.1 定理 7.3 可用来讨论实对称矩阵的摄动问题. 可将 $\boldsymbol{E}$ 看作由于种种原因在矩阵 $\boldsymbol{A}$ 的元素中产生的误差构成的“误差矩阵”, 而 $\boldsymbol{B}$ 就是实际计算时的矩阵. 设对 $\boldsymbol{E}$ 的每个元素 e_{ij} 都有 $|e_{ij}| \leqslant \varepsilon$, 其中 $\varepsilon > 0$ 是某个常数, 则 $\boldsymbol{B}$ 的特征值 $\lambda(\boldsymbol{B}) = \lambda(\boldsymbol{A}) +$

$\Delta\lambda$ 与 $\boldsymbol{A}$ 的特征值 $\lambda(\boldsymbol{A})$ 之间满足 (7.6) 式, 即 $\varepsilon_1 \geqslant \Delta\lambda \geqslant \varepsilon_n$. 因此, 有

$$|\Delta\lambda| \leqslant \rho(\boldsymbol{E}) \leqslant \|\boldsymbol{E}\|_F \leqslant n\varepsilon.$$

这表明, 当 $\boldsymbol{A}$ 的诸元素有一个小扰动时, 在 $\boldsymbol{A}$ 的特征值中造成的扰动与其同阶.

定理 7.4 (分隔定理) 设 $\boldsymbol{A}$ 是 n 阶实对称矩阵, $\boldsymbol{B} = \boldsymbol{Q}^{\mathrm{T}}\boldsymbol{A}\boldsymbol{Q}$, 其中 $\boldsymbol{Q} \in \mathbb{R}^{n\times(n-1)}$ 满足 $\boldsymbol{Q}^{\mathrm{T}}\boldsymbol{Q} = \boldsymbol{I}_{n-1}$. 再设 $\boldsymbol{A}$ 和 $\boldsymbol{B}$ 的特征值分别为

$$\lambda_1 \geqslant \lambda_2 \geqslant \cdots \geqslant \lambda_n \quad \text{和} \quad \mu_1 \geqslant \mu_2 \geqslant \cdots \geqslant \mu_{n-1},$$

则有

$$\lambda_1 \geqslant \mu_1 \geqslant \lambda_2 \geqslant \mu_2 \geqslant \cdots \geqslant \mu_{n-1} \geqslant \lambda_n. \tag{7.7}$$

特别地, 在定理 7.4 中选取 $\boldsymbol{Q} = [\boldsymbol{e}_1, \cdots, \boldsymbol{e}_{i-1}, \boldsymbol{e}_{i+1}, \cdots, \boldsymbol{e}_n]$, 即得如下结论.




推论 7.1 设 $\boldsymbol{B}$ 是 n 阶实对称矩阵 $\boldsymbol{A}$ 的一个 $n-1$ 阶主子阵, 并假定 $\boldsymbol{A}$ 和 $\boldsymbol{B}$ 的特征值分别为

$$\lambda_1 \geqslant \lambda_2 \geqslant \cdots \geqslant \lambda_n \quad 和 \quad \mu_1 \geqslant \mu_2 \geqslant \cdots \geqslant \mu_{n-1},$$

则有

$$\lambda_1 \geqslant \mu_1 \geqslant \lambda_2 \geqslant \mu_2 \geqslant \cdots \geqslant \mu_{n-1} \geqslant \lambda_n.$$

反复应用推论 7.1, **便有以下推论**.

推论 7.2 设 $\boldsymbol{A}$ 是一个 n 阶实对称矩阵, $\boldsymbol{B}$ 是 $\boldsymbol{A}$ 的一个 k 阶主子阵 $(1 \leqslant k \leqslant n-1)$, 并假定 $\boldsymbol{A}$ 和 $\boldsymbol{B}$ 的特征值分别为

$$\lambda_1 \geqslant \lambda_2 \geqslant \cdots \geqslant \lambda_n \quad 和 \quad \mu_1 \geqslant \mu_2 \geqslant \cdots \geqslant \mu_k,$$

则有

$$\lambda_i \geqslant \mu_i \geqslant \lambda_{n-k+i}, \quad i = 1, 2, \cdots, k. \tag{7.8}$$

为了细致描述 n 阶矩阵的特征值在复平面上的分布范围, 下面引进 Gerschgorin 圆盘 (简称盖尔圆或盖氏圆).

定义 7.2 设 $\boldsymbol{A}=(a_{ij})\in\mathbb{C}^{n\times n}$, 令 $R_i=\sum\limits_{j=1,\,j\neq i}^{n}|a_{ij}|$, 则称

$$G_i=\{z|z\in\mathbb{C}:|z-a_{ii}|\leqslant R_i\},\quad i=1,2,\cdots n \tag{7.9}$$

为 $\boldsymbol{A}$ 的第 i 个盖氏圆.

定理 7.5 设 λ 是 $\boldsymbol{A}\in\mathbb{C}^{n\times n}$ 的任一特征值, 则 $\lambda\in\bigcup\limits_{i=1}^{n}G_i$, 即 $\boldsymbol{A}$ 的任一特征值都落在它的 n 个盖氏圆盘的并集内.

证明 设 $\boldsymbol{A}$ 的对应于特征值 λ 的特征向量为 $\boldsymbol{x}=(x_1,x_2,\cdots,x_n)^{\mathrm{T}}$. 选取 i_0 使得 $|x_{i_0}|=\max\limits_{1\leqslant i\leqslant n}|x_i|$, 则由 $\boldsymbol{Ax}=\lambda\boldsymbol{x}$ 可得

$$\sum_{j=1}^{n} a_{i_0 j}x_j = \lambda x_{i_0} \Longrightarrow (\lambda - a_{i_0 i_0})x_{i_0} = \sum_{j=1, j\neq i_0}^{n} a_{i_0 j}x_j$$

$$\Longrightarrow \ |\lambda - a_{i_0 i_0}| = \left| \sum_{j=1, j\neq i_0}^{n} a_{i_0 j}\frac{x_j}{x_{i_0}} \right| \leqslant R_{i_0},$$

即 $\lambda \in G_{i_0} \subset \bigcup_{i=1}^{n} G_i$. 证毕. □

定理 7.5 用一组圆盘覆盖矩阵的特征值分布区域, 下面介绍用另外一组几何图形覆盖矩阵的特征值分布区域的定理, 后者可以看作是前者的推广.

定理 7.6 设 λ 是 $\boldsymbol{A} \in C^{n\times n}\,(n > 1)$ 的任一特征值, 则 λ 位于某个

$$\Omega_{ij} = \left\{ z \,|\, z \in \mathbb{C}, |z - a_{ii}||z - a_{jj}| \leq R_i R_j,\ i \neq j;\ i, j = 1, 2, \cdots, n \right\}$$

之中, 称 $\Omega_{ij}(i \neq j)$ 为 $\boldsymbol{A}$ 的 Cassini (卡西尼) 卵形.

证明 设 $\boldsymbol{A}$ 的对应于特征值 λ 的特征值向量为 $\boldsymbol{x} = (x_1, x_2, \cdots, x_n)^{\mathrm{T}}$. 选取 $i_0 \neq j_0$ 满足 $|x_{i_0}| \geqslant |x_{j_0}| \geqslant |x_k| (k \neq i_0, j_0)$, 下证 $\lambda \in \Omega_{i_0 j_0}$.

(1) 如果 $x_{j_0} = 0$, 则 $x_{i_0} \neq 0,\ x_k = 0\ (k \neq i_0)$. 由 $\boldsymbol{Ax} = \lambda \boldsymbol{x}$ 可得

$$\lambda x_{i_0} = \sum_{k=1}^{n} a_{i_0 k} x_k = a_{i_0 i_0} x_{i_0} \Rightarrow \lambda = a_{i_0 i_0}.$$

故

$$\left|\lambda - a_{i_0 i_0}\right| \left|\lambda - a_{j_0 j_0}\right| = 0 \leq R_{i_0} R_{j_0}.$$

(2) 如果 $x_{j_0} \neq 0$, 则 $\left|x_{i_0}\right| \geqslant \left|x_{j_0}\right| > 0$, 再由 $\boldsymbol{Ax} = \lambda \boldsymbol{x}$ 可得

$$(\lambda - a_{ii}) x_i = \sum_{k \neq i} a_{ik} x_k,\ \ (i = 1, 2, \cdots, n).$$

取 $i = i_0$ 时, 可得

$$|\lambda - a_{i_0 i_0}||x_{i_0}| \leq \sum_{k \neq i_0} |a_{i_0 k}||x_k| \leq |x_{j_0}| R_{i_0}.$$

取 $i = j_0$ 时, 可得

$$|\lambda - a_{j_0 j_0}||x_{j_0}| \leq \sum_{k \neq j_0} |a_{j_0 k}||x_k| \leq |x_{i_0}| R_{j_0}.$$

因此 $|\lambda - a_{i_0 i_0}||\lambda - a_{j_0 j_0}| \leq R_{i_0} R_{j_0}$. 综述 (1) 和 (2) 即得 $\lambda \in \Omega_{i_0 j_0}$. 证毕. □

推论 7.3 设 $\boldsymbol{A} = (a_{ij}) \in \mathbb{C}^{n \times n} (n > 1)$ 满足 $|a_{ii}||a_{jj}| > R_i R_j \, (i \neq j)$, 则 $\det(\boldsymbol{A}) \neq 0$.

证明 设 λ 是 $\boldsymbol{A}$ 的任一特征值, 那么必有 Ω_{ij} 使得 $\lambda \in \Omega_{ij}$, 即

$$|\lambda - a_{ii}||\lambda - a_{jj}| \leq R_i R_j.$$

如果 $\lambda = 0$, 则有 $|a_{ii}||a_{jj}| \leq R_i R_j$, 这与题设矛盾, 故 $\lambda \neq 0$. 从而 $\det(\boldsymbol{A}) \neq 0$. 证毕. □

§7.2 幂法和反幂法

§7.2.1 幂法

幂法是通过求矩阵的特征向量来求出特征值的一种迭代法. 它主要用来求按模最大的特征值和相应的特征向量的. 其优点是算法简单, 容易计算机实现, 缺点是收敛速度慢, 其有效性依赖于矩阵特征值的分布情况.

适于使用幂法的常见情形是: $\boldsymbol{A}$ 的特征值可按模的大小排列为 $|\lambda_1| > |\lambda_2| \geqslant \cdots \geqslant |\lambda_n|$, 且其对应特征向量 $\boldsymbol{\xi}_1, \boldsymbol{\xi}_2, \cdots, \boldsymbol{\xi}_n$ 线性无关. 此时, 任意非零向量 $\boldsymbol{x}^{(0)}$ 均可用 $\boldsymbol{\xi}_1, \boldsymbol{\xi}_2, \cdots, \boldsymbol{\xi}_n$ 线性表示, 即

$$\boldsymbol{x}^{(0)} = \alpha_1 \boldsymbol{\xi}_1 + \alpha_2 \boldsymbol{\xi}_2 + \cdots + \alpha_n \boldsymbol{\xi}_n, \tag{7.10}$$

且 $\alpha_1,\ \alpha_2, \cdots, \alpha_n$ 不全为零. 作向量序列 $\boldsymbol{x}^{(k)} = \boldsymbol{A}^k \boldsymbol{x}^{(0)}$, 则

$$\begin{aligned}\boldsymbol{x}^{(k)} &= \boldsymbol{A}^k\boldsymbol{x}^{(0)} = \alpha_1\boldsymbol{A}^k\boldsymbol{\xi}_1 + \alpha_2\boldsymbol{A}^k\boldsymbol{\xi}_2 + \cdots + \alpha_n A^k\boldsymbol{\xi}_n \\ &= \alpha_1\lambda_1^k\boldsymbol{\xi}_1 + \alpha_2\lambda_2^k\boldsymbol{\xi}_2 + \cdots + \alpha_n\lambda_n^k\boldsymbol{\xi}_n \\ &= \lambda_1^k\left[\alpha_1\boldsymbol{\xi}_1 + \alpha_2\left(\frac{\lambda_2}{\lambda_1}\right)^k\boldsymbol{\xi}_2 + \cdots + \alpha_n\left(\frac{\lambda_n}{\lambda_1}\right)^k\boldsymbol{\xi}_n\right].\end{aligned}$$

由此可见, 若 $\alpha_1 \neq 0$, 则因 $k \to \infty$ 时, 有

$$\left(\frac{\lambda_i}{\lambda_1}\right)^k \to 0, \quad (i = 2, \cdots, n),$$

故当 k 充分大时, 必有

$$\boldsymbol{x}^{(k)} \approx \lambda_1^k\alpha_1\boldsymbol{\xi}_1,$$

即 $\boldsymbol{x}^{(k)}$ 可以近似看成 λ_1 对应的特征向量; 而 $\boldsymbol{x}^{(k)}$ 与 $\boldsymbol{x}^{(k-1)}$ 分量之比为

$$\frac{\boldsymbol{x}_i^{(k)}}{\boldsymbol{x}_i^{(k-1)}} \approx \frac{\lambda_1^k\alpha_1(\boldsymbol{\xi}_1)_i}{\lambda_1^{k-1}\alpha_1(\boldsymbol{\xi}_1)_i} = \lambda_1.$$

Back

Close

于是利用向量序列 $\{\boldsymbol{x}^{(k)}\}$ 既可求出按模最大的特征值 λ_1, 又可求出对应的特征向量 $\boldsymbol{\xi}_1$.

在实际计算中, 考虑到当 $|\lambda_1| > 1$ 时, $\lambda_1^k \to \infty$; 当 $|\lambda_1| < 1$ 时, $\lambda_1^k \to 0$, 因而计算 $\boldsymbol{x}^{(k)}$ 时可能会导致计算机“上溢”或“下溢”现象发生, 故采取每步将 $\boldsymbol{x}^{(k)}$ 归一化处理的办法, 即将 $\boldsymbol{x}^{(k)}$ 的各分量都除以模最大的分量, 使 $\|\boldsymbol{x}^{(k)}\|_\infty = 1$. 于是, 求 $\boldsymbol{A}$ 按模最大的特征值 λ_1 和对应的特征向量 $\boldsymbol{\xi}_1$ 的算法, 可归纳为如下步骤.

算法 7.1 (幂法)

步 1, 输入矩阵 $\boldsymbol{A}$, 初始向量 $\boldsymbol{v}^{(0)}$, 误差限 ε, 最大迭代次数 N. 记 m_0 是 $\boldsymbol{v}^{(0)}$ 按模最大的分量, $\boldsymbol{x}^{(0)} = \boldsymbol{v}^{(0)}/m_0$. 置 $k := 0$.

步 2, 计算 $\boldsymbol{v}^{(k+1)} = \boldsymbol{A}\boldsymbol{x}^{(k)}$. 记 m_{k+1} 是 $\boldsymbol{v}^{(k+1)}$ 按模最大的分量, $\boldsymbol{x}^{(k+1)} = \boldsymbol{v}^{(k+1)}/m_{k+1}$.

步 3, 若 $|m_{k+1} - m_k| < \varepsilon$, 停算, 输出近似特征值 m_{k+1} 和近

似特征向量 $\boldsymbol{x}^{(k+1)}$; 否则, 转步 4.

步 4, 若 $k < N$, 置 $k := k + 1$, 转步 2; 否则输出计算失败信息, 停算.

算法 7.1 称为幂法. **幂法的算法结构简单**, **容易编程实现**. **其** MATLAB **程序如下**:

```
function [lam,v,k]=mypower(A,x,tol,N)
%用幂法计算矩阵的模最大特征值和对应的特征向量
%输入:A为n阶方阵,x为初始向量,
%      tol为控制精度,N为最大迭代次数
%输出:lam为按模最大的特征值,
%      v为对应的特征向量,k为迭代次数
if nargin<4, N=1000; end
if nargin<3, tol=1e-6; end
```

```
m=0; k=0;
while(k<N)
    v=A*x;
    [m1,t]=max(abs(v));
    m1=v(t); x=v/m1;
    err=abs(m1-m);
    if err<tol, break; end
    m=m1; k=k+1;
end
lam=m1; v=x;
```

例 7.1 利用程序 mypower.m 求矩阵 $\boldsymbol{A}$ 按模最大的特征值 λ_1 和对应的特征向量 $\boldsymbol{\xi}_1$, 其中

$$\boldsymbol{A}=\begin{bmatrix} 1 & -1 & & & \\ -1 & 2 & -1 & & \\ & \ddots & \ddots & \ddots & \\ & & -1 & 99 & -1 \\ & & & -1 & 100 \end{bmatrix}.$$

解 用幂法的 MATLAB 程序, 编写 MATLAB 脚本文件 ex71.m, 取容许误差为 $\varepsilon = 10^{-6}$, 在命令窗口运行之, 迭代 661 次得到模最大的特征值 $\lambda_1 = 100.7461$.

下面证明算法 7.1 的收敛性定理.

定理 7.7 设矩阵 $\boldsymbol{A}$ 的特征值可按模的大小排列为 $|\lambda_1| > |\lambda_2| \geqslant \cdots \geqslant |\lambda_n|$, 且其对应特征向量 $\boldsymbol{\xi}_1, \boldsymbol{\xi}_2, \cdots, \boldsymbol{\xi}_n$ 线性无关. 序列 $\{\boldsymbol{x}^{(k)}\}$ 由算法 7.1 产生, 则有

$$\lim_{k\to\infty} \boldsymbol{x}^{(k)} = \frac{\boldsymbol{\xi}_1}{\max\{\boldsymbol{\xi}_1\}} := \boldsymbol{\xi}_1^0, \qquad \lim_{k\to\infty} m_k = \lambda_1, \tag{7.11}$$

式中: $\boldsymbol{\xi}_1^0$ 为将 $\boldsymbol{\xi}_1$ 归一化后得到的向量; $\max\{\boldsymbol{\xi}_1\}$ 为向量 $\boldsymbol{\xi}_1$ 模最大的分量.

证明 由算法 7.1 的步 2 和步 3 知

$$\boldsymbol{x}^{(k)} = \frac{\boldsymbol{v}^{(k)}}{m_k} = \frac{\boldsymbol{A}\boldsymbol{x}^{(k-1)}}{m_k} = \frac{\boldsymbol{A}^2\boldsymbol{x}^{(k-2)}}{m_k m_{k-1}} = \cdots = \frac{\boldsymbol{A}^k\boldsymbol{x}^{(0)}}{m_k m_{k-1}\cdots m_1}.$$

由于 $\boldsymbol{x}^{(k)}$ 的最大分量为 1, 即 $\max\{\boldsymbol{x}^{(k)}\} = 1$, 故

$$m_k m_{k-1}\cdots m_1 = \max\{\boldsymbol{A}^k\boldsymbol{x}^{(0)}\}.$$

从而

$$\boldsymbol{x}^{(k)} = \frac{\boldsymbol{A}^k\boldsymbol{x}^{(0)}}{\max\{\boldsymbol{A}^k\boldsymbol{x}^{(0)}\}} = \frac{\lambda_1^k\left[\alpha_1\boldsymbol{\xi}_1 + \sum_{i=2}^{n}\alpha_i\left(\frac{\lambda_i}{\lambda_1}\right)^k\boldsymbol{\xi}_i\right]}{\max\left\{\lambda_1^k\left[\alpha_1\boldsymbol{\xi}_1 + \sum_{i=2}^{n}\alpha_i\left(\frac{\lambda_i}{\lambda_1}\right)^k\boldsymbol{\xi}_i\right]\right\}}$$

$$= \frac{\alpha_1\boldsymbol{\xi}_1 + \sum_{i=2}^{n}\alpha_i\left(\frac{\lambda_i}{\lambda_1}\right)^k\boldsymbol{\xi}_i}{\max\left\{\alpha_1\boldsymbol{\xi}_1 + \sum_{i=2}^{n}\alpha_i\left(\frac{\lambda_i}{\lambda_1}\right)^k\boldsymbol{\xi}_i\right\}}.$$

可见

$$\lim_{k\to\infty} \boldsymbol{x}^{(k)} = \frac{\alpha_1\boldsymbol{\xi}_1}{\max\{\alpha_1\boldsymbol{\xi}_1\}} = \frac{\boldsymbol{\xi}_1}{\max\{\boldsymbol{\xi}_1\}} = \boldsymbol{\xi}_1^0.$$

又

$$\boldsymbol{v}^{(k)} = \boldsymbol{A}\boldsymbol{x}^{(k-1)} = \frac{\boldsymbol{A}^k\boldsymbol{x}^{(0)}}{m_{k-1}\cdots m_1} = \frac{\boldsymbol{A}^k\boldsymbol{x}^{(0)}}{\max\{\boldsymbol{A}^{k-1}\boldsymbol{x}^{(0)}\}}$$

$$= \frac{\lambda_1^k\left[\alpha_1\boldsymbol{\xi}_1 + \sum_{i=2}^{n}\alpha_i\left(\frac{\lambda_i}{\lambda_1}\right)^k\boldsymbol{\xi}_i\right]}{\lambda_1^{k-1}\max\left\{\alpha_1\boldsymbol{\xi}_1 + \sum_{i=2}^{n}\alpha_i\left(\frac{\lambda_i}{\lambda_1}\right)^{k-1}\boldsymbol{\xi}_i\right\}},$$

注意到 m_k 是 $\boldsymbol{v}^{(k)}$ 模最大的分量, 即有

$$m_k = \max\{\boldsymbol{v}^{(k)}\} = \lambda_1\frac{\max\left\{\alpha_1\boldsymbol{\xi}_1 + \sum_{i=2}^{n}\alpha_i\left(\frac{\lambda_i}{\lambda_1}\right)^k\boldsymbol{\xi}_i\right\}}{\max\left\{\alpha_1\boldsymbol{\xi}_1 + \sum_{i=2}^{n}\alpha_i\left(\frac{\lambda_i}{\lambda_1}\right)^{k-1}\boldsymbol{\xi}_i\right\}},$$

从而 $\lim\limits_{k\to\infty} m_k = \lambda_1$ 成立. 证毕. □

定理 7.8 在定理 7.7 的条件下, 幂法 (算法 7.1) 是线性收敛的.

证明 设 k 充分大时, $\boldsymbol{A}^k\boldsymbol{x}^{(0)}$ 的按模最大分量是它的第 j 个分量, 则

$$\begin{aligned} m_k - \lambda_1 &= \max\{\boldsymbol{v}^{(k)}\} - \lambda_1 = \frac{\max\{\boldsymbol{A}^k\boldsymbol{x}^{(0)}\}}{\max\{\boldsymbol{A}^{k-1}\boldsymbol{x}^{(0)}\}} - \lambda_1 \\ &= \frac{\left[\beta_1\lambda_1^k\boldsymbol{\xi}_1 + \beta_2\lambda_2^k\boldsymbol{\xi}_2 + \cdots + \beta_n\lambda_n^k\boldsymbol{\xi}_n\right]_j}{\left[\beta_1\lambda_1^{k-1}\boldsymbol{\xi}_1 + \beta_2\lambda_2^{k-1}\boldsymbol{\xi}_2 + \cdots + \beta_n\lambda_n^{k-1}\boldsymbol{\xi}_n\right]_j} - \lambda_1 \\ &= \frac{\left[\beta_2\lambda_2^{k-1}(\lambda_2-\lambda_1)\boldsymbol{\xi}_2 + \cdots + \beta_n\lambda_n^{k-1}(\lambda_n-\lambda_1)\boldsymbol{\xi}_n\right]_j}{\left[\beta_1\lambda_1^{k-1}\boldsymbol{\xi}_1 + \beta_2\lambda_2^{k-1}\boldsymbol{\xi}_2 + \cdots + \beta_n\lambda_n^{k-1}\boldsymbol{\xi}_n\right]_j}. \end{aligned}$$

于是有

$$
\begin{aligned}
m_k - \lambda_1 &= \left(\frac{\lambda_2}{\lambda_1}\right)^{k-1} \frac{\left[\beta_2(\lambda_2-\lambda_1)\boldsymbol{\xi}_2 + \sum_{i=3}^{n} \beta_i \left(\frac{\lambda_i}{\lambda_2}\right)^{k-1} (\lambda_i-\lambda_1)\boldsymbol{\xi}_i\right]_j}{\left[\beta_1\boldsymbol{\xi}_1 + \sum_{i=2}^{n} \beta_i \left(\frac{\lambda_i}{\lambda_1}\right)^{k-1} \boldsymbol{\xi}_i\right]_j} \\
&= \left(\frac{\lambda_2}{\lambda_1}\right)^{k-1} M_k, \quad M_k \to M,
\end{aligned}
$$

式中: M 为常数. 所以, 当 $k \to \infty$时, 有

$$
\frac{|m_{k+1} - \lambda_1|}{|m_k - \lambda_1|} = \left|\frac{M_{k+1}(\lambda_2/\lambda_1)^k}{M_k(\lambda_2/\lambda_1)^{k-1}}\right| \to \left|\frac{\lambda_2}{\lambda_1}\right|,
$$

这就证明了幂法的线性收敛速度. 证毕. □

定理 7.8 表明, 幂法的收敛速度与比值 $|\lambda_2/\lambda_1|$ 的大小有关, $|\lambda_2/\lambda_1|$ 越小, 收敛速度越快, 当此比值接近于 1 时, 收敛速度是非

常缓慢的. 因此, 可以对矩阵作一原点位移, 令

$$\boldsymbol{B} = \boldsymbol{A} - \alpha \boldsymbol{I},$$

式中: α 为参数. 选择此参数可使矩阵 $\boldsymbol{B}$ 的上述比值更小, 以加快幂法的收敛速度. 设矩阵 $\boldsymbol{A}$ 的特征值为 $\lambda_1, \lambda_2, \cdots, \lambda_n$, 对应的特征向量为 $\boldsymbol{\xi}_1, \boldsymbol{\xi}_2, \cdots, \boldsymbol{\xi}_n$, 则矩阵 $\boldsymbol{B}$ 的特征值为 $\lambda_1-\alpha, \lambda_2-\alpha, \cdots, \lambda_n-\alpha$, $\boldsymbol{B}$ 的特征向量和 $\boldsymbol{A}$ 的特征向量相同. 假设原点位移后, $\boldsymbol{B}$ 的特征值 $\lambda_1 - \alpha$ 仍为模最大的特征值, 选择 α 的目的是使

$$\max_{2 \leqslant i \leqslant n} \frac{|\lambda_i - \alpha|}{|\lambda_1 - \alpha|} < \left| \frac{\lambda_2}{\lambda_1} \right|. \tag{7.12}$$

适当地选择 α 可使幂法的收敛速度得到加速. 此时 $m_k \to \lambda_1 - \alpha$, $m_k + \alpha \to \lambda_1$, 而 $\boldsymbol{x}^{(k)}$ 仍然收敛于 $\boldsymbol{A}$ 的特征向量 $\boldsymbol{\xi}_1^0$. 这种加速收敛的方法称为*原点位移法*.

在实际计算中, 由于矩阵的特征值分布情况事先一般是不知

道的, 参数 α 的选取存在困难, 因为 α 的选取要保证 $\lambda_1 - \alpha$ 仍然是矩阵 $\boldsymbol{B}\,(= \boldsymbol{A} - \alpha \boldsymbol{I})$ 模最大的特征值, 故原点位移法是很难实现的. 但是, 在反幂法中, 原点位移参数 α 是非常容易选取的, 因此, 带原点位移的反幂法已成为改进特征值和特征向量精度的标准算法. 采用原点位移加速技术的幂法 MATLAB 程序如下:

```
function [lam,v,k]=mopower(A,x,alpha,tol,N)
%用原点位移幂法求矩阵的模最大特征值和对应的特征向量
%输入:A为n阶方阵,x为初始向量,tol为控制精度,
%      N最大迭代次数,alpha为原点位移参数
%输出:lam返回按模最大的特征值,
%      v返回对应的特征向量,k返回迭代次数
if nargin<5, N=1000; end
if nargin<4, tol=1e-6; end
```

```
m=0; k=0;
A=A-alpha*eye(length(x));
while(k<N)
    v=A*x;
    [m1,t]=max(abs(v));
    m1=v(t); x=v/m1;
    err=abs(m1-m);
    if err<tol, break; end
    m=m1; k=k+1;
end
lam=m1+alpha;v=x;
```

例 7.2 利用原点位移幂法通用程序, 取 $\alpha = 50$, 求例 7.1 中的矩阵 $\boldsymbol{A}$ 按模最大的特征值 λ_1 和对应的特征向量 $\boldsymbol{\xi}_1$.

解 用原点位移幂法的 MATLAB 程序, 编写 M 文件 ex72.m, 在命令窗口运行该文件, 迭代 353 次得到模最大的特征值 $\lambda_1 = 100.7461$.

由计算结果可以看出, 在同样的精度控制下, 带原点位移加速的幂法只需要迭代 353 次, 而纯粹的幂法则需要迭代 661 次 (例 7.1).

§7.2.3 反幂法

设 $\boldsymbol{A}$ 可逆, 则对 $\boldsymbol{A}$ 的逆阵 $\boldsymbol{A}^{-1}$ 施以幂法称为*反幂法*. 由于 $\boldsymbol{A}\boldsymbol{\xi}_i = \lambda_i\boldsymbol{\xi}_i$ 时, 成立 $\boldsymbol{A}^{-1}\boldsymbol{\xi}_i = \lambda_i^{-1}\boldsymbol{\xi}_i$. 因此, 若 $|\lambda_1| \geqslant |\lambda_2| \geqslant \cdots \geqslant |\lambda_{n-1}| > |\lambda_n|$, 则 λ_n^{-1} 是 $\boldsymbol{A}^{-1}$ 按模最大的特征值, 此时按反幂法, 必有

$$m_k \to \lambda_n^{-1}, \quad \boldsymbol{x}^{(k)} \to \boldsymbol{\xi}_n^0,$$




Back
Close

且其收敛率为 $|\lambda_n/\lambda_{n-1}|$. 任取初始向量 $\boldsymbol{x}^{(0)}$, 构造向量序列

$$\boldsymbol{x}^{(k+1)} = \boldsymbol{A}^{-1}\boldsymbol{x}^{(k)}, \quad k = 0, 1, 2, \cdots \tag{7.13}$$

按幂法计算即可. 但用式 (7.13) 计算, 首先要求 $\boldsymbol{A}^{-1}$, 这比较麻烦而且是不经济的. 实际计算中, 通常用解方程组的办法, 即用

$$\boldsymbol{A}\boldsymbol{x}^{(k+1)} = \boldsymbol{x}^{(k)}, \quad k = 0, 1, 2, \cdots, \tag{7.14}$$

求 $\boldsymbol{x}^{(k+1)}$. 为防止计算机溢出, 实际计算时所用的公式为

$$\begin{cases} \boldsymbol{v}^{(k)} = \boldsymbol{x}^{(k)} / \max(\boldsymbol{x}^{(k)}), \\ \boldsymbol{A}\boldsymbol{x}^{(k+1)} = \boldsymbol{v}^{(k)}, \end{cases} \quad k = 0, 1, 2, \cdots, \tag{7.15}$$

式中: $\max(\boldsymbol{x}^{(k)})$ 为 $\boldsymbol{x}^{(k)}$ 模最大的分量.

反幂法主要用于已知矩阵的近似特征值为 α 时, 求矩阵的特征向量并提高特征值的精度. 此时, 可以用原点位移法来加速迭代

过程, 于是式 (7.15) 相应为

$$\begin{cases} \boldsymbol{v}^{(k)} = \boldsymbol{x}^{(k)} / \max(\boldsymbol{x}^{(k)}), \\ (\boldsymbol{A} - \alpha \boldsymbol{I})\boldsymbol{x}^{(k+1)} = \boldsymbol{v}^{(k)}, \end{cases} \quad k = 0, 1, 2, \cdots . \tag{7.16}$$

反幂法的计算步骤如下.

算法 7.2 (反幂法)

步 1, 选取初值 $\boldsymbol{x}^{(0)}$, 近似值 α, 误差限 ε, 最大迭代次数 N. 记 m_0 为 $\boldsymbol{x}^{(0)}$ 中按模最大的分量, $\boldsymbol{v}^{(0)} = \boldsymbol{x}^{(0)}/m_0$. 置 $k := 0$.

步 2, 解方程组 $(\boldsymbol{A} - \alpha \boldsymbol{I})\boldsymbol{x}^{(k+1)} = \boldsymbol{v}^{(k)}$ 得 $\boldsymbol{x}^{(k+1)}$.

步 3, 记 m_{k+1} 为 $\boldsymbol{x}^{(k+1)}$ 中按模最大的分量, $\boldsymbol{v}^{(k+1)} = \boldsymbol{x}^{(k+1)}/m_{k+1}$.

步 4, 若 $|m_{k+1}^{-1} - m_k^{-1}| < \varepsilon$, 则置 $\lambda := m_{k+1}^{-1} + \alpha$, 输出 λ 和 $\boldsymbol{x}^{(k+1)}$, 停算; 否则, 转步 5.

步 5, 若 $k < N$, 置 $k := k + 1$, 转步 2, 否则输出计算失败信息, 停算.

注 7.2 (1) 算法 7.2 计算出与数 α 最接近的特征值及相应的特征向量. 若取 $\alpha = 0$, 则求出 $\boldsymbol{A}$ 的按模最小的特征值.

(2) 通常首先用幂法求出 $\boldsymbol{A}$ 的按模最大的近似特征值作为算法 7.2 中的 α 值, 再使用该算法对 α 和相应的特征向量进行精确化.

(3) 为节省计算量, 通常先用列主元 LU 分解将矩阵 $\boldsymbol{A} - \alpha \boldsymbol{I}$ 分解为下三角矩阵 $\boldsymbol{L}$ 和上三角矩阵 $\boldsymbol{U}$, 这样在迭代过程中每一步就只需求解两个三角方程组即可.

反幂法的 MATLAB **程序如下**:

```
function [lam,v,k]=mvpower(A,x,alpha,tol,N)
```

```
%用反幂法计算矩阵与alpha最接近的特征值和对应的特征向量
%输入:A为n阶方阵,x为初始向量,tol为精度,
%      N为最大迭代数,alpha为某个常数
%输出:lam返回与alpha最接近的特征值,
%      v返回对应的特征向量,k返回迭代次数
if nargin<5, N=500; end
if nargin<4, tol=1e-5; end
m=0.5; k=0;
A=A-alpha*eye(length(x));
[L,U,P]=lu(A);
while (k<N)
   [m1,t]=max(abs(x));
   m1=x(t); v=x/m1;
```

```
    z=L\(P*v); x=U\z;
    err=abs(1/m1-1/m);
    if err<=tol, break; end
    k=k+1; m=m1;
end
lam=alpha+1/m;
```

例 7.3 利用反幂法程序, 求例 7.1 中的矩阵 $\boldsymbol{A}$ 最接近 101, 99, 2 和 0 的特征值和对应的特征向量.

解 注意到此处 α 的值分别取 $101, 99, 2, 0$. 用反幂法的 MATLAB 程序, 编写 M 文件 ex73.m, 取容许误差为 $\varepsilon = 10^{-6}$, 在命令窗口运行该文件, 得到 4 个近似特征值: 100.7462; 99.2107; 1.7893; 0.2538, 这是矩阵 $\boldsymbol{A}$ 的两个模最大和两个模最小的特征值.

§7.3 Jacobi 方法

Jacobi 方法用于求解实对称矩阵的全部特征值和对应的特征向量. 其数学原理如下:

(1) n 阶实对称矩阵的特征值全为实数, 其对应的特征向量线性无关且两两正交.

(2) 相似矩阵具有相同的特征值.

(3) 若 n 阶实矩阵 $\boldsymbol{A}$ 是对称的, 则存在正交矩阵 $\boldsymbol{Q}$, 使得 $\boldsymbol{Q}^{\mathrm{T}}\boldsymbol{A}\boldsymbol{Q}=\boldsymbol{D}$, 其中 $\boldsymbol{D}$ 是一个对角矩阵, 它的对角元素 $\lambda_1,\lambda_2,\cdots,\lambda_n$ 就是 $\boldsymbol{A}$ 的特征值, $\boldsymbol{Q}$ 的第 i 列向量就是 λ_i 对应的特征向量.

Jacobi 方法就是基于上述原理, 用一系列正交变换对角化 $\boldsymbol{A}$, 即逐步消去 $\boldsymbol{A}$ 的非对角元, 从而得到 $\boldsymbol{A}$ 的全部特征值.

§7.3.1 实对称矩阵的旋转正交相似变换

首先回顾一下第 2 章介绍过的一种正交变换—Givens 变换, 它是 Jacobi 方法的基本工具.

定义 7.3 设 $1 \leqslant i < j \leqslant n$, 则称矩阵

$$\boldsymbol{G}_{ij} = \begin{bmatrix} 1 & & & & & & & & \\ & \ddots & & & & & & & \\ & & \cos\varphi & & & & \sin\varphi & & \\ & & & 1 & & & & & \\ & & & & \ddots & & & & \\ & & & & & 1 & & & \\ & & -\sin\varphi & & & & \cos\varphi & & \\ & & & & & & & \ddots & \\ & & & & & & & & 1 \end{bmatrix} \begin{matrix} \\ \\ i \\ \\ \\ \\ j \\ \\ \\ \end{matrix} \tag{7.17}$$

$$\begin{matrix} & & i & & j & & & & \end{matrix}$$

为 (i,j) 平面的旋转矩阵, 或 Givens 变换矩阵.

显然, $\boldsymbol{G} = \boldsymbol{G}_{ij}$ 为正交矩阵, 即 $\boldsymbol{G}^{\mathrm{T}}\boldsymbol{G} = \boldsymbol{I}$. 对于向量 $\boldsymbol{x} \in \mathbb{R}^n$,




Back
Close

由线性变换 $\boldsymbol{y}=\boldsymbol{G}\boldsymbol{x}$ 得到的 $\boldsymbol{y}$ 的分量为

$$\begin{cases} y_i = x_i\cos\varphi + x_j\sin\varphi, \\ y_j = -x_i\sin\varphi + x_j\cos\varphi, \\ y_k = x_k, \quad k\neq i,j, \end{cases} \tag{7.18}$$

即用 $\boldsymbol{G}_{ij}$ 对向量 $\boldsymbol{x}$ 作用, 只改变其第 i,j 两个分量.

由矩阵 $\boldsymbol{G}=\boldsymbol{G}_{ij}$ 确定的正交变换 $\boldsymbol{y}=\boldsymbol{G}\boldsymbol{x}$ 称为平面旋转变换, 或 Givens 变换. 根据式 (7.18) 容易验证, 矩阵 $\boldsymbol{G}_{ij}$ 具有下列基本性质.

定理 7.9 设 $\boldsymbol{x}\in\mathbb{R}^n$ 的第 j 个分量 $x_j\neq 0$, $1\leqslant i<j\leqslant n$. 若令

$$c=\cos\varphi=\frac{x_i}{\sqrt{x_i^2+x_j^2}}, \qquad s=\sin\varphi=\frac{x_j}{\sqrt{x_i^2+x_j^2}}, \tag{7.19}$$

则 $\boldsymbol{y}=\boldsymbol{G}_{ij}\boldsymbol{x}$ 的分量为

$$\begin{cases} y_i=\sqrt{x_i^2+x_j^2}, \quad y_j=0, \\ y_k=x_k, \quad k\neq i,j. \end{cases} \tag{7.20}$$

定理 7.9 表明, 可以用 Givens 变换将向量的某个分量变为零元素.

下面讨论 Givens 变换对实对称矩阵的作用. 用旋转矩阵 $\boldsymbol{G}_{ij}$ 对实对称矩阵 $\boldsymbol{A}=(a_{ij})_{n\times n}$ 作正交相似变换, 所得矩阵记为 $\boldsymbol{A}_1$, 即

$$\boldsymbol{A}_1=\boldsymbol{G}_{ij}\boldsymbol{A}\boldsymbol{G}_{ij}^{\mathrm{T}}=(a_{ij}^{(1)}).$$

显然

$$\boldsymbol{A}_1^{\mathrm{T}}=(\boldsymbol{G}_{ij}\boldsymbol{A}\boldsymbol{G}_{ij}^{\mathrm{T}})^{\mathrm{T}}=\boldsymbol{G}_{ij}\boldsymbol{A}\boldsymbol{G}_{ij}^{\mathrm{T}}=\boldsymbol{A}_1,$$

Back

Close

即 $\boldsymbol{A}_1$ 仍为实对称矩阵. 直接计算, 得

$$\begin{cases} a_{ii}^{(1)} = a_{ii}\cos^2\varphi + a_{jj}\sin^2\varphi + 2a_{ij}\cos\varphi\sin\varphi, \\ a_{jj}^{(1)} = a_{ii}\sin^2\varphi + a_{jj}\cos^2\varphi - 2a_{ij}\cos\varphi\sin\varphi, \\ a_{ij}^{(1)} = a_{ji}^{(1)} = a_{ij}(\cos^2\varphi - \sin^2\varphi) - (a_{ii} - a_{jj})\cos\varphi\sin\varphi, \\ a_{il}^{(1)} = a_{li}^{(1)} = a_{il}\cos\varphi + a_{jl}\sin\varphi, \quad l \neq i, j, \\ a_{jl}^{(1)} = a_{lj}^{(1)} = -a_{il}\sin\varphi + a_{jl}\cos\varphi, \quad l \neq i, j, \\ a_{lm}^{(1)} = a_{ml}^{(1)} = a_{ml}, \quad m, l \neq i, j. \end{cases} \tag{7.21}$$

不难看出, $\boldsymbol{A}$ 经过 $\boldsymbol{G}_{ij}$ 的正交相似变换后, $\boldsymbol{A}_1$ 的元素和 $\boldsymbol{A}$ 的元素相比, 只有第 i 行, 第 j 行和第 i 列, 第 j 列元素发生了变化, 而其他元素和 $\boldsymbol{A}$ 是相同的.

由式 (7.21) 的最后一个等式可知, 若 $a_{ij} \neq 0$, 则可适当选取 φ

的值, 使得 $a_{ij}^{(1)} = a_{ji}^{(1)} = 0$. 事实上, 令

$$a_{ij}(\cos^2\varphi - \sin^2\varphi) - (a_{ii} - a_{jj})\cos\varphi\sin\varphi = 0,$$

解得

$$\cot 2\varphi = \frac{a_{ii} - a_{jj}}{2a_{ij}} = \frac{1 - \tan^2\varphi}{2\tan\varphi}, \quad -\frac{\pi}{4} < \varphi \leqslant \frac{\pi}{4}. \tag{7.22}$$

在 Jacobi 方法中, 总是按上式选取 φ. 在实际计算时, 为避免使用三角函数, 可令

$$t = \tan\varphi, \quad c = \cos\varphi, \quad s = \sin\varphi, \quad d = \frac{a_{ii} - a_{jj}}{2a_{ij}}. \tag{7.23}$$

由式 (7.22), 得

$$t^2 + 2dt - 1 = 0. \tag{7.24}$$

式 (7.24) 有两个根, 取其绝对值最小者为 t, 即

$$t = \begin{cases} -d + \sqrt{d^2+1}, & d \geqslant 0, \\ -d - \sqrt{d^2+1}, & d < 0. \end{cases} \tag{7.25}$$

若记

$$c = \cos\varphi = \frac{1}{\sqrt{1+t^2}}, \quad s = \sin\varphi = \frac{t}{\sqrt{1+t^2}} = ct, \tag{7.26}$$

这时, 式 (7.21) 可写为

$$\begin{cases} a_{ii}^{(1)} = a_{ii}c^2 + a_{jj}s^2 + 2csa_{ij}, \\ a_{jj}^{(1)} = a_{ii}s^2 + a_{jj}c^2 - 2csa_{ij}, \quad a_{ij}^{(1)} = a_{ji}^{(1)} = 0, \\ a_{il}^{(1)} = a_{li}^{(1)} = ca_{il} + sa_{jl}, \ a_{jl}^{(1)} = a_{lj}^{(1)} = -sa_{il} + ca_{jl}, \quad l \neq i, j; \\ a_{lm}^{(1)} = a_{ml}^{(1)} = a_{ml}, \quad m, l \neq i, j. \end{cases} \tag{7.27}$$

利用等式 $a_{ij}(c^2-s^2)-(a_{ii}-a_{jj})cs=0$, 可以验证

$$\left(a_{ii}^{(1)}\right)^2+\left(a_{jj}^{(1)}\right)^2=a_{ii}^2+a_{jj}^2+2a_{ij}^2, \tag{7.28}$$

即 $\boldsymbol{A}$ 经过一次这种正交相似变换后, 所得到的矩阵 $\boldsymbol{A}_1$ 的对角元素平方和增加了 $2a_{ij}^2$.

§7.3.2 Jacobi 方法及其收敛性

选择 $\boldsymbol{A}_0=\boldsymbol{A}$ 中一对非零的非对角元素 a_{ij}, a_{ji}, 使用平面旋转矩阵 $\boldsymbol{G}_{ij}$ 作正交相似变换得 $\boldsymbol{A}_1$, 可使 $\boldsymbol{A}_1$ 的这对非对角元素 $a_{ij}^{(1)}=a_{ji}^{(1)}=0$; 再选择 $\boldsymbol{A}_1$ 中一对非零的非对角元素作上述旋转正交相似变换得 $\boldsymbol{A}_2$, 可使 $\boldsymbol{A}_2$ 的这对非对角元素为 0. 如此不断地作旋转正交相似变换, 可产生一个矩阵序列 $\boldsymbol{A}=\boldsymbol{A}_0, \boldsymbol{A}_1, \cdots, \boldsymbol{A}_k, \cdots$. 虽然 $\boldsymbol{A}$ 至多只有 $n(n-1)/2$ 对非零非对角元素, 但不能期望通过

$n(n-1)/2$ 次旋转正交相似变换使其对角化. 因为每次旋转变换虽然能使一对特定的非对角元素化为 0, 但这次变换可能将前面已经化为 0 了的一对非对角元素变成非 0.

但是, 在 Jacobi 方法中的每一步, 如由 $\boldsymbol{A}_{k-1}$ 变成 $\boldsymbol{A}_k$, 取其绝对值最大的一对非零非对角元素, 即取

$$\left|a_{i_k j_k}^{(k-1)}\right| = \max_{\substack{1\leqslant i,j\leqslant n \\ i\neq j}} \left|a_{ij}^{(k-1)}\right| \tag{7.29}$$

作旋转相似变换, 这时记旋转矩阵 $\boldsymbol{G}_{ij} = \boldsymbol{G}_{i_k j_k}$. 后面将证明, 这样产生的矩阵序列 $\boldsymbol{A}_0, \boldsymbol{A}_1, \cdots, \boldsymbol{A}_k, \cdots$ 趋向于对角矩阵, 即 Jacobi 方法是收敛的.

在实际计算中, 可预先取一个小的控制量 $\varepsilon > 0$, 若成立

$$\left|a_{ij}^{(k)}\right| < \varepsilon, \quad i, j = 1, 2, \cdots, n, \ i \neq j, \tag{7.30}$$

则可视 $\boldsymbol{A}_k$ 为对角矩阵, 从而结束计算. $\boldsymbol{A}_k$ 的对角元素可视为 $\boldsymbol{A}$ 的特征值.

Jacobi 方法也可以求 $\boldsymbol{A}$ 的所有特征向量. 事实上, 由

$$\begin{aligned}\boldsymbol{A}_k &= \boldsymbol{G}_k\boldsymbol{A}_{k-1}\boldsymbol{G}_k^{\mathrm{T}} = \boldsymbol{G}_k\boldsymbol{G}_{k-1}\boldsymbol{A}_{k-2}\boldsymbol{G}_{k-1}^{\mathrm{T}}\boldsymbol{G}_k^{\mathrm{T}} = \cdots \\ &= \boldsymbol{G}_k\boldsymbol{G}_{k-1}\cdots\boldsymbol{G}_1\boldsymbol{A}\boldsymbol{G}_1^{\mathrm{T}}\cdots\boldsymbol{G}_{k-1}^{\mathrm{T}}\boldsymbol{G}_k^{\mathrm{T}},\end{aligned}$$

若记

$$\boldsymbol{Q}_k = \boldsymbol{G}_1^{\mathrm{T}}\cdots\boldsymbol{G}_{k-1}^{\mathrm{T}}\boldsymbol{G}_k^{\mathrm{T}}, \tag{7.31}$$

则

$$\boldsymbol{A}_k = \boldsymbol{Q}_k^{\mathrm{T}}\boldsymbol{A}\boldsymbol{Q}_k. \tag{7.32}$$

式中: $\boldsymbol{Q}_k$ 为正交矩阵.

若 $\boldsymbol{A}_k$ 可视为对角矩阵, 其对角元即为 $\boldsymbol{A}$ 的特征值, 其第 i 个对角元 $a_{ii}^{(k)}$ 对应的特征向量就是 $\boldsymbol{Q}_k$ 的第 i 列元素构成的向量. $\boldsymbol{Q}_k$




Back
Close

的计算可与 $\boldsymbol{A}$ 的旋转相似变换同步进行. 若令 $\boldsymbol{Q}_0 = \boldsymbol{I}$, 则

$$\boldsymbol{Q}_k = \boldsymbol{Q}_{k-1}\boldsymbol{G}_k^{\mathrm{T}}. \tag{7.33}$$

若 $\boldsymbol{G}_k = \boldsymbol{G}_{ij}$, 得 $\boldsymbol{Q}_k$ 的计算公式如下:

$$\begin{cases} q_{li}^{(k)} = q_{li}^{(k-1)}c + q_{lj}^{(k-1)}s, \quad l = 1, 2, \cdots, n, \\ q_{lj}^{(k)} = -q_{li}^{(k-1)}s + q_{lj}^{(k-1)}c, \quad l = 1, 2, \cdots, n, \\ q_{km}^{(k)} = q_{km}^{(k-1)}, \quad k, m \neq i, j. \end{cases} \tag{7.34}$$

也就是说, 除了第 i, j 列元素发生变化外, 其他元素不变. 若不需要计算特征向量, 则可省略此步.

根据上述讨论, 可得 Jacobi 方法的计算步骤如下.

算法 7.3 (Jacobi 方法)

步 1, 输入矩阵 $\boldsymbol{A}$, $\boldsymbol{Q} = \boldsymbol{I}$, 初始向量 $\boldsymbol{x}$, 误差限 ε, 最大迭代次数 N. 置 $k := 1$.

步 2, 在矩阵中找绝对值最大的非对角元

$$\mu = \left|a_{i_r j_r}\right| = \max_{\substack{1\leqslant i,j\leqslant n \\ i\neq j}} |a_{ij}|,$$

置 $i := i_r$, $j := j_r$.

步 3, 按式 (7.23) ~ 式 (7.27) 计算 d, t, c, s 的值和矩阵 $\boldsymbol{A}_1$ 的元素 $a_{lm}^{(1)}$, $l, m = 1, 2, \cdots, n$.

步 4, 更新 $\boldsymbol{Q}$ 的元素:

$$\begin{cases} q_{li} := q_{li}c + q_{lj}s, \\ q_{lj} := -q_{li}s + q_{lj}c, \end{cases} \qquad l = 1, 2, \cdots, n.$$

步 5, 若 $\mu < \varepsilon$, 输出 $\boldsymbol{A}_1$ 的对角元和 $\boldsymbol{Q}$ 的列向量, 停算; 否则, 转步 6.

步 6, 若 $k < N$, 置 $k := k + 1$, 转步 2; 否则输出计算失败信息, 停算.

根据算法 7.3, 编制 Jacobi 方法的 MATLAB 程序如下:

```
%Jacobi方法程序-Jacobi_eig.m
function [lambda,Q]=Jacobi_eig(A,tol)
%用Jacobi方法求实对称矩阵A的全部特征值和特征向量
%输入:A为n阶对称方阵,tol为容许误差
%输出:lambda为向量,其分量为A的特征值,
%      Q为矩阵,其元素为矩阵A的n个特征向量
if nargin<2, tol=1e-6; end
[n]=size(A,1); Q=eye(n);
%计算A的非对角元绝对值最大元素所在的行p和列q
[w1,p]=max(abs(A-diag(diag(A))));
[w2,q]=max(w1); p=p(q);
while(1)
```

```
d=(A(p,p)-A(q,q))/(2*A(p,q));
if(d>=0)
   t=-d+sqrt(d^2+1);
else
   t=-d-sqrt(d^2+1);
end
c=1/sqrt(t^2+1); s=c*t; G=[c s; -s c];
A([p q],:)=G*A([p q],:);
A(:,[p q])=A(:,[p q])*G';
Q(:,[p q])=Q(:,[p q])*G';
[w1,p]=max(abs(A-diag(diag(A))));
[w2,q]=max(w1); p=p(q);
if (abs(A(p,q))<tol*sqrt(sum(diag(A).^2)/n))
```

```
        break;
    end
end
lambda=sort(diag(A));
```

例 7.4 利用 Jacobi 方法程序, 求例 7.1 中的矩阵 $\boldsymbol{A}$ 的全部特征值和对应的特征向量.

解 在 MATLAB 命令窗口执行程序 ex74.m 得到矩阵 $\boldsymbol{A}$ 的全部特征值 $\hat{\boldsymbol{\lambda}}$. 此外, 利用 MATLAB 自带的函数 eig 求得其特征值为 $\boldsymbol{\lambda}$, 计算其误差 $\|\hat{\boldsymbol{\lambda}} - \boldsymbol{\lambda}\|_2 = 3.1756 \times 10^{-9}$. 特征值的分布如图 7.1 所示.

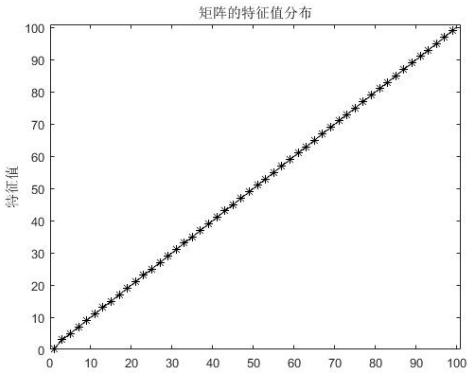

图 7.1 矩阵 $\boldsymbol{A}$ 的特征值分布

下面考虑算法 7.3 (Jacobi 方法) 的收敛性. 记实对称矩阵 $\boldsymbol{A}$ 的非对角元素的平方和为

$$S(\boldsymbol{A}) = \sum_{\substack{l,m=1 \\ l \neq m}}^{n} a_{lm}^2. \tag{7.35}$$

设 $\boldsymbol{A}_{k+1} = \boldsymbol{G}_{ij}\boldsymbol{A}_k\boldsymbol{G}_{ij}^{\mathrm{T}}$, 则由式 (7.27) 不难验证

$$S(\boldsymbol{A}_{k+1}) = S(\boldsymbol{A}_k) - 2\big[a_{ij}^{(k)}\big]^2. \tag{7.36}$$

即经过这种正交相似变换后, $\boldsymbol{A}_{k+1}$ 的非对角元素平方和减少了 $2[a_{ij}^{(k)}]^2$. 同时, 由式 (7.28), 有

$$\big[a_{ii}^{(k+1)}\big]^2 + \big[a_{jj}^{(k+1)}\big]^2 = \big[a_{ii}^{(k)}\big]^2 + \big[a_{jj}^{(k)}\big]^2 + 2\big[a_{ij}^{(k)}\big]^2, \tag{7.37}$$

即对角元素的平方和增加了 $2[a_{ij}^{(k)}]^2$. 若在 Jacobi 方法中, 每次旋转正交相似变换使 $\boldsymbol{A}_k$ 的绝对值最大的非对角元素化为 0, 则成立以下定理.

Back

Close

定理 7.10 记实对称矩阵 $\boldsymbol{A}=\boldsymbol{A}_0$, 若在 Jacobi 方法中, 每次旋转正交相似变换使 $\boldsymbol{A}_k$ 的绝对值最大的非对角元素化为 0, 则得到的矩阵序列 $\{\boldsymbol{A}_k\}$ 趋向于对角矩阵.

证明 设 $\boldsymbol{A}_k$ 的绝对值最大的非对角元素为 $a_{ij}^{(k)}$, 故有

$$\left[a_{ij}^{(k)}\right]^2 \geqslant \frac{1}{n(n-1)} S(\boldsymbol{A}_k).$$

用旋转正交相似变换将其化为 0, 得 $\boldsymbol{A}_{k+1}$, 此时

$$\begin{aligned} S(\boldsymbol{A}_{k+1}) &= S(\boldsymbol{A}_k) - 2\left[a_{ij}^{(k)}\right]^2 \leqslant S(\boldsymbol{A}_k) - \frac{2}{n(n-1)} S(\boldsymbol{A}_k) \\ &= \left[1 - \frac{2}{n(n-1)}\right] S(\boldsymbol{A}_k) \leqslant \left[1 - \frac{2}{n(n-1)}\right]^{k+1} S(\boldsymbol{A}). \end{aligned}$$

由于

$$1 - \frac{2}{n(n-1)} < 1,$$

所以

$$\lim_{k\to\infty} S(\boldsymbol{A}_{k+1}) = 0,$$

即 $\boldsymbol{A}_{k+1}$ 趋向于对角矩阵, 故 Jacobi 方法是收敛的. 证毕. □

§7.4 QR 方法

QR 方法用于求一般矩阵的全部特征值, 是目前最有效的方法之一. 本节就实矩阵的情形进行介绍.

众所周知, 对于任何实对称矩阵 $\boldsymbol{A} \in \mathbb{R}^{n\times n}$, 存在正交矩阵 $\boldsymbol{Q}$ 使得

$$\boldsymbol{Q}^{\mathrm{T}}\boldsymbol{A}\boldsymbol{Q} = \mathrm{diag}(\lambda_1, \lambda_2, \cdots, \lambda_n),$$

式中: $\lambda_1, \lambda_2, \cdots, \lambda_n$ 为 $\boldsymbol{A}$ 的全部特征值; $\boldsymbol{Q}$ 的列向量为对应的特征向量.

而对于一般的 $\boldsymbol{A} \in \mathbb{R}^{n\times n}$, 有下面的实 Schur 分解定理.

定理 7.11 对于任何实矩阵 $\boldsymbol{A} \in \mathbb{R}^{n\times n}$, 存在正交矩阵 $\boldsymbol{Q}$ 使

得

$$\boldsymbol{Q}^{\mathrm{T}}\boldsymbol{A}\boldsymbol{Q}=\begin{bmatrix}\boldsymbol{R}_{11} & \boldsymbol{R}_{12} & \cdots & \boldsymbol{R}_{1m}\\ & \boldsymbol{R}_{22} & \cdots & \boldsymbol{R}_{2m}\\ & & \ddots & \vdots\\ & & & \boldsymbol{R}_{mm}\end{bmatrix}, \tag{7.38}$$

式中: 对角块 $\boldsymbol{R}_{ii}\,(i=1,2,\cdots,m)$ 为 1×1 或 2×2 的子矩阵, 1 阶子矩阵的元素是 $\boldsymbol{A}$ 的实特征值, 2 阶子矩阵的两个特征值是 $\boldsymbol{A}$ 的一对复共轭特征值.

式 (7.38) 通常称为矩阵 $\boldsymbol{A}$ 的实 Schur 分解, 而右边的分块上三角矩阵称为 $\boldsymbol{A}$ 的实 Schur 标准形. 显然, 只要求得一个实矩阵的实 Schur 标准形, 就很容易求得它的全部特征值.

因此, 通常希望构造一种迭代 (相似变换), 希望它能逼近矩阵 $\boldsymbol{A}$ 的实 Schur 标准形. 例如, 对于给定的矩阵 $\boldsymbol{A}\in\mathbb{R}^{n\times n}$, 令

$\boldsymbol{A}_1 := \boldsymbol{A}$, 构造迭代:

$$\begin{cases} \boldsymbol{A}_k = \boldsymbol{Q}_k \boldsymbol{R}_k, \\ \boldsymbol{A}_{k+1} = \boldsymbol{R}_k \boldsymbol{Q}_k, \end{cases} \quad k = 1, 2, \cdots, \tag{7.39}$$

式中: $\boldsymbol{Q}_k$ 为正交矩阵; $\boldsymbol{R}_k$ 为上三角矩阵.

可以证明, 在一定条件下, 由式 (7.39) 产生的矩阵序列 $\{\boldsymbol{A}_k\}$ 将“逼近”于 $\boldsymbol{A}$ 的实 Schur 标准形.

然而, 式 (7.39) 作为一种实用的迭代法是没有竞争力的. 一是每步迭代的运算量太大 (大约是 $O(n^3)$); 二是收敛速度太缓慢 (依赖于特征值的分离程度). 因此, 要想其成为一种高效的迭代方法, 必须设法尽可能地减少每步迭代的运算量, 提高其收敛速度. 一个可行的办法是, 首先把矩阵 $\boldsymbol{A}$ 正交相似变换为上 Hessenberg 形, 然后再用正交相似变换对它进行迭代. 下面就循着这个思路介绍实用的 QR 方法来求实矩阵 $\boldsymbol{A}$ 的全部特征值.




Back
Close

§7.4.1 化一般矩阵为上 Hessenberg 矩阵

在用 QR 方法求矩阵特征值时, Householder 矩阵有两个作用: 一是对 $\boldsymbol{A}$ 作正交相似变换, 把 $\boldsymbol{A}$ 化为上 Hessenberg 矩阵; 二是对矩阵作正交三角分解.

首先讨论把 $\boldsymbol{A}$ 化为上 Hessenberg 矩阵. 设 $\boldsymbol{A}_1 = \boldsymbol{A} = (a_{ij}^{(1)})$ 是 n 阶实方阵, 取 $\boldsymbol{x} = (0, a_{21}^{(1)}, \cdots, a_{n1}^{(1)})^{\mathrm{T}}$, 记 $a_1 = \mathrm{sgn}(x_2)\|\boldsymbol{x}\|_2$, 则由定理 2.2 和定理 2.3 构造 Householder 矩阵

$$\boldsymbol{H}_1 = \begin{bmatrix} 1 & 0 & \cdots & 0 \\ 0 & * & \cdots & * \\ \vdots & \vdots & \ddots & \vdots \\ 0 & * & \cdots & * \end{bmatrix},$$

使得

$$\boldsymbol{H}_1\boldsymbol{x} = a_1\boldsymbol{e}_2.$$

所以 $\boldsymbol{H}_1\boldsymbol{A}_1$ 的第 1 列为

$$\boldsymbol{H}_1\begin{bmatrix} a_{11}^{(1)} \\ a_{21}^{(1)} \\ \vdots \\ a_{n1}^{(1)} \end{bmatrix} = \boldsymbol{H}_1\boldsymbol{x} + \boldsymbol{H}_1\begin{bmatrix} a_{11}^{(1)} \\ 0 \\ \vdots \\ 0 \end{bmatrix} = a_1\boldsymbol{e}_2 + \begin{bmatrix} a_{11}^{(1)} \\ 0 \\ \vdots \\ 0 \end{bmatrix} = \begin{bmatrix} a_{11}^{(1)} \\ a_1 \\ \vdots \\ 0 \end{bmatrix}.$$

因为用 $\boldsymbol{H}_1$ 右乘一个矩阵不改变该矩阵的第 1 列, 于是

$$\boldsymbol{A}_2 = \boldsymbol{H}_1\boldsymbol{A}_1\boldsymbol{H}_1 = \begin{bmatrix} a_{11}^{(1)} & a_{12}^{(2)} & \cdots & a_{1n}^{(2)} \\ a_1 & a_{22}^{(2)} & \cdots & a_{2n}^{(2)} \\ 0 & a_{32}^{(2)} & \cdots & a_{3n}^{(2)} \\ \vdots & \vdots & \ddots & \vdots \\ 0 & a_{n2}^{(2)} & \cdots & a_{nn}^{(2)} \end{bmatrix}.$$

再取 $\boldsymbol{x} = (0, 0, a_{32}^{(2)}, \cdots, a_{n2}^{(2)})^{\mathrm{T}}$, 记 $a_2 = \mathrm{sgn}(x_3)\|\boldsymbol{x}\|_2$, 构造 $\boldsymbol{H}_2$ 为

$$\boldsymbol{H}_2 = \begin{bmatrix} 1 & 0 & 0 & \cdots & 0 \\ 0 & 1 & 0 & \cdots & 0 \\ 0 & 0 & * & \cdots & * \\ \vdots & \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & * & \cdots & * \end{bmatrix},$$




Back
Close

使得

$$\boldsymbol{H}_2\boldsymbol{x} = a_2\boldsymbol{e}_3.$$

所以 $\boldsymbol{H}_2\boldsymbol{A}_2$ 的第 1 列与 $\boldsymbol{A}_2$ 的第 1 列相同, 而 $\boldsymbol{H}_2\boldsymbol{A}_2$ 的第 2 列变为

$$\boldsymbol{H}_2\boldsymbol{x} + \boldsymbol{H}_2 \begin{bmatrix} a_{12}^{(2)} \\ a_{22}^{(2)} \\ 0 \\ \vdots \\ 0 \end{bmatrix} = a_2\boldsymbol{e}_3 + \begin{bmatrix} a_{12}^{(2)} \\ a_{22}^{(2)} \\ 0 \\ \vdots \\ 0 \end{bmatrix} = \begin{bmatrix} a_{12}^{(2)} \\ a_{22}^{(2)} \\ a_2 \\ \vdots \\ 0 \end{bmatrix}.$$

而用 $\boldsymbol{H}_2$ 右乘一个矩阵不改变该矩阵的第 1 列和第 2 列, 于是

$$\boldsymbol{A}_3 = \boldsymbol{H}_2\boldsymbol{A}_2\boldsymbol{H}_2 = \begin{bmatrix} * & * & * & \cdots & * \\ a_1 & * & * & \cdots & * \\ 0 & a_2 & * & \cdots & * \\ 0 & 0 & * & \cdots & * \\ \vdots & \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & * & \cdots & * \end{bmatrix}.$$

Back

Close

这样下去, 经过 $n-2$ 次变换后, $\boldsymbol{A}_1$ 就化为上 Hessenberg 矩阵 $\boldsymbol{A}_{n-1}$, 即

$$\boldsymbol{A}_{n-1} = \boldsymbol{H}_{n-2}\cdots\boldsymbol{H}_2\boldsymbol{H}_1\boldsymbol{A}_1\boldsymbol{H}_1\boldsymbol{H}_2\cdots\boldsymbol{H}_{n-2}$$

$$= \begin{bmatrix} * & * & * & * & \cdots & * \\ a_1 & * & * & * & \cdots & * \\ & a_2 & * & * & \cdots & * \\ & & a_3 & * & \cdots & * \\ & & & \ddots & \ddots & \vdots \\ & & & & a_{n-1} & * \end{bmatrix}.$$

如果 $\boldsymbol{A}_1$ 是对称矩阵, 则 $\boldsymbol{A}_{n-1}$ 仍是对称矩阵, 此时 $\boldsymbol{A}_{n-1}$ 将是对称三对角矩阵:

$$\boldsymbol{A}_{n-1} = \begin{bmatrix} * & a_1 & & & \\ a_1 & * & a_2 & & \\ & a_2 & * & a_3 & \\ & & \ddots & \ddots & a_{n-1} \\ & & & a_{n-1} & * \end{bmatrix}.$$

Back

Close

以上利用 Householder 变换约化 $\boldsymbol{A}$ 为上 Hessenberg 形的方法, 可总结为如下实用算法.

算法 7.4 (上 Hessenberg 化)

步 1, 输入 $\boldsymbol{A} = (a_{ij})$, $k := 1$.

步 2, 计算 $n-k$ 阶 Householder 矩阵 $\widetilde{\boldsymbol{H}}_k$, 使

$$\widetilde{\boldsymbol{H}}_k \begin{bmatrix} a_{k+1,k} \\ a_{k+2,k} \\ \vdots \\ a_{n,k} \end{bmatrix} = \begin{bmatrix} * \\ 0 \\ \vdots \\ 0 \end{bmatrix},$$

置 $\boldsymbol{A} := \boldsymbol{H}_k \boldsymbol{A} \boldsymbol{H}_k$, 其中 $\boldsymbol{H}_k = \mathrm{diag}(\boldsymbol{I}_k, \widetilde{\boldsymbol{H}}_k)$.

步 3, 若 $k < n-2$, 则 $k := k+1$, 转步 2; 否则输出有关信息, 停算.

算法 7.4 计算出 $\boldsymbol{A}$ 的上 Hessenberg 形就存在 $\boldsymbol{A}$ 所对应的存储单元内, 所需运算量是 $5n^3/3$. 如果需要累计 $\boldsymbol{Q} = \boldsymbol{H}_1\boldsymbol{H}_2\cdots\boldsymbol{H}_{n-2}$, 则还需增加 $2n^3/3$ 的运算量.

将矩阵 $\boldsymbol{A}$ 化为上 Hessenberg 矩阵 (算法 7.4) 的 MATLAB 程序如下:

```
function [A,Q]=mhessen(A)
%用Householder变换化n阶矩阵A为上Hessenberg矩阵
%调用函数:mhouse.m
n=size(A,1); Q=eye(n);
for k=1:(n-2)
    x=A(k+1:n,k); [v,beta]=mhouse(x);
    H=(eye(length(v))-beta*v*v');
    A(k+1:n,1:n)=H*A(k+1:n,1:n);
```

```
        A(1:n,k+1:n)=A(1:n,k+1:n)*H;
        Q=Q*blkdiag(eye(k),H);
    end
```

例 7.5 利用 MATLAB 程序 mhessen.m, 将下列矩阵化为上 Hessenberg 矩阵:

$$\boldsymbol{A}=\begin{bmatrix} -1 & 2 & 3 & 5 \\ 2 & -3 & 8 & 1 \\ 3 & 8 & -2 & 7 \\ 5 & 1 & 7 & 6 \end{bmatrix}.$$

解 在 MATLAB 命令窗口输入:

```
>> A=[-1 2 3 5; 2 -3 8 1; 3 8 -2 7; 5 1 7 6];
>> [A,Q]=mhessen(A)
```

```
A =
   -1.0000   -6.1644    0.0000    0.0000
   -6.1644   11.7368    1.8380    0.0000
    0.0000    1.8380   -6.5929    5.9938
    0.0000    0.0000    5.9938   -4.1439
Q =
    1.0000         0         0         0
         0    0.3244   -0.0418   -0.9450
         0    0.4867    0.8640    0.1289
         0    0.8111   -0.5017    0.3007
```

一般来说, 上 Hessenberg 分解是不唯一的, 但可以证明下面的结果.

定理 7.12 设 $\boldsymbol{A} \in \mathbb{R}^{n\times n}$ 有如下两个上 Hessenberg 分解:

$$\boldsymbol{U}^{\mathrm{T}}\boldsymbol{A}\boldsymbol{U} = \boldsymbol{H}, \quad \boldsymbol{V}^{\mathrm{T}}\boldsymbol{A}\boldsymbol{V} = \boldsymbol{G}, \tag{7.40}$$

式中: $\boldsymbol{U} = [\boldsymbol{u}_1, \boldsymbol{u}_2, \cdots, \boldsymbol{u}_n]$ 和 $\boldsymbol{V} = [\boldsymbol{v}_1, \boldsymbol{v}_2, \cdots, \boldsymbol{v}_n]$ 为 n 阶正交矩阵; $\boldsymbol{H} = (h_{ij})$ 和 $\boldsymbol{G} = (g_{ij})$ 为上 Hessenberg 矩阵. 若 $\boldsymbol{u}_1 = \boldsymbol{v}_1$, 且 $\boldsymbol{H}$ 的次对角元 $h_{i+1,i} \neq 0\,(i = 1, 2, \cdots, n)$, 则存在对角元均为 1 或 -1 的对角矩阵 $\boldsymbol{D}$, 使得

$$\boldsymbol{U} = \boldsymbol{V}\boldsymbol{D}, \quad \boldsymbol{H} = \boldsymbol{D}\boldsymbol{G}\boldsymbol{D}, \tag{7.41}$$

即 $\boldsymbol{u}_i = \pm\boldsymbol{v}_i$, $|h_{ij}| = |g_{ij}|$, $i, j = 1, 2, \cdots, n$.

证明 对矩阵的阶数 n 用归纳法. 对于 $n = 1$ 时结论显然成立. 假设 $n = m$ 时结论成立, 即

$$\boldsymbol{u}_i = \varepsilon_i\boldsymbol{v}_i, \quad i = 1, 2, \cdots, m, \tag{7.42}$$

式中: $\varepsilon_1 = 1$, $\varepsilon_i = 1$ 或 -1, $i = 2, \cdots, m$. 下面证明存在 ε_{m+1} 为 1 或 -1 使得

$$\boldsymbol{u}_{m+1} = \varepsilon_{m+1}\boldsymbol{v}_{m+1}.$$

由式 (7.40), 得

$$\boldsymbol{AU} = \boldsymbol{UH}, \quad \boldsymbol{AV} = \boldsymbol{VG}.$$

分别比较上面两个矩阵等式两边的第 m 列, 可得

$$\boldsymbol{A}\boldsymbol{u}_m = h_{1m}\boldsymbol{u}_1 + \cdots + h_{mm}\boldsymbol{u}_m + h_{m+1,m}\boldsymbol{u}_{m+1}, \tag{7.43}$$

$$\boldsymbol{A}\boldsymbol{v}_m = g_{1m}\boldsymbol{v}_1 + \cdots + g_{mm}\boldsymbol{v}_m + g_{m+1,m}\boldsymbol{v}_{m+1}. \tag{7.44}$$

分别在式 (7.43) 和式 (7.44) 两边左乘 $\boldsymbol{u}_i^{\mathrm{T}}$ 和 $\boldsymbol{v}_i^{\mathrm{T}}$ $(i = 1, 2, \cdots, m)$, 得

$$h_{im} = \boldsymbol{u}_i^{\mathrm{T}}\boldsymbol{A}\boldsymbol{u}_m, \quad i = 1, 2, \cdots, m, \tag{7.45}$$

$$g_{im} = \boldsymbol{v}_i^{\mathrm{T}}\boldsymbol{A}\boldsymbol{v}_m, \quad i = 1, 2, \cdots, m. \tag{7.46}$$

由式 (7.42)、式 (7.45) 和式 (7.46), 得

$$h_{im} = \varepsilon_i \varepsilon_m g_{im}, \ i = 1, 2, \cdots, m. \tag{7.47}$$

将式 (7.47) 代入式 (7.43), 并利用式 (7.42) 和式 (7.44), 得

$$\begin{aligned} h_{m+1,m}\boldsymbol{u}_{m+1} &= \boldsymbol{A}\boldsymbol{u}_m - \varepsilon_1\varepsilon_m g_{1m}\boldsymbol{u}_1 - \cdots - \varepsilon_m\varepsilon_m g_{mm}\boldsymbol{u}_m \\ &= \varepsilon_m(\boldsymbol{A}\boldsymbol{v}_m - \varepsilon_1^2 g_{1m}\boldsymbol{v}_1 - \cdots - \varepsilon_m^2 g_{mm}\boldsymbol{v}_m) \\ &= \varepsilon_m(\boldsymbol{A}\boldsymbol{v}_m - g_{1m}\boldsymbol{v}_1 - \cdots - g_{mm}\boldsymbol{v}_m) \\ &= \varepsilon_m g_{m+1,m}\boldsymbol{v}_{m+1}. \end{aligned} \tag{7.48}$$

上式两边取范数, 得

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

由于 $h_{m+1,m} \neq 0$, 故式 (7.48) 蕴含着

$$\boldsymbol{u}_{m+1} = \varepsilon_{m+1}\boldsymbol{v}_{m+1},$$

式中: $\varepsilon_{m+1} = 1$ 或 -1. 证毕. □

注 7.3 一个上 Hessenberg 矩阵 $\boldsymbol{H} = (h_{ij})$, 如果其次对角元均不为零, 即 $h_{i+1,i} \neq 0$, $i = 1, 2, \cdots, n-1$, 则它是不可约的. 定理 7.12 表明, 如果 $\boldsymbol{Q}^{\mathrm{T}}\boldsymbol{A}\boldsymbol{Q} = \boldsymbol{H}$ 为不可约的上 Hessenberg 矩阵, 则 $\boldsymbol{Q}$ 和 $\boldsymbol{H}$ 完全由 $\boldsymbol{Q}$ 的第 1 列确定 (这里是在相差一个正负号意义下的唯一).

§7.4.2 上 Hessenberg 矩阵的 QR 分解

对于上 Hessenberg 矩阵

$$\boldsymbol{H} = \begin{bmatrix} h_{11}^{(1)} & h_{12}^{(1)} & h_{13}^{(1)} & \cdots & h_{1n}^{(1)} \\ h_{21}^{(1)} & h_{22}^{(1)} & h_{23}^{(1)} & \cdots & h_{2n}^{(1)} \\ & h_{32}^{(1)} & h_{33}^{(1)} & \cdots & h_{2n}^{(1)} \\ & & \ddots & \ddots & \vdots \\ & & & h_{n,n-1}^{(1)} & h_{nn}^{(1)} \end{bmatrix},$$

通常可以通过 $n-1$ 次 Givens 变换将它化成上三角矩阵, 从而得到 $\boldsymbol{H}$ 的 QR 分解式. 具体步骤是:

(1) 记 $\boldsymbol{H}_1 = \boldsymbol{H}$. 设 $h_{21}^{(1)} \neq 0$ (否则可进行下一步), 取 Givens 矩阵

$$\boldsymbol{G}_{21} = \begin{bmatrix} c_1 & s_1 & & & \\ -s_1 & c_1 & & & \\ & & 1 & & \\ & & & \ddots & \\ & & & & 1 \end{bmatrix},$$

式中:

$$c_1 = \frac{h_{11}^{(1)}}{r_1}, \quad s_1 = \frac{h_{21}^{(1)}}{r_1}, \quad r_1 = \sqrt{(h_{11}^{(1)})^2 + (h_{21}^{(1)})^2}.$$

则

$$\boldsymbol{G}_{21}\boldsymbol{H}_1 = \begin{bmatrix} r_1 & h_{12}^{(2)} & h_{13}^{(2)} & \cdots & h_{1n}^{(2)} \\ 0 & h_{22}^{(2)} & h_{23}^{(2)} & \cdots & h_{2n}^{(2)} \\ & h_{32}^{(2)} & h_{33}^{(2)} & \cdots & h_{3n}^{(2)} \\ & & \ddots & \ddots & \vdots \\ & & & h_{n,n-1}^{(2)} & h_{nn}^{(2)} \end{bmatrix} := \boldsymbol{H}_2.$$

(2) 设 $h_{32}^{(1)} \neq 0$ (否则可进行下一步), 再取 Givens 矩阵

$$\boldsymbol{G}_{32} = \begin{bmatrix} 1 & & & & & \\ & c_2 & s_2 & & & \\ & -s_2 & c_2 & & & \\ & & & 1 & & \\ & & & & \ddots & \\ & & & & & 1 \end{bmatrix},$$

式中:

$$c_2 = \frac{h_{22}^{(2)}}{r_2}, \quad s_2 = \frac{h_{32}^{(2)}}{r_2}, \quad r_2 = \sqrt{(h_{22}^{(2)})^2 + (h_{32}^{(2)})^2}.$$

则

$$\boldsymbol{G}_{32}\boldsymbol{H}_2 = \begin{bmatrix} r_1 & h_{12}^{(3)} & h_{13}^{(3)} & \cdots & h_{1,n-1}^{(3)} & h_{1n}^{(3)} \\ 0 & r_2 & h_{23}^{(3)} & \cdots & h_{2,n-1}^{(3)} & h_{2n}^{(3)} \\ & 0 & h_{33}^{(3)} & \cdots & h_{3,n-1}^{(3)} & h_{3n}^{(3)} \\ & & h_{43}^{(3)} & \cdots & h_{4,n-1}^{(3)} & h_{4n}^{(3)} \\ & & & \ddots & \vdots & \vdots \\ & & & & h_{n,n-1}^{(3)} & h_{nn}^{(3)} \end{bmatrix} := \boldsymbol{H}_3.$$




Back
Close

(3) 假设上述过程已经进行了 $k-1$ 步, 有

$$\boldsymbol{H}_k = \boldsymbol{G}_{k,k-1}\boldsymbol{H}_{k-1} = \begin{bmatrix} r_1 & \cdots & h_{1,k-1}^{(k)} & h_{1k}^{(k)} & \cdots & h_{1,n-1}^{(k)} & h_{1n}^{(k)} \\ & \ddots & & & & & \\ & & r_{k-1} & h_{k-1,k}^{(k)} & \cdots & h_{k-1,n-1}^{(k)} & h_{k-1,n}^{(k)} \\ & & & h_{kk}^{(k)} & \cdots & h_{k,n-1}^{(k)} & h_{kn}^{(k)} \\ & & & h_{k+1,k}^{(k)} & \cdots & h_{k+1,n-1}^{(k)} & h_{k+1,n}^{(k)} \\ & & & & \ddots & \vdots & \vdots \\ & & & & & h_{n,n-1}^{(k)} & h_{nn}^{(k)} \end{bmatrix}.$$

设 $h_{k+1,k}^{(k)} \neq 0$, 取 Givens 矩阵

$$\boldsymbol{G}_{k+1,k} = \begin{bmatrix} 1 & & & & & & \\ & \ddots & & & & & \\ & & 1 & & & & \\ & & & c_k & s_k & & \\ & & & -s_k & c_k & & \\ & & & & & 1 & \\ & & & & & & \ddots \\ & & & & & & 1 \end{bmatrix},$$

式中:

$$c_k = \frac{h_{kk}^{(k)}}{r_k}, \quad s_k = \frac{h_{k+1,k}^{(k)}}{r_k}, \quad r_k = \sqrt{(h_{kk}^{(k)})^2 + (h_{k+1,k}^{(k)})^2}.$$

于是

$$\boldsymbol{G}_{k+1,k}\boldsymbol{H}_k = \begin{bmatrix} r_1 & \cdots & h_{1,k}^{(k+1)} & h_{1,k+1}^{(k+1)} & \cdots & h_{1,n-1}^{(k+1)} & h_{1n}^{(k+1)} \\ & \ddots & & & & & \\ & & r_k & h_{k,k+1}^{(k+1)} & \cdots & h_{k,n-1}^{(k+1)} & h_{kn}^{(k+1)} \\ & & & h_{k+1,k+1}^{(k+1)} & \cdots & h_{k+1,n-1}^{(k+1)} & h_{k+1,n}^{(k+1)} \\ & & & h_{k+2,k+1}^{(k+1)} & \cdots & h_{k+2,n-1}^{(k+1)} & h_{k+2,n}^{(k+1)} \\ & & & & \ddots & \vdots & \vdots \\ & & & & & h_{n,n-1}^{(k+1)} & h_{nn}^{(k+1)} \end{bmatrix} := \boldsymbol{H}_{k+1}.$$

因此, 最多作 $n-1$ 次 Givens 变换, 即得

$$\boldsymbol{G}_{n,n-1}\cdots\boldsymbol{G}_{32}\boldsymbol{G}_{21}\boldsymbol{H}=\begin{bmatrix} r_1 & h_{12}^{(n)} & h_{13}^{(n)} & \cdots & h_{1n}^{(n)} \\ & r_2 & h_{23}^{(n)} & \cdots & h_{2n}^{(n)} \\ & & r_3 & \cdots & h_{3n}^{(n)} \\ & & & \ddots & \vdots \\ & & & & r_n \end{bmatrix}=\boldsymbol{R}.$$

因为 $\boldsymbol{G}_{k,k-1}\,(k=2,\cdots,n)$ 均为正交阵, 故

$$\boldsymbol{H}=\boldsymbol{G}_{21}^{\mathrm{T}}\boldsymbol{G}_{32}^{\mathrm{T}}\cdots\boldsymbol{G}_{n,n-1}^{\mathrm{T}}\boldsymbol{R}=\boldsymbol{Q}\boldsymbol{R},$$

式中: $\boldsymbol{Q}=\boldsymbol{G}_{21}^{\mathrm{T}}\boldsymbol{G}_{32}^{\mathrm{T}}\cdots\boldsymbol{G}_{n,n-1}^{\mathrm{T}}$ 仍为正交阵.

可算出完成这一过程的运算量约为 $4n^2$, 比一般矩阵的 QR 分解的运算量 $O(n^3)$ 少了一个数量级.

值得注意的是, 可以证明 $\widetilde{\boldsymbol{H}}=\boldsymbol{R}\boldsymbol{Q}=\boldsymbol{Q}^{\mathrm{T}}\boldsymbol{H}\boldsymbol{Q}$ 仍为上 Hessenberg 矩阵, 于是可按上述步骤一直迭代下去, 直到 $\boldsymbol{H}$ 正交相似于

上三角矩阵或块上三角矩阵 (对角块为 1×1 或 2×2 矩阵) 为止, 从而求得矩阵 $\boldsymbol{H}$ 的全部特征值和相应的特征向量.

上 Hessenberg 矩阵 QR 分解的 MATLAB 程序如下:

```
function A=hessen_qrtran(A,m)
%本程序输入n阶上Hessenberg矩阵A, 用Givens变换
%对其左上角m阶主子块进行QR分解,再作相似变换,
%最后输出变换后的上Hessenberg矩阵A.
Q=eye(m);
for i=1:m-1
   xi=A(i,i); xk=A(i+1,i);
   if xk~=0
      d=sqrt(xi^2+xk^2);
      c=xi/d; s=xk/d;
```




```
        G=[c, s; -s, c];
        A(i:i+1,i:m)=G*A(i:i+1,i:m);
        Q(1:m,i:i+1)=Q(1:m,i:i+1)*G';
    end
end
A(1:m,1:m)=A(1:m,1:m)*Q;
```

例 7.6 利用程序将上 Hessenberg 矩阵 $\boldsymbol{A}$ 进行 QR 变换, 其中

$$\boldsymbol{A}=\begin{bmatrix} 2 & 3 & 5 & 7 & 8 \\ 4 & 2 & 3 & 5 & 9 \\ 0 & 8 & 3 & 6 & 2 \\ 0 & 0 & 7 & 1 & 3 \\ 0 & 0 & 0 & 6 & 9 \end{bmatrix}.$$

解 在 MATLAB 命令窗口输入:

```
>> A=[2 3 5 7 8; 4 2 3 5 9; 0 8 3 6 2;...
      0 0 7 1 3; 0 0 0 6 9];
>> A=hessen_qrtran(A,5)
A =
    4.8000    5.3682    7.6664   10.5793   -4.9445
    7.3321    2.7238    4.9535    0.3480   -6.8327
         0    7.2218    1.1745    2.7581   -3.2311
         0         0    6.0298    7.7534   -4.9785
         0         0         0   -1.5283    0.5483
```

§7.4.3 基本 QR 方法

本节介绍求一般方阵全部特征值的 QR 方法. 令 $\boldsymbol{A}_1 = \boldsymbol{A}$, 对 $\boldsymbol{A}_1$ 作 QR 分解:

$$\boldsymbol{A}_1 = \boldsymbol{Q}_1 \boldsymbol{R}_1,$$

然后令 $\boldsymbol{A}_2 = \boldsymbol{R}_1 \boldsymbol{Q}_1$, 再对 $\boldsymbol{A}_2$ 作 QR 分解:

$$\boldsymbol{A}_2 = \boldsymbol{Q}_2 \boldsymbol{R}_2,$$

并令 $\boldsymbol{A}_3 = \boldsymbol{R}_2 \boldsymbol{Q}_2$, 这样下去就得到一个矩阵序列 $\{\boldsymbol{A}_k\}$, 其产生过程可概述如下:

$$\begin{cases} \boldsymbol{A}_1 = \boldsymbol{A}, \\ \boldsymbol{A}_k = \boldsymbol{Q}_k \boldsymbol{R}_k, \qquad k = 1, 2, \cdots. \\ \boldsymbol{A}_{k+1} = \boldsymbol{R}_k \boldsymbol{Q}_k, \end{cases} \tag{7.49}$$

容易证明, $\boldsymbol{A}_{k+1}$ 与 $\boldsymbol{A}_k$ 相似, 故 $\{\boldsymbol{A}_k\}$ 有相同的特征值.




在一定条件下, $\{\boldsymbol{A}_k\}$ 本质上收敛于上三角矩阵 (或分块上三角矩阵). 若它们收敛于上三角矩阵, 则该上三角矩阵的对角元就是原矩阵 $\boldsymbol{A}$ 的全部特征值; 若收敛于分块上三角矩阵, 则这些分块矩阵的特征值也就是 $\boldsymbol{A}$ 的特征值.

由于当 $\boldsymbol{A}$ 为一般的实矩阵时, $\{\boldsymbol{A}_k\}$ 的收敛速度较慢, 故在 QR 方法的实际应用中, 通常先将 $\boldsymbol{A}$ 化为相似的上 Hessenberg 矩阵, 再求特征值以加快收敛速度. 它的计算过程如下.

算法 7.5 (基本 QR 方法)

步 1, 输入上 Hessenberg 矩阵 $\boldsymbol{A} \in \mathbb{R}^{n\times n}$.

步 2, 记 $\boldsymbol{A}_1 := \boldsymbol{A}$. 对于 $k = 1, 2, \cdots$, 有

(1) $\boldsymbol{A}_k = \boldsymbol{Q}_k \boldsymbol{R}_k$ (QR 分解).

(2) $\boldsymbol{A}_{k+1} = \boldsymbol{Q}_k^{\mathrm{T}} \boldsymbol{A}_k \boldsymbol{Q}_k = \boldsymbol{R}_k \boldsymbol{Q}_k$ (正交相似变换).

基本 QR 方法的 MATLAB 程序如下:

```
function [iter,D]=qr_eig(A,tol,N)
%用基本QR算法求n阶实方阵A的全部特征值
%输入:A为实对称矩阵,tol为控制精度,N为最大迭代次数
%输出:iter为迭代次数,D为A的全部特征值
%调用函数:mhessen.m,hessen_qrtran.m,eig-仅用于1,2矩阵
if nargin<3, N=500; end
if nargin<2, tol=1e-5; end
n=size(A,1); D=zeros(n,1);
i=n; m=n; iter=0;  %初始化
A=mhessen(A);  %化矩阵A为Hessenberg矩阵
while (iter<=N)  %用基本QR算法进行迭代
    iter=iter+1;
```

```
 if m<=2
     la=eig(A(1:m,1:m)); D(1:m)=la';
     break;
 end
%对上Hessenberg矩阵作QR分解并作正交相似变换
A=hessen_qrtarn(A,m);
%下面的程序段判断是否终止
for k=m-1:-1:1
   if abs(A(k+1,k))<tol
       if m-k<=2
          la=eig(A(k+1:m,k+1:m));
          j=i-m+k+1; D(j:i)=la';
          i=j-1; m=k; break;
```

```
            end
        end
    end
end
```

例 7.7 利用程序 qr_eig.m, 求下列矩阵的全部特征值:

$$\boldsymbol{A}=\begin{bmatrix} 3 & 2 & 3 & 4 & 5 & 6 & 7 \\ 11 & 1 & 2 & 3 & 4 & 5 & 6 \\ 2 & 8 & 9 & 1 & 2 & 3 & 4 \\ -4 & 2 & 9 & 11 & 13 & 15 & 8 \\ -1 & -2 & -3 & -1 & -1 & -1 & -1 \\ 3 & 2 & 3 & 4 & 13 & 15 & 8 \\ -2 & -2 & -3 & -4 & -5 & -3 & -3 \end{bmatrix}.$$

解 在 MATLAB 命令窗口输入:

```
>> A=[3 2 3 4 5 6 7;11 1 2 3 4 5 6;2 8 9 1 2 3 4; ...
      -4 2 9 11 13 15 8; -1 -2 -3 -1 -1 -1 -1; ...
```

```
      3 2 3 4 13 15 8; -2 -2 -3 -4 -5 -3 -3];
>> [iter,D]=qr_eig(A)
iter =
      622
D =
   18.4123 + 0.0000i
   11.1805 + 0.0000i
    1.7099 - 4.2522i
    1.7099 + 4.2522i
    4.4983 + 0.0000i
   -2.2327 + 0.0000i
   -0.2783 + 0.0000i
```

下面分析基本 QR 方法的收敛性.




Back
Close

定义 7.4 若由 QR 方法产生的序列 $\{\boldsymbol{A}_k\}$ 当 $k \to \infty$ 收敛于分块上三角矩阵 (对角块为一阶或二阶子块), 则称 QR 方法是收敛的. 若序列 $\{\boldsymbol{A}_k\}$ 当 $k \to \infty$ 时, 其对角元均收敛且严格下三角部分元素收敛于 0, 则称 $\{\boldsymbol{A}_k\}$ 基本收敛到上三角阵.

值得注意的是, 基本收敛的概念并未指出 $\{\boldsymbol{A}_k\}$ 严格上三角部分元素是否收敛. 但对求矩阵 $\boldsymbol{A}$ 的特征值而言, 基本收敛足够了.

算法 7.5 具有下列性质.

性质 7.1 在算法 7.5 中, 若记

$$\widetilde{\boldsymbol{Q}}_k = \boldsymbol{Q}_1\boldsymbol{Q}_2\cdots\boldsymbol{Q}_k, \quad \widetilde{\boldsymbol{R}}_k = \boldsymbol{R}_k\boldsymbol{R}_{k-1}\cdots\boldsymbol{R}_1, \tag{7.50}$$

显然 $\widetilde{\boldsymbol{Q}}_k$ 为正交矩阵, $\widetilde{\boldsymbol{R}}_k$ 为上三角矩阵. 则有




Back
Close

(1) $\boldsymbol{Q}_k$ 和 $\boldsymbol{A}_{k+1}$ 都是上 Hessenberg 矩阵.

(2) $\boldsymbol{A}_{k+1} = \widetilde{\boldsymbol{Q}}_k^{\mathrm{T}} \boldsymbol{A} \widetilde{\boldsymbol{Q}}_k \sim \boldsymbol{A}$ (相似性). (7.51)

(3) $\boldsymbol{A}^k = \widetilde{\boldsymbol{Q}}_k \widetilde{\boldsymbol{R}}_k$ ($\boldsymbol{A}$ 的 k 次幂 $\boldsymbol{A}^k$ 的 QR 分解). (7.52)

证明 用归纳法证明性质 (3). 当 $k = 1$ 时, 有

$$\boldsymbol{A} = \boldsymbol{A}_1 = \boldsymbol{Q}_1 \boldsymbol{R}_1 = \widetilde{\boldsymbol{Q}}_1 \widetilde{\boldsymbol{R}}_1.$$

设 $\boldsymbol{A}^{k-1} = \widetilde{\boldsymbol{Q}}_{k-1} \widetilde{\boldsymbol{R}}_{k-1}$, 则

$$\begin{aligned} \boldsymbol{A}^k &= \boldsymbol{A}(\widetilde{\boldsymbol{Q}}_{k-1} \widetilde{\boldsymbol{R}}_{k-1}) = \widetilde{\boldsymbol{Q}}_{k-1} (\widetilde{\boldsymbol{Q}}_{k-1}^{\mathrm{T}} \boldsymbol{A} \widetilde{\boldsymbol{Q}}_{k-1}) \widetilde{\boldsymbol{R}}_{k-1} \\ &= \widetilde{\boldsymbol{Q}}_{k-1} \boldsymbol{A}_k \widetilde{\boldsymbol{R}}_{k-1} = \widetilde{\boldsymbol{Q}}_{k-1} \boldsymbol{Q}_k \boldsymbol{R}_k \widetilde{\boldsymbol{R}}_{k-1} = \widetilde{\boldsymbol{Q}}_k \widetilde{\boldsymbol{R}}_k. \end{aligned}$$

证毕. □




性质 (1) 称为上 Hessenberg 形在 QR 变换下的不变性. 它的意义是, 算法 7.5 可以始终对上 Hessenberg 矩阵进行. 这时, QR 分解中每列的消元只要作一次 Givens 变换, 从而简化了 QR 变换的计算. 性质 (2) 表示, 由 QR 方法生成的矩阵序列 $\{\boldsymbol{A}_k\}$ 保持原矩阵 $\boldsymbol{A}$ 的特征值不变. 性质 (3) 说明, QR 方法即 QR 变换过程, 实质上是对 $\boldsymbol{A}$ 的 k 次幂 $\boldsymbol{A}^k$ 进行 QR 分解的过程. 由此可见, QR 方法与幂法有内在的联系. 下面给出 QR 方法的一个最简单的收敛性定理, 它表明, 在一定条件下可以把 QR 方法看成幂法的推广. 为此, 先给出下面的引理.

引理 7.1 设 $\boldsymbol{Q} = [\boldsymbol{q}_1, \boldsymbol{Q}_{n-1}] \in \mathbb{R}^{n\times n}$ 为正交矩阵, 其中 $\boldsymbol{q}_1$ 为 $\boldsymbol{Q}$ 的第 1 列. 对于矩阵 $\boldsymbol{A}$, 若记

$$\boldsymbol{Q}^{\mathrm{T}}\boldsymbol{A}\boldsymbol{Q} = \begin{bmatrix} \boldsymbol{q}_1^{\mathrm{T}}\boldsymbol{A}\boldsymbol{q}_1 & \boldsymbol{\beta}^{\mathrm{T}} \\ \boldsymbol{\alpha} & \boldsymbol{C} \end{bmatrix}, \quad \text{其中 } \boldsymbol{\alpha} = \boldsymbol{Q}_{n-1}^{\mathrm{T}}\boldsymbol{A}\boldsymbol{q}_1 \in \mathbb{R}^{n-1},$$

则有

$$\|\boldsymbol{\alpha}\|_2 = \|\boldsymbol{A}\boldsymbol{q}_1 - (\boldsymbol{q}_1^{\mathrm{T}}\boldsymbol{A}\boldsymbol{q}_1)\boldsymbol{q}_1\|_2. \tag{7.53}$$

证明 因为对任意的 $\boldsymbol{x} \in \mathbb{R}^n$ 都有 $\|\boldsymbol{Q}^{\mathrm{T}}\boldsymbol{x}\|_2 = \|\boldsymbol{x}\|_2$. 故

$$\begin{aligned}
&\|\boldsymbol{A}\boldsymbol{q}_1 - (\boldsymbol{q}_1^{\mathrm{T}}\boldsymbol{A}\boldsymbol{q}_1)\boldsymbol{q}_1\|_2 = \|[\boldsymbol{q}_1, \boldsymbol{Q}_{n-1}]^{\mathrm{T}}[\boldsymbol{A}\boldsymbol{q}_1 - (\boldsymbol{q}_1^{\mathrm{T}}\boldsymbol{A}\boldsymbol{q}_1)\boldsymbol{q}_1]\|_2 \\
&= \left\| \begin{bmatrix} \boldsymbol{q}_1^{\mathrm{T}}[\boldsymbol{A}\boldsymbol{q}_1 - (\boldsymbol{q}_1^{\mathrm{T}}\boldsymbol{A}\boldsymbol{q}_1)\boldsymbol{q}_1] \\ \boldsymbol{Q}_{n-1}^{\mathrm{T}}[\boldsymbol{A}\boldsymbol{q}_1 - (\boldsymbol{q}_1^{\mathrm{T}}\boldsymbol{A}\boldsymbol{q}_1)\boldsymbol{q}_1] \end{bmatrix} \right\|_2 = \left\| \begin{bmatrix} 0 \\ \boldsymbol{\alpha} \end{bmatrix} \right\|_2 = \|\boldsymbol{\alpha}\|_2.
\end{aligned}$$

证毕. □

定理 7.13 设对称矩阵 $\boldsymbol{A} \in \mathbb{R}^{n\times n}$ 满足

$$|\lambda_1| > |\lambda_2| \geqslant \cdots \geqslant |\lambda_n| > 0,$$

对应的规范正交化特征向量为 $\boldsymbol{x}_1, \boldsymbol{x}_2, \cdots, \boldsymbol{x}_n$. 如果单位坐标向量

$$\boldsymbol{e}_1 = (1, 0, \cdots, 0)^{\mathrm{T}} = \sum_{i=1}^{n} \alpha_i \boldsymbol{x}_i$$

中的 $\alpha_1 \neq 0$, 那么由算法 7.5 生成的矩阵序列 $\{\boldsymbol{A}_k\}$ 具有收敛性质

$$\lim_{k\to\infty} \boldsymbol{A}_k \boldsymbol{e}_1 = \lambda_1 \boldsymbol{e}_1. \tag{7.54}$$

证明 记 $\widetilde{\boldsymbol{Q}}_k$ 的第 1 列为 $\widetilde{\boldsymbol{q}}_1^{(k)} = \widetilde{\boldsymbol{Q}}_k \boldsymbol{e}_1$, $\widetilde{\boldsymbol{R}}_k$ 的第 1 个对角元为 $\widetilde{r}_{11}^{(k)}$, 则由式 (7.52) 可知

$$\boldsymbol{A}^k \boldsymbol{e}_1 = \widetilde{\boldsymbol{Q}}_k \widetilde{\boldsymbol{R}}_k \boldsymbol{e}_1 = \widetilde{\boldsymbol{Q}}_k (\widetilde{r}_{11}^{(k)} \boldsymbol{e}_1) = \widetilde{r}_{11}^{(k)} \widetilde{\boldsymbol{q}}_1^{(k)}.$$

注意到 $\|\widetilde{\boldsymbol{q}}_1^{(k)}\|_2 = 1$, $\widetilde{r}_{11}^{(k)} \neq 0$ (因矩阵 $\boldsymbol{A}$ 非奇异). 从而 $|\widetilde{r}_{11}^{(k)}| = \|\boldsymbol{A}^k \boldsymbol{e}_1\|_2$. 于是, 根据幂法的收敛性, 有

$$\lim_{k\to\infty} \widetilde{\boldsymbol{q}}_1^{(k)} = \lim_{k\to\infty} \frac{\boldsymbol{A}^k \boldsymbol{e}_1}{\widetilde{r}_{11}^{(k)}} = \boldsymbol{z}_1, \tag{7.55}$$

式中: $\boldsymbol{z}_1$ 为矩阵 $\boldsymbol{A}$ 的对应于 λ_1 的规范化特征向量 (可以相差一个

常数因子 ± 1). 进而, 由式 (7.51), 可以把 $\boldsymbol{A}_{k+1}$ 写成

$$\boldsymbol{A}_{k+1} = \widetilde{\boldsymbol{Q}}_k^{\mathrm{T}} \boldsymbol{A} \widetilde{\boldsymbol{Q}}_k = [\widetilde{\boldsymbol{q}}_1^{(k)}, \widetilde{\boldsymbol{Q}}_{k-1}]^{\mathrm{T}} \boldsymbol{A} [\widetilde{\boldsymbol{q}}_1^{(k)}, \widetilde{\boldsymbol{Q}}_{k-1}]$$

$$= \begin{bmatrix} a_{11}^{(k+1)} & * \\ \boldsymbol{\alpha}^{(k+1)} & * \end{bmatrix},$$

式中: $a_{11}^{(k+1)} = (\widetilde{\boldsymbol{q}}_1^{(k)})^{\mathrm{T}} \boldsymbol{A} \widetilde{\boldsymbol{q}}_1^{(k)}$, $\boldsymbol{\alpha}^{(k+1)} = (\widetilde{\boldsymbol{Q}}_{k-1}^{(k)})^{\mathrm{T}} \boldsymbol{A} \widetilde{\boldsymbol{q}}_1^{(k)}$. 根据式 (7.55), 得

$$\lim_{k\to\infty} a_{11}^{(k+1)} = \lim_{k\to\infty} (\widetilde{\boldsymbol{q}}_1^{(k)})^{\mathrm{T}} \boldsymbol{A} \widetilde{\boldsymbol{q}}_1^{(k)} = \boldsymbol{z}_1^{\mathrm{T}} \boldsymbol{A} \boldsymbol{z}_1 = \lambda_1.$$

故由引理 7.1, 得

$$\lim_{k\to\infty} \|\boldsymbol{\alpha}^{(k+1)}\|_2 = \lim_{k\to\infty} \|\boldsymbol{A}\widetilde{\boldsymbol{q}}_1^{(k)} - [(\widetilde{\boldsymbol{q}}_1^{(k)})^{\mathrm{T}} \boldsymbol{A} \widetilde{\boldsymbol{q}}_1^{(k)}] \widetilde{\boldsymbol{q}}_1^{(k)}\|_2$$

$$= \|\boldsymbol{A}\boldsymbol{z}_1 - [\boldsymbol{z}_1^{\mathrm{T}} \boldsymbol{A} \boldsymbol{z}_1] \boldsymbol{z}_1\|_2 = 0.$$

因此, $\lim_{k\to\infty} \boldsymbol{\alpha}^{(k+1)} = \mathbf{0}$. 于是有

$$\lim_{k\to\infty} \boldsymbol{A}_{k+1} = \begin{bmatrix} \lambda_1 & * \\ \mathbf{0} & * \end{bmatrix},$$

即式 (7.54) 成立. 证毕. □

作为定理 7.13 的推广, 有下面的收敛性结果.

定理 7.14 设 $\boldsymbol{A} = \boldsymbol{X}\boldsymbol{\Lambda}\boldsymbol{X}^{-1}$, 其中 $\boldsymbol{\Lambda} = \mathrm{diag}(\lambda_1, \lambda_2, \cdots, \lambda_n)$. 如果 ① $|\lambda_1| > |\lambda_2| > \cdots > |\lambda_n| > 0$; ② $\boldsymbol{X}^{-1}$ 具有 LU 分解: $\boldsymbol{X}^{-1} = \boldsymbol{L}\boldsymbol{U}$. 则 QR 方法基本收敛于上三角矩阵.

证明 由式 (7.51) 可知, 只需分析 $\widetilde{\boldsymbol{Q}}_k$ 的极限情况. 而由式 (7.52), 只需分析 $\boldsymbol{A}^k$ 的极限情况. 注意到

$$\boldsymbol{A}^k = \boldsymbol{X}\boldsymbol{\Lambda}^k\boldsymbol{X}^{-1} = \boldsymbol{X}\boldsymbol{\Lambda}^k\boldsymbol{L}\boldsymbol{U} = \boldsymbol{X}(\boldsymbol{\Lambda}^k\boldsymbol{L}\boldsymbol{\Lambda}^{-k})\boldsymbol{\Lambda}^k\boldsymbol{U}.$$

令

$$\boldsymbol{\Lambda}^k \boldsymbol{L} \boldsymbol{\Lambda}^{-k} = \boldsymbol{I} + \boldsymbol{E}_k,$$

则

$$\boldsymbol{A}^k = \boldsymbol{X}(\boldsymbol{I} + \boldsymbol{E}_k)\boldsymbol{\Lambda}^k \boldsymbol{U}.$$

由于 $\boldsymbol{L}$ 是单位下三角形, 故

$$(\boldsymbol{E}_k)_{ij} = \begin{cases} 0, & i \leqslant j, \\ l_{ij}(\lambda_i/\lambda_j)^k, & i > j. \end{cases}$$

由假设条件 ① 知, $\boldsymbol{E}_k \to \boldsymbol{O}$ 且 $(\boldsymbol{E}_k)_{ij}$ 的收敛速度是 $|\lambda_i/\lambda_j|$.

设 $\boldsymbol{X} = \boldsymbol{Q}\boldsymbol{R}$ 且 $\boldsymbol{R}$ 的对角元均为正数, 则有

$$\begin{aligned} \boldsymbol{A}^k &= \boldsymbol{Q}\boldsymbol{R}(\boldsymbol{I} + \boldsymbol{E}_k)\boldsymbol{\Lambda}^k \boldsymbol{U} \\ &= \boldsymbol{Q}(\boldsymbol{I} + \boldsymbol{R}\boldsymbol{E}_k\boldsymbol{R}^{-1})\boldsymbol{R}\boldsymbol{\Lambda}^k \boldsymbol{U}. \end{aligned}$$

因为 $\boldsymbol{E}_k \to \boldsymbol{O}\,(k \to \infty)$, 故当 k 充分大时, $\boldsymbol{I}+\boldsymbol{R}\boldsymbol{E}_k\boldsymbol{R}^{-1}$ 非奇异, 所以有唯一的 QR 分解 $\boldsymbol{I}+\boldsymbol{R}\boldsymbol{E}_k\boldsymbol{R}^{-1}=\widehat{\boldsymbol{Q}}_k\widehat{\boldsymbol{R}}_k$ ($\widehat{\boldsymbol{R}}_k$ 的对角元为正), 而且当 $k\to\infty$ 时, $\widehat{\boldsymbol{Q}}_k\to\boldsymbol{I}$, $\widehat{\boldsymbol{R}}_k\to\boldsymbol{I}$. 此时, $\boldsymbol{A}^k$ 有如下分解

$$\boldsymbol{A}^k=(\boldsymbol{Q}\widehat{\boldsymbol{Q}}_k)(\widehat{\boldsymbol{R}}_k\boldsymbol{R}\boldsymbol{\Lambda}^k\boldsymbol{U}).$$

妨碍上式成为 $\boldsymbol{A}^k$ 的 QR 分解的仅仅是上式右端第 2 个因子 (上三角矩阵) 的对角元可能非正. 为补救这一点, 可引入两个对角正交矩阵

$$\boldsymbol{D}_1=\operatorname{diag}\left(\frac{\lambda_1}{|\lambda_1|},\cdots,\frac{\lambda_n}{|\lambda_n|}\right),\quad \boldsymbol{D}_2=\operatorname{diag}\left(\frac{\boldsymbol{U}_{11}}{|\boldsymbol{U}_{11}|},\cdots,\frac{\boldsymbol{U}_{nn}}{|\boldsymbol{U}_{nn}|}\right),$$

式中: $\boldsymbol{U}_{ii}\,(i=1,2,\cdots,n)$ 为矩阵 $\boldsymbol{U}$ 的对角元. 于是

$$\boldsymbol{A}^k=\left((\boldsymbol{Q}\widehat{\boldsymbol{Q}}_k)(\boldsymbol{D}_2\boldsymbol{D}_1^k)\right)\left(\boldsymbol{D}_1^{-k}\boldsymbol{D}_2^{-1}\widehat{\boldsymbol{R}}_k\boldsymbol{R}\boldsymbol{\Lambda}^k\boldsymbol{U}\right)$$

是 $\boldsymbol{A}^k$ 的唯一 QR 分解. 从而由式 (7.51), 有

$$\boldsymbol{A}_{k+1} = \big(\boldsymbol{Q}\widehat{\boldsymbol{Q}}_k\boldsymbol{D}_2\boldsymbol{D}_1^k\big)^{\mathrm{T}}\boldsymbol{A}\big(\boldsymbol{Q}\widehat{\boldsymbol{Q}}_k\boldsymbol{D}_2\boldsymbol{D}_1^k\big).$$

将 $\boldsymbol{A} = \boldsymbol{X}\boldsymbol{\Lambda}\boldsymbol{X}^{-1} = \boldsymbol{Q}\boldsymbol{R}\boldsymbol{\Lambda}\boldsymbol{R}^{-1}\boldsymbol{Q}^{-1}$ 代入上式, 得

$$\boldsymbol{A}_{k+1} = \big(\boldsymbol{D}_2\boldsymbol{D}_1^k\big)^{\mathrm{T}}\big(\widehat{\boldsymbol{Q}}_k^{-1}\boldsymbol{R}\boldsymbol{\Lambda}\boldsymbol{R}^{-1}\widehat{\boldsymbol{Q}}_k\big)\big(\boldsymbol{D}_2\boldsymbol{D}_1^k\big).$$

因为 $\widehat{\boldsymbol{Q}}_k^{-1}\boldsymbol{R}\boldsymbol{\Lambda}\boldsymbol{R}^{-1}\widehat{\boldsymbol{Q}}_k \to \boldsymbol{R}\boldsymbol{\Lambda}\boldsymbol{R}^{-1} \equiv \bar{\boldsymbol{R}}$ (上三角矩阵), 所以 $\boldsymbol{A}_k$ 的对角线以下元素收敛于 0. 因为 $\boldsymbol{D}_1^k$ 可能不收敛, 故 $\boldsymbol{A}_k$ 基本收敛于 $\bar{\boldsymbol{R}}$. 证毕. □

§7.4.4 带原点位移的 QR 方法

定理 7.14 表明 $\boldsymbol{A}_k$ 的对角元 $a_{ii}^{(k)} \to \lambda_i\,(k \to \infty)$. 从证明过程中可以看出 $\boldsymbol{A}_k$ 的下三角部分的元素趋于 0 的速度由 $k \to \infty$

时, $\widehat{\boldsymbol{Q}}_k \to \boldsymbol{I}$ 和 $\widehat{\boldsymbol{R}}_k \to \boldsymbol{I}$ 的速度, 亦即由 $\boldsymbol{E}_k \to \boldsymbol{O}$ 的速度所决定. 而从矩阵 $\boldsymbol{E}_k$ 的构成可以看出, $\boldsymbol{E}_k$ 的第 i 行元素趋于 0 的速度由 $|\lambda_i/\lambda_{i-1}|$ 决定, 第 i 列元素趋于 0 的速度由 $|\lambda_{i+1}/\lambda_i|$ 决定. 可以证明, $\boldsymbol{A}_k$ 的下三角部分的元素趋于 0 的情况也是这样. 所以 $a_{ii}^{(k)} \to \lambda_i$ 的速度由 $\boldsymbol{A}_k$ 的第 i 行和第 i 列的下三角部分的元素趋于 0 的速度确定, 这个速度即为 $O(\rho_i)$, 其中

$$\rho_i = \max\left\{\frac{|\lambda_i|}{|\lambda_{i-1}|},\ \frac{|\lambda_{i+1}|}{|\lambda_i|},\ \lambda_0 = +\infty,\ \lambda_{n+1} = 0\right\}.$$

在实际计算中, 线性收敛速度是不令人满意的, 特别是当 ρ_i 不算很小时, 收敛将是十分缓慢的. 为此, 将使用原点位移的方法进行加速. 现在 $\rho_n = |\lambda_n/\lambda_{n-1}|$, 如果将算法用于矩阵 $\boldsymbol{A} - s\boldsymbol{I}$, 则 $a_{nn}^{(k)}$ 将以商 $|\lambda_n - s|/|\lambda_{n-1} - s|$ 线性收敛于 $\lambda_n - s$. 当 s 是 λ_n 的一个较好的近似时, 收敛是很快的. 基于这个想法, 可构造原点位移 QR 方

法如下.

算法 7.6 (原点位移 QR 方法)

步 1, 输入上 Hessenberg 矩阵 $\boldsymbol{A} \in \mathbb{R}^{n\times n}$.

步 2, 记 $\boldsymbol{A}_1 := \boldsymbol{A}$. 对于 $k = 1, 2, \cdots$,

(1) $\boldsymbol{A}_k - s_k\boldsymbol{I} = \boldsymbol{Q}_k\boldsymbol{R}_k$ (QR 分解).

(2) $\boldsymbol{A}_{k+1} = \boldsymbol{Q}_k^{\mathrm{T}}\boldsymbol{A}_k\boldsymbol{Q}_k = \boldsymbol{R}_k\boldsymbol{Q}_k + s_k\boldsymbol{I}$ (正交相似变换).

其中, 由 $\boldsymbol{A}_k$ 到 $\boldsymbol{A}_{k+1}$ 的变换称为原点位移的 QR 变换.

与算法 7.5 相类似, 由此生成的矩阵序列 $\{\boldsymbol{A}_k\}$ 和 $\{\boldsymbol{Q}_k\}$ 都是上 Hessenberg 形, 并且 $\boldsymbol{A}_k$ 与 $\boldsymbol{A}$ 相似. 现在的问题是如何选择 s_k 使得收敛速度加快. 下面讨论一类特殊情形.

假设 $\boldsymbol{A} \in \mathbb{R}^{n\times n}$ 满足定理 7.13 的条件, 并设由基本 QR 方法

生成的 $\boldsymbol{A}_k$ 右下角 2 阶子矩阵

$$\boldsymbol{J}_k = \begin{bmatrix} a_{n-1,n-1}^{(k)} & a_{n-1,n}^{(k)} \\ a_{n,n-1}^{(k)} & a_{n,n}^{(k)} \end{bmatrix} \tag{7.56}$$

的特征值为 $\lambda_{n-1}^{(k)}$ 和 $\lambda_n^{(k)}$, 则有

$$\lim_{k\to\infty} a_{n,n}^{(k)} = \lambda_n, \quad \lim_{k\to\infty} \lambda_{n-1}^{(k)} = \lambda_{n-1}, \quad \lim_{k\to\infty} \lambda_n^{(k)} = \lambda_n. \tag{7.57}$$

下面考虑位移量 s_k 的取法. 因为矩阵 $\boldsymbol{A}_k - s_k\boldsymbol{I}$ 的特征值 $\mu_k = \lambda_k - s_k$, 因此, 如果取 $s_k \approx \lambda_n$, 那么在定理 7.13 的条件下, 有

$$\left|\frac{\mu_{k+1}}{\mu_k}\right| = \left|\frac{\lambda_{k+1} - \lambda_n}{\lambda_k - \lambda_n}\right| < \left|\frac{\lambda_{k+1}}{\lambda_k}\right|, \quad k = 1, 2, \cdots, n-1.$$

这表明算法 7.6 的收敛速度比算法 7.5 快. 因此, 由子矩阵 (7.56) 的收敛性质 (7.57), 位移量 s_k 有下列两种取法:

(1) $s_k = a_{nn}^{(k)}$.

(2) 当 $\lambda_{n-1}^{(k)}$ 和 $\lambda_n^{(k)}$ 为实数时, 取 s_k 为其中与 $a_{nn}^{(k)}$ 最接近的一个.

这两种位移策略特别适用于对称矩阵, 因为此时子矩阵 $\boldsymbol{J}_k$ 对称, 从而它的两个特征值都是实数. 可以证明, 采用这种位移策略的算法 7.6, $\boldsymbol{A}_k$ 基本收敛于上三角矩阵, 收敛是二阶的, 并且 $a_{n,n-1}^{(k)}$ 最先趋于零, 从而首先得到绝对值最小的特征值 λ_n.

§7.4.5 双重步位移隐式 QR 方法

对于一般的实方阵 $\boldsymbol{A} \in \mathbb{R}^{n\times n}$, 前面讨论的算法 7.5 和算法 7.6 都不太好. 首先, 这两种方法都是显式方法, 即每次迭代都需要明显地作矩阵的 QR 分解, 并用所得到的正交矩阵作相似变换 (实际上是作三角矩阵与正交矩阵的乘积), 计算量和存储量都很大. 其次, 基本 QR 算法 7.5 若收敛则是线性的, 而原点位移 QR 算法 7.6

只有当特征值是实数时才有可能改善收敛性. 这是可以理解的, 因为这两种算法都是在实数域上进行运算, 因此当实矩阵 $\boldsymbol{A}$ 有复特征值时, 即使收敛也是很缓慢的.

理论分析和实际计算的经验表明: QR 迭代产生的矩阵序列右下角最先显露 $\boldsymbol{A}$ 的特征值. 在原点位移 QR 方法中正是利用这一特点来选取位移参数 s_k 的. 如果显露的是 $\boldsymbol{A}$ 的实特征值, 即 $a_{nn}^{(k)}$ 是 $\boldsymbol{A}$ 的较好的近似特征值时, 就可以简单地选取 $s_k = a_{nn}^{(k)}$. 然而, 如果显露的是 $\boldsymbol{A}$ 的复共轭特征值时, 即式 (7.56) 中的矩阵 $\boldsymbol{J}_k$ 的特征值是一对互相共轭的复数 μ_1 和 μ_2, 且与 $\boldsymbol{A}$ 的特征值比较接近时, 就应该选择 $\boldsymbol{J}_k$ 某一特征值 μ_i 作为位移参数. 但这样一来, 就引进了复运算, 而这是所不希望的.

实数运算的优点是, 计算简单, 工作量小. 为了用实数运算来求一般实方阵的全部特征值, 特别是复特征值, 可以引进双重步位

移隐式 QR 方法. 它的基本思想是: 首先把原点位移推广到复数域 (当然包括实数), 并且每次迭代作两次原点位移的 QR 变换, 因此称为双重步 QR 方法, 这时的运算都可以是复数; 然后把这两次 QR 变换合在一起, 转化成实数运算, 并且构成隐式方法, 即不明显地进行 QR 变换. 它的优点是, 迭代过程都是实数运算, 原点位移加速了收敛性, 隐式方法可减少计算工作量并节省存储空间.

1. 双重步位移隐式 QR 变换

双重步位移隐式 QR 方法的核心是双重步位移隐式 QR 变换. 设矩阵 $\boldsymbol{A} \in \mathbb{R}^{n\times n}$ 是不可约的上 Hessenberg 矩阵, 它的右下角的 2 阶子矩阵

$$\begin{bmatrix} a_{n-1,n-1} & a_{n-1,n} \\ a_{n,n-1} & a_{n,n} \end{bmatrix}$$

的特征值为 μ_1 和 μ_2, 它们可能都是实数, 也可能是一对共轭复数.

记

$$s = a_{n-1,n-1} + a_{n,n}, \quad t = a_{n-1,n-1}a_{n,n} - a_{n,n-1}a_{n-1,n},$$

s 和 t 都是实数, 则有

$$\mu_1 + \mu_2 = s, \quad \mu_1\mu_2 = t. \tag{7.58}$$

现在取 μ_1 和 μ_2 作为位移量, 作两次原点位移的 QR 变换:

$$\begin{cases} \boldsymbol{A} - \mu_1\boldsymbol{I} = \boldsymbol{Q}_1\boldsymbol{R}_1, \ \boldsymbol{B} = \boldsymbol{Q}_1^{\mathrm{H}}\boldsymbol{A}\boldsymbol{Q}_1 = \boldsymbol{R}_1\boldsymbol{Q}_1 + \mu_1\boldsymbol{I}, \\ \boldsymbol{B} - \mu_2\boldsymbol{I} = \boldsymbol{Q}_2\boldsymbol{R}_2, \ \boldsymbol{C} = \boldsymbol{Q}_2^{\mathrm{H}}\boldsymbol{B}\boldsymbol{Q}_2 = \boldsymbol{R}_2\boldsymbol{Q}_2 + \mu_2\boldsymbol{I}, \end{cases} \tag{7.59}$$

式中: $\boldsymbol{Q}_1, \boldsymbol{Q}_2$ 为酉矩阵, 也是上 Hessenberg 矩阵; $\boldsymbol{B}$ 和 $\boldsymbol{C}$ 都是上 Hessenberg 矩阵; $\boldsymbol{R}_1$ 和 $\boldsymbol{R}_2$ 都是上三角矩阵. 双重步位移的 QR 变换 (7.59) 具有如下性质: 若记

$$\boldsymbol{H} = (\boldsymbol{A} - \mu_2\boldsymbol{I})(\boldsymbol{A} - \mu_1\boldsymbol{I}), \quad \boldsymbol{Q} = \boldsymbol{Q}_1\boldsymbol{Q}_2, \quad \boldsymbol{R} = \boldsymbol{R}_2\boldsymbol{R}_1, \tag{7.60}$$

式中: $\boldsymbol{Q}$ 为酉矩阵; $\boldsymbol{R}$ 为上三角矩阵. 则有

Back

Close

$$\boldsymbol{H} = \boldsymbol{Q}\boldsymbol{R}, \quad \boldsymbol{C} = \boldsymbol{Q}^{\mathrm{H}}\boldsymbol{A}\boldsymbol{Q}. \tag{7.61}$$

事实上, 有

$$\begin{aligned}
\boldsymbol{H} &= (\boldsymbol{A} - \mu_2\boldsymbol{I})(\boldsymbol{A} - \mu_1\boldsymbol{I}) = (\boldsymbol{A} - \mu_2\boldsymbol{I})\boldsymbol{Q}_1\boldsymbol{R}_1 \\
&= \boldsymbol{Q}_1(\boldsymbol{Q}_1^{\mathrm{H}}\boldsymbol{A}\boldsymbol{Q}_1 - \mu_2\boldsymbol{I})\boldsymbol{R}_1 = \boldsymbol{Q}_1(\boldsymbol{B} - \mu_2\boldsymbol{I})\boldsymbol{R}_1 \\
&= \boldsymbol{Q}_1(\boldsymbol{Q}_2\boldsymbol{R}_2)\boldsymbol{R}_1 = \boldsymbol{Q}\boldsymbol{R}, \\
\boldsymbol{C} &= \boldsymbol{Q}_2^{\mathrm{H}}\boldsymbol{B}\boldsymbol{Q}_2 = \boldsymbol{Q}_2^{\mathrm{H}}(\boldsymbol{Q}_1^{\mathrm{H}}\boldsymbol{A}\boldsymbol{Q}_1)\boldsymbol{Q}_2 = \boldsymbol{Q}^{\mathrm{H}}\boldsymbol{A}\boldsymbol{Q}.
\end{aligned}$$

进一步, 注意到

$$\begin{aligned}
\boldsymbol{H} &= \boldsymbol{A}^2 - (\mu_1 + \mu_2)\boldsymbol{A} + \mu_1\mu_2\boldsymbol{I} = \boldsymbol{A}^2 - s\boldsymbol{A} + t\boldsymbol{I} \\
&= \begin{bmatrix}
h_{11} & \times & \times & \times & \cdots & \times \\
h_{21} & \times & \times & \times & \cdots & \times \\
h_{31} & \times & \times & \times & \cdots & \times \\
 & \times & \times & \times & \cdots & \times \\
 & & \ddots & \ddots & \ddots & \vdots \\
 & & & \times & \times & \times
\end{bmatrix}
\end{aligned} \tag{7.62}$$

是实矩阵, 其中

$$\begin{cases} h_{11} = a_{11}^2 + a_{12}a_{21} - sa_{11} + t, \\ h_{21} = a_{21}(a_{11} + a_{22} - s), \\ h_{31} = a_{21}a_{32}. \end{cases} \tag{7.63}$$

所以式 (7.61) 的第 1 式 $\boldsymbol{H} = \boldsymbol{QR}$ 是实的 QR 分解, 即其中的酉矩阵 $\boldsymbol{Q}$ 必定是正交矩阵, $\boldsymbol{R}$ 必是实上三角矩阵. 从而式 (7.61) 的第 2 式为正交相似变换 $\boldsymbol{C} = \boldsymbol{Q}^{\mathrm{T}}\boldsymbol{AQ}$.

综上所述, 为了确保计算得到的 $\boldsymbol{C}$ 仍为实矩阵, 根据式 (7.61), 自然考虑按如下的步骤来计算 $\boldsymbol{C}$:

(1) 计算 $\boldsymbol{H} = \boldsymbol{A}^2 - s\boldsymbol{A} + t\boldsymbol{I}$.

(2) 计算 $\boldsymbol{H}$ 的 QR 分解:$\boldsymbol{H} = \boldsymbol{QR}$.

(3) 计算 $\boldsymbol{C}$ 的正交相似变换: $\boldsymbol{C} = \boldsymbol{Q}^{\mathrm{T}}\boldsymbol{AQ}$.

Back

Close

然而, 如此计算的第 1 步形成 $\boldsymbol{H}$ 的运算量就为 $O(n^3)$, 这使得前面为减少每次迭代所需运算量所做的努力付之东流. 幸运的是, 定理 7.12 表明, 不论采取什么方法计算正交矩阵 $\widetilde{\boldsymbol{Q}}$ 使得 $\widetilde{\boldsymbol{Q}}^{\mathrm{T}}\boldsymbol{A}\widetilde{\boldsymbol{Q}} = \widetilde{\boldsymbol{C}}$ 成为上 Hessenberg 矩阵, 只要保证 $\widetilde{\boldsymbol{Q}}$ 的第 1 列与 $\boldsymbol{Q}$ 的第 1 列一样, 则 $\widetilde{\boldsymbol{C}}$ 就与 $\boldsymbol{C}$ 在本质上是一样的 (所有元素的绝对值相等). 因此, 可以有很大的自由度去寻求更有效的方法来实现 $\boldsymbol{A}$ 到 $\boldsymbol{C}$ 的变换.

首先, 从式 (7.61) 的第 1 式 $\boldsymbol{H} = \boldsymbol{Q}\boldsymbol{R}$ 知, $\boldsymbol{Q}$ 的第 1 列与 $\boldsymbol{H}$ 的第 1 列共线, 即 $\boldsymbol{Q}\boldsymbol{e}_1$ 由 $\boldsymbol{H}\boldsymbol{e}_1$ 单位化得到. 而由式 (7.62) 容易算出

$$\boldsymbol{H}\boldsymbol{e}_1 = (h_{11}, h_{21}, h_{31}, 0, \cdots, 0)^{\mathrm{T}},$$

式中: h_{11}, h_{21}, h_{31} 由式 (7.63) 得出.

其次, 如果 Householder 变换 $\boldsymbol{P}_0$ 将 $\boldsymbol{H}\boldsymbol{e}_1$ 变为 $\alpha\boldsymbol{e}_1$, 即 $\boldsymbol{P}_0(\boldsymbol{H}\boldsymbol{e}_1) = \alpha\boldsymbol{e}_1$, 其中 $\alpha \in \mathbb{R}$, 则易知, $\boldsymbol{P}_0$ 的第 1 列就与 $\boldsymbol{H}\boldsymbol{e}_1$ 共线, 从而

$\boldsymbol{P}_0\boldsymbol{e}_1 = \boldsymbol{Q}\boldsymbol{e}_1$. 而由 Householder 变换的性质, $\boldsymbol{P}_0$ 可按如下方式确定:

$$\boldsymbol{P}_0 = \mathrm{diag}(\widetilde{\boldsymbol{P}}_0, \boldsymbol{I}_{n-3}),$$

式中:

$$\widetilde{\boldsymbol{P}}_0 = \boldsymbol{I}_3 - \beta\boldsymbol{v}\boldsymbol{v}^{\mathrm{T}}, \quad \beta = 2/\boldsymbol{v}^{\mathrm{T}}\boldsymbol{v},$$

$$\boldsymbol{v} = (h_{11} + \alpha\mathrm{sign}(h_{11}), h_{21}, h_{31})^{\mathrm{T}}, \ \alpha = \sqrt{h_{11}^2 + h_{21}^2 + h_{31}^2}.$$

现令

$$\boldsymbol{D}_1 = \boldsymbol{P}_0\boldsymbol{A}\boldsymbol{P}_0,$$

那么只要找到第 1 列为 $\boldsymbol{e}_1$ 的正交矩阵 $\widetilde{\boldsymbol{Q}}$ 使 $\widetilde{\boldsymbol{Q}}^{\mathrm{T}}\boldsymbol{D}_1\widetilde{\boldsymbol{Q}} = \widetilde{\boldsymbol{H}}$ 为上 Hessenberg 矩阵, 那么 $\widetilde{\boldsymbol{H}}$ 就是希望得到的矩阵 $\boldsymbol{C}$. 这只需确定 $n-1$ 个 Householder 变换 $\boldsymbol{P}_1, \cdots, \boldsymbol{P}_{n-1}$, 使

$$(\boldsymbol{P}_{n-1}\cdots\boldsymbol{P}_1)\boldsymbol{D}_1(\boldsymbol{P}_1\cdots\boldsymbol{P}_{n-1}) = \widetilde{\boldsymbol{H}}$$




为上 Hessenberg 矩阵, 即有 $\widetilde{\boldsymbol{Q}} = \boldsymbol{P}_1 \cdots \boldsymbol{P}_{n-1}$ 的第 1 列为 $\boldsymbol{e}_1$. 而且由于 $\boldsymbol{D}_1$ 所具有的特殊性质, 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 $\boldsymbol{P}_0$ 将 $\boldsymbol{A}$ 相似变换为 $\boldsymbol{D}_1$ 只改变了 $\boldsymbol{A}$ 的前三行和前三列, 故 $\boldsymbol{D}_1$ 具有如下形状, 即

$$\boldsymbol{D}_1 = \boldsymbol{P}_0 \boldsymbol{A} \boldsymbol{P}_0 = \begin{bmatrix} \times & \times & \times & \times & \cdots & \times & \times \\ \times & \times & \times & \times & \cdots & \times & \times \\ \oplus & \times & \times & \times & \cdots & \times & \times \\ \oplus & \oplus & \times & \times & \cdots & \times & \times \\ & & & \times & \cdots & \vdots & \vdots \\ & & & & \ddots & \vdots & \vdots \\ & & & & & \times & \times \end{bmatrix},$$

仅比上 Hessenberg 形多 3 个非零元 “$\oplus$”. 由 $\boldsymbol{D}_1$ 的这种特殊性, 易知用来约化 $\boldsymbol{D}_1$ 为上 Hessenberg 形的第一个 Householder 变换 $\boldsymbol{P}_1$

具有如下形状, 即

$$\boldsymbol{P}_1 = \mathrm{diag}(1, \widetilde{\boldsymbol{P}}_1, \boldsymbol{I}_{n-4}),$$

式中: $\widetilde{\boldsymbol{P}}_1$ 为 3 阶 Householder 变换, 而且 $\boldsymbol{P}_1\boldsymbol{D}_1\boldsymbol{P}_1$ 具有如下形状, 即

$$\boldsymbol{D}_2 = \boldsymbol{P}_1\boldsymbol{D}_1\boldsymbol{P}_1 = \boldsymbol{P}_1\boldsymbol{P}_0\boldsymbol{A}\boldsymbol{P}_0\boldsymbol{P}_1 = \begin{bmatrix} \times & \times & \times & \times & \cdots & \times & \times \\ \times & \times & \times & \times & \cdots & \times & \times \\ 0 & \times & \times & \times & \cdots & \times & \times \\ 0 & \oplus & \times & \times & \cdots & \times & \times \\ & \oplus & \oplus & \times & \cdots & \vdots & \vdots \\ & & & & \ddots & \vdots & \vdots \\ & & & & & \times & \times \end{bmatrix},$$

如此递推地进行下去, 不难发现, 第 k 次约化所用的 Householder 变换 $\boldsymbol{P}_k$ 具有如下形状, 即

$$\boldsymbol{P}_k = \mathrm{diag}(\boldsymbol{I}_k, \widetilde{\boldsymbol{P}}_k, \boldsymbol{I}_{n-k-3}),$$

式中: $\widetilde{\boldsymbol{P}}_k$ 为 3 阶 Householder 变换, $k = 2, \cdots, n-3$, 而且 $\boldsymbol{D}_{n-2} := \boldsymbol{P}_{n-3}\boldsymbol{D}_{n-2}\boldsymbol{P}_{n-3}$ 具有如下形状

$$\begin{aligned}\boldsymbol{D}_{n-2} &= \boldsymbol{P}_{n-3}\boldsymbol{D}_{n-3}\boldsymbol{P}_{n-3} \\ &= \boldsymbol{P}_{n-3}\boldsymbol{P}_{n-4}\cdots\boldsymbol{P}_1\boldsymbol{D}_1\boldsymbol{P}_1\cdots\boldsymbol{P}_{n-4}\boldsymbol{P}_{n-3} \\ &= \begin{bmatrix} \times & \times & \times & \cdots & \times & \times \\ \times & \times & \times & \cdots & \times & \times \\ & \times & \times & \cdots & \times & \times \\ & & \ddots & \ddots & \vdots & \vdots \\ & & & \times & \times & \times \\ & & & \oplus & \times & \times \end{bmatrix}.\end{aligned}$$

因此, 最后一次约化所用的 Householder 变换 $\boldsymbol{P}_{n-2}$ 具有如下形状, 即

$$\boldsymbol{P}_{n-1} = \mathrm{diag}(\boldsymbol{I}_{n-2}, \widetilde{\boldsymbol{P}}_{n-2}),$$

式中: $\widetilde{\boldsymbol{P}}_{n-2}$ 为 2 阶 Householder 变换. 而且 $\boldsymbol{D}_{n-1} := \boldsymbol{P}_{n-2}\boldsymbol{D}_{n-2}\boldsymbol{P}_{n-2}$

具有如下形状, 即

$$\begin{aligned}\boldsymbol{D}_{n-1} &= \boldsymbol{P}_{n-2}\boldsymbol{D}_{n-2}\boldsymbol{P}_{n-2} \\ &= \boldsymbol{P}_{n-2}\cdots\boldsymbol{P}_1\boldsymbol{P}_0\boldsymbol{A}\boldsymbol{P}_0\boldsymbol{P}_1\cdots\boldsymbol{P}_{n-2} \\ &= \begin{bmatrix} \times & \times & \times & \cdots & \times & \times \\ \times & \times & \times & \cdots & \times & \times \\ & \times & \times & \cdots & \times & \times \\ & & \ddots & \ddots & \vdots & \vdots \\ & & & \times & \times & \times \\ & & & & \times & \times \end{bmatrix} := \boldsymbol{D}. \end{aligned} \tag{7.64}$$

这样, 就找到了一种由 $\boldsymbol{A}$ 到 $\boldsymbol{C}$ 的变换方法, 它既避免了复运算的出现, 又减少了运算量. 当然, 这一变换过程对 $\boldsymbol{J}_k$ 的两个特征值都是实数的情形也是可行的. 因此, 不必在选取位移参数时区别显露的是实特征值还是复共轭特征值的情形, 而只需取作 $\boldsymbol{J}_k$ 的两个特征值即可.

定理 7.15 由式 (7.64) 得到的矩阵 $\boldsymbol{D}$“基本收敛”于变换 (7.61) 中的矩阵 $\boldsymbol{C}$.

证明 若记正交矩阵

$$\boldsymbol{U} = \boldsymbol{P}_0\boldsymbol{P}_1\cdots\boldsymbol{P}_{n-2},$$

则式 (7.64) 即为 $\boldsymbol{D} = \boldsymbol{U}^{\mathrm{T}}\boldsymbol{A}\boldsymbol{U}$. 易知, $\boldsymbol{U}$ 的第 1 列与 $\boldsymbol{P}_0$ 的第 1 列相同, 即

$$\boldsymbol{U}\boldsymbol{e}_1 = \boldsymbol{P}_0\boldsymbol{e}_1.$$

另外, 用 Householder 变换对矩阵 $\boldsymbol{H}$ 作 QR 分解的过程是

$$\widetilde{\boldsymbol{P}}_{n-1}\cdots\widetilde{\boldsymbol{P}}_2\widetilde{\boldsymbol{P}}_1\boldsymbol{H} = \boldsymbol{R},$$

式中: $\widetilde{\boldsymbol{P}}_1 = \boldsymbol{P}_0$. 对于 $i = 2, \cdots, n-2$, 有

$$\widetilde{\boldsymbol{P}}_i = \begin{bmatrix} \boldsymbol{I}_{i-1} & & \\ & \widetilde{\boldsymbol{V}}_i & \\ & & \boldsymbol{I}_{n-i-2} \end{bmatrix},$$

式中: $\widetilde{\boldsymbol{V}}_i \in \mathbb{R}^{3\times 3}$ 为 Householder 矩阵.

当 $i = n-1$ 时, 有

$$\widetilde{\boldsymbol{P}}_{n-1} = \begin{bmatrix} \boldsymbol{I}_{n-2} & \\ & \widetilde{\boldsymbol{V}}_{n-1} \end{bmatrix},$$

式中: $\widetilde{\boldsymbol{V}}_{n-1} \in \mathbb{R}^{2\times 2}$ 为 Householder 矩阵. 若记正交矩阵

$$\boldsymbol{Q} = \widetilde{\boldsymbol{P}}_1\widetilde{\boldsymbol{P}}_2\cdots\widetilde{\boldsymbol{P}}_{n-1} = \boldsymbol{P}_0\widetilde{\boldsymbol{P}}_2\cdots\widetilde{\boldsymbol{P}}_{n-1},$$

则由 $\boldsymbol{H} = \boldsymbol{QR}$. 注意到 $\boldsymbol{Q}$ 的第 1 列

$$\boldsymbol{Q}\boldsymbol{e}_1 = \boldsymbol{P}_0\boldsymbol{e}_1 = \boldsymbol{U}\boldsymbol{e}_1,$$

根据定理 7.12, 在化上 Hessenberg 形 $\boldsymbol{C} = \boldsymbol{Q}^{\mathrm{T}}\boldsymbol{AQ}$ 的变换中, 矩阵 $\boldsymbol{Q}$ 和 $\boldsymbol{C}$ 本质上由 $\boldsymbol{Q}$ 的第 1 列唯一确定 (相差一个 1 或 -1 因子的意义下). 因此, 定理的结论成立. 证毕. □

综上所述, 可得双重步位移的 QR 变换的迭代过程.

算法 7.7 (双重步位移隐式 QR 变换)

步 1, 输入不可约上 Hessenberg 矩阵 $\boldsymbol{A} = (a_{ij}) \in \mathbb{R}^{n\times n}$.

步 2, $k := 0$; $m := n-1$; $s := a_{mm} + a_{nn}$;

$$t := a_{mm}a_{nn} - a_{mn}a_{nm};$$

$$x := a_{11}^2 + a_{12}a_{21} - sa_{11} + t;$$

$$y := a_{21}(a_{11} + a_{22} - s);\ z := a_{21}a_{32}.$$

步 3, 若 $k = n-2$, 则转步 5; 否则, 确定 Householder 矩阵 $\widetilde{\boldsymbol{P}}_k \in \mathbb{R}^{3\times 3}$ 使

$$\widetilde{\boldsymbol{P}}_k \begin{bmatrix} x \\ y \\ z \end{bmatrix} = \begin{bmatrix} * \\ 0 \\ 0 \end{bmatrix},$$

Back

Close

置 $\boldsymbol{A} := \boldsymbol{P}_k \boldsymbol{A} \boldsymbol{P}_k$, 其中 $\boldsymbol{P}_k = \mathrm{diag}(\boldsymbol{I}_k, \widetilde{\boldsymbol{P}}_k, \boldsymbol{I}_{n-k-3})$.

步 4, 更新:

$$x := a_{k+2,k+1}, \quad y := a_{k+3,k+1}, \quad z := \begin{cases} a_{k+4,k+1}, & k < n-3 \\ 0, & k = n-3. \end{cases}$$

置 $k := k+1$, 转步 3.

步 5 确定 Householder 矩阵 $\widetilde{\boldsymbol{P}}_{n-2} \in \mathbb{R}^{2\times 2}$ 使

$$\widetilde{\boldsymbol{P}}_{n-2} \begin{bmatrix} x \\ y \end{bmatrix} = \begin{bmatrix} * \\ 0 \end{bmatrix},$$

置 $\boldsymbol{A} := \boldsymbol{P}_{n-2} \boldsymbol{A} \boldsymbol{P}_{n-2}$, 其中 $\boldsymbol{P}_k = \mathrm{diag}(\boldsymbol{I}_{n-2}, \widetilde{\boldsymbol{P}}_{n-2})$. 迭代结束.

算法 7.7 的运算量是 $6n^2$. 如果需要把正交变换累积起来, 还需再增加运算量 $6n^2$.

双重步位移隐式 QR 变换的 MATLAB 程序如下:




Back
Close

```
function A=ddi_qrtran(A,n)
%双重位移隐式QR变换
%输入:A为不可约n阶上Hessenberg矩阵,
%      其最后2x2阶主子阵有特征值a和b
%输出:A=Q^TAQ,这里Q=P1...P_{n-2}是
%      一系列Householder矩阵的乘积
%      且Q^T(A-aI)(A-bI)是上三角形矩阵
I3=eye(3); I2=eye(2); s=A(n-1,n-1)+A(n,n);
t=A(n-1,n-1)*A(n,n)-A(n-1,n)*A(n,n-1);
x=A(1,1)^2+A(1,2)*A(2,1)-s*A(1,1)+t;
y=A(2,1)*(A(1,1)+A(2,2)-s); z=A(2,1)*A(3,2);
for k=0:n-3
   [v,beta]=mhouse([x,y,z]'); q=max(1,k);
```




Back
Close

```
    A(k+1:k+3,q:n)=(I3-beta*v*v')*A(k+1:k+3,q:n);
    r=min(k+4,n);
    A(1:r,k+1:k+3)=A(1:r,k+1:k+3)*(I3-beta*v*v');
    x=A(k+2,k+1); y=A(k+3,k+1);
    if (k<n-3), z=A(k+4,k+1); end
end
[v,beta]=mhouse([x,y]');
A(n-1:n,n-2:n)=(I2-beta*v*v')*A(n-1:n,n-2:n);
A(1:n,n-1:n)=A(1:n,n-1:n)*(I2-beta*v*v');
```

2. 双重步位移隐式 QR 方法

前面的讨论已经解决了用 QR 方法求一个给定实矩阵的实 Schur 分解的几个关键性问题. 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




Back
Close

矩阵的次对角元素何时可以忽略不计. 一种简单而实用的准则是: 当

$$|a_{i+1,i}| \leqslant (|a_{ii}| + |a_{i+1,i+1}|)\varepsilon \tag{7.65}$$

时, 就将 $a_{i+1,i}$ 看作是 0.

将算法 7.4 和算法 7.7 与收敛准则 (7.65) 结合起来, 就得到了双重步位移隐式 QR 方法. 这一算法是计算一给定的 n 阶实矩阵 $\boldsymbol{A}$ 实 Schur 分解: $\boldsymbol{Q}^{\mathrm{T}}\boldsymbol{A}\boldsymbol{Q} = \boldsymbol{T}$, 其中 $\boldsymbol{Q}$ 为正交矩阵, $\boldsymbol{T}$ 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. 由于其中的 QR 变换不明显, 故称为隐式方法.

算法 7.8 (双重步位移隐式 QR 方法)

步 1, 输入矩阵 $\boldsymbol{A} = (a_{ij}) \in \mathbb{R}^{n\times n}$.

步 2, 上 Hessenberg 化. 用算法 7.4 计算 $\boldsymbol{A}$ 的上 Hessenberg

分解 $\boldsymbol{A} := \boldsymbol{Q}^{\mathrm{T}}\boldsymbol{A}\boldsymbol{Q}$.

步 3, 若满足收敛性准则, 停算; 否则返回步 2 继续进行双重步位移隐式 QR 迭代.

双重步位移隐式 QR 方法的 MATLAB 程序如下:

```
function [iter,D]=ddiqr_eig(A,tol)
%用双重步位移隐式QR方法求实方阵的全部特征值
%输入:A为n阶上Hessenberg形实方阵,tol为控制精度(默认是1e-5)
%输出:iter为迭代次数,D为A的全部特征值
if nargin<2, tol=1e-5; end
n=size(A,1);
D=zeros(n,1); i=n; m=n; iter=0;  %初始化
[A]=hessenb(A);  %化矩阵A为Hessenberg矩阵
```




```
while (m>0)
    %用双重位移隐式QR方法进行迭代
    if m<=2
        la=eig(A(1:m,1:m));
        D(1:m)=la'; break;
    end
    iter=iter+1;
    A=ddiqr_tran(A,m);
    %对上Hessenberg矩阵作QR分解,并作正交相似变换
    for k=m-1:-1:1  %下面的程序段是判断是否终止
        if abs(A(k+1,k))<tol
            if m-k<=2
                la=eig(A(k+1:m,k+1:m));
```

```
                j=i-m+k+1; D(j:i)=la';
                i=j-1; m=k; break;
            end
        end
    end
end
```

例 7.8 用双重步位移隐式 QR 方法求例 7.7 中矩阵 $\boldsymbol{A}$ 的全部特征值.

解 编写 M 文件 ex78.m, 在 MATLAB 命令窗口输入 ex78 得

```
>> ex78
算    法          迭代次数     CPU时间
基本QR方法          622         0.0702
双位移QR方法         8          0.0201
```




基本QR方法特征值	双位移QR方法特征值
18.4123 + 0.0000i	18.4123 + 0.0000i
11.1805 + 0.0000i	11.1805 + 0.0000i
1.7099 - 4.2522i	1.7099 - 4.2522i
1.7099 + 4.2522i	1.7099 + 4.2522i
4.4983 + 0.0000i	-2.2327 + 0.0000i
-2.2327 + 0.0000i	4.4983 + 0.0000i
-0.2783 + 0.0000i	-0.2783 + 0.0000i

§7.4.6 特征向量的计算方法

本节讨论在用 (双重步位移隐式) QR 方法求得给定矩阵的特征值之后, 如何求其对应的特征向量. 设 $\boldsymbol{A} \in \mathbb{R}^{n\times n}$, 并假定已用 QR 方法求得 $\boldsymbol{A}$ 的特征值 λ 的一个近似 $\widetilde{\lambda}$. 现在讨论如何求对应于 λ 的特征向量.

目前, 解决这一问题最好的方法是带原点位移的反幂法 (也称为反迭代法, 逆迭代法), 其基本迭代格式如下:

$$(\boldsymbol{A}-\alpha\boldsymbol{I})\boldsymbol{y}^{(k)}=\boldsymbol{x}^{(k-1)}, \tag{7.66a}$$

$$\boldsymbol{x}^{(k)}=\boldsymbol{y}^{(k)}/\|\boldsymbol{y}^{(k)}\|_2, \quad k=1,2,\cdots, \tag{7.66b}$$

式中: α 为选定的位移参数; $\boldsymbol{x}^{(0)}$ 为给定的初始向量.

从式 (7.66a) 可以看出, 每迭代一次就需要解一个线性方程组, 这要比幂法的运算量大得多. 但由于方程组的系数矩阵不随 k 变化, 故可事先对它进行列主元 LU 分解, 然后每次迭代只需解两个三角形方程组即可. 顺便指出, 式 (7.66b) 只是为了防止迭代“溢出”而做的归一化处理, 在实际计算时可以用 $\|\cdot\|_\infty$ 进行归一化.

现假定 $\boldsymbol{A}$ 是非亏损的, 即存在 $\boldsymbol{X} = [\boldsymbol{\xi}_1, \boldsymbol{\xi}_2, \cdots, \boldsymbol{\xi}_n] \in \mathbb{C}^{n\times n}$ 非奇异, 使得

$$\boldsymbol{X}^{-1}\boldsymbol{A}\boldsymbol{X} = \boldsymbol{\Lambda} \equiv \mathrm{diag}\{\lambda_1, \lambda_2, \cdots, \lambda_n\}. \tag{7.67}$$

而且不失一般性, 还可以假设 $\|\boldsymbol{\xi}_i\|_2 = 1\,(i = 1, 2, \cdots, n)$. 现将初始向量 $\boldsymbol{x}^{(0)}$ 按 $\boldsymbol{\xi}_1, \boldsymbol{\xi}_2, \cdots, \boldsymbol{\xi}_n$ 展开

$$\boldsymbol{x}^{(0)} = \sum_{i=1}^{n} \beta_i \boldsymbol{\xi}_i. \tag{7.68}$$

再假定 α 与 $\boldsymbol{A}$ 的特征值 λ_s 最靠近, 并有

$$0 < |\alpha - \lambda_s| < |\alpha - \lambda_i|, \quad i \neq s, \tag{7.69a}$$

$$\beta_s \neq 0. \tag{7.69b}$$

由式 (7.66a)、式 (7.67) 和式 (7.68), 得

$$\begin{aligned}\boldsymbol{y}^{(k)} &= \theta_k(\boldsymbol{A}-\alpha\boldsymbol{I})^{-k}\boldsymbol{x}^{(0)}\\ &= \theta_k\sum_{i=1}^{n}\beta_k(\boldsymbol{A}-\alpha\boldsymbol{I})^{-k}\boldsymbol{\xi}_i = \theta_k\sum_{i=1}^{n}\beta_k(\lambda_i-\alpha)^{-k}\boldsymbol{\xi}_i\\ &= \theta_k\beta_s(\lambda_s-\alpha)^{-k}\left[\boldsymbol{\xi}_s+\sum_{i\neq s}^{n}\left(\frac{\lambda_s-\alpha}{\lambda_i-\alpha}\right)^k\frac{\beta_i}{\beta_s}\boldsymbol{\xi}_i\right]\\ &:= \theta_k\beta_s(\lambda_s-\alpha)^{-k}(\boldsymbol{\xi}_s+\boldsymbol{u}_k), \qquad (7.70)\end{aligned}$$

式中: θ_k 为正数, 而 $\boldsymbol{u}_k\to 0\,(k\to\infty)$, 其收敛速度依赖于

$$\frac{|\lambda_s-\alpha|}{\min\limits_{i\neq s}|\alpha-\lambda_i|}$$

的大小. 将式 (7.70) 代入式 (7.66b), 得

$$\boldsymbol{x}^{(k)} = \frac{\boldsymbol{y}^{(k)}}{\|\boldsymbol{y}^{(k)}\|_2} = \frac{\eta_k}{\|\boldsymbol{\xi}_s+\boldsymbol{u}_k\|_2}(\boldsymbol{\xi}_s+\boldsymbol{u}_k),$$

式中:

$$\eta_k = \frac{\theta_k \beta_s (\lambda_s - \alpha)^{-k}}{\left|\theta_k \beta_s (\lambda_s - \alpha)^{-k}\right|} = \frac{\beta_s}{|\beta_s|}\left(\frac{\lambda_s - \alpha}{|\lambda_s - \alpha|}\right)^{-k}$$

为满足 $|\eta_k| = 1$ 的复数. 从而有

$$\begin{aligned}
\operatorname{dist}\left(\boldsymbol{x}^{(k)}, \boldsymbol{\xi}_s\right) &= \|\mathcal{P}_{\boldsymbol{x}^{(k)}} - \mathcal{P}_{\boldsymbol{\xi}_s}\|_2 = \|\boldsymbol{x}^{(k)}(\boldsymbol{x}^{(k)})^{\mathrm{H}} - \boldsymbol{\xi}_s \boldsymbol{\xi}_s^{\mathrm{H}}\|_2 \\
&= \left\| \frac{(\boldsymbol{\xi}_s + \boldsymbol{u}_k)(\boldsymbol{\xi}_s + \boldsymbol{u}_k)^{\mathrm{H}}}{(\boldsymbol{\xi}_s + \boldsymbol{u}_k)^{\mathrm{H}}(\boldsymbol{\xi}_s + \boldsymbol{u}_k)} - \boldsymbol{\xi}_s \boldsymbol{\xi}_s^{\mathrm{H}} \right\|_2 \\
&\to 0, \quad k \to \infty,
\end{aligned}$$

收敛速度依赖于 $|\lambda_s - \alpha| / \min\limits_{i \neq s} |\alpha - \lambda_i|$ 的大小. 换言之, 即 $\boldsymbol{x}^{(k)}$ 将按方向收敛于 $\boldsymbol{A}$ 的特征向量. α 与 λ_s 越靠近, 收敛速度越快.

由此可见, 从收敛速度的角度来考虑, 用式 (7.66) 迭代时, 自然是 α 取得越靠近 $\boldsymbol{A}$ 的某个特征值越好. 但是, 当 α 与 $\boldsymbol{A}$ 的某个特征值很靠近时, $\boldsymbol{A} - \alpha \boldsymbol{I}$ 就很接近于一个奇异矩阵, 每一步迭代就

需要解一个非常病态的线性方程组. 幸运的是理论分析以及大量的计算实践表明: $\boldsymbol{A}-\alpha\boldsymbol{I}$ 的病态性并不影响其收敛速度, 而且当 α 很靠近 $\boldsymbol{A}$ 的某个特征值时, 常常只需要一次迭代就可以得到相当好的近似特征向量. 为此, 下面进行简要的理论分析.

设 λ 是 $\boldsymbol{A}$ 的特征值, $\boldsymbol{y}\in\mathbb{C}^n$ 满足 $\|\boldsymbol{y}\|_2=1$. 定义

$$\boldsymbol{r}=(\boldsymbol{A}-\lambda\boldsymbol{I})\boldsymbol{y} \tag{7.71}$$

为向量 $\boldsymbol{y}$ 的残差向量. 由式 (7.71), 得

$$(\boldsymbol{A}-\boldsymbol{r}\boldsymbol{y}^{\mathrm{H}})\boldsymbol{y}=\lambda\boldsymbol{y},$$

即 $\boldsymbol{y}$ 是矩阵 $\boldsymbol{A}-\boldsymbol{r}\boldsymbol{y}^{\mathrm{H}}$ 对应于 λ 的特征向量. 如果 $\|\boldsymbol{r}\|_2$ 很小, 则 $\|\boldsymbol{r}\boldsymbol{y}^{\mathrm{H}}\|_2$ 也会很小. 从而, 当 $\|\boldsymbol{r}\|_2$ 很小时, $\boldsymbol{y}$ 就是 $\boldsymbol{A}$ 的对应于 λ 的一个很好的近似特征向量. 即可用 $\boldsymbol{y}$ 的残差向量大小来衡量 $\boldsymbol{y}$ 可否作为对应于 λ 的近似特征向量.




进一步, 假定

$$|\alpha-\lambda|=\min_{\widetilde{\lambda}\in\lambda(\boldsymbol{A})}|\alpha-\widetilde{\lambda}|\leqslant\varepsilon_1, \tag{7.72}$$

式中: ε_1 为很小的正数, 通常 $\varepsilon_1=O(\epsilon)$; ϵ 为机器精度. 再假定对给定的初始向量 $\boldsymbol{x}^{(0)}$, 用列主元 Gauss 消去法求解方程组 (7.66a) 得到向量 $\boldsymbol{y}^{(1)}$ 的计算值 $\widetilde{\boldsymbol{y}}^{(1)}$. 则由 Gauss 消去法的舍入误差分析结果可知

$$(\boldsymbol{A}-\alpha\boldsymbol{I}+\boldsymbol{E})\widetilde{\boldsymbol{y}}^{(1)}=\boldsymbol{x}^{(0)}, \tag{7.73}$$

式中: $\|\boldsymbol{E}\|_2\leqslant\varepsilon_2$ (通常 $\varepsilon_2=O(\epsilon)$). 这样, 由式 (7.66b) 计算得

$$\boldsymbol{x}^{(1)}=\widetilde{\boldsymbol{y}}^{(1)}/\|\widetilde{\boldsymbol{y}}^{(1)}\|_2, \tag{7.74}$$

这里忽略了计算 $\boldsymbol{x}^{(1)}$ 所产生的误差.

利用式 (7.73), 可得向量 $\boldsymbol{x}^{(1)}$ 的残差向量为

$$\boldsymbol{r} = (\boldsymbol{A} - \lambda \boldsymbol{I})\boldsymbol{x}^{(1)} = (\alpha - \lambda)\boldsymbol{x}^{(1)} - \boldsymbol{E}\boldsymbol{x}^{(1)} + \frac{\boldsymbol{x}^{(0)}}{\|\widetilde{\boldsymbol{y}}^{(1)}\|_2}.$$

于是有

$$\|\boldsymbol{r}\|_2 \leqslant \varepsilon_1 + \varepsilon_2 + \|\widetilde{\boldsymbol{y}}^{(1)}\|_2^{-1}, \tag{7.75}$$

这里假定 $\|\boldsymbol{x}^{(0)}\|_2 = 1$.

由此可见, 如果计算得到的 $\widetilde{\boldsymbol{y}}^{(1)}$ 具有很大的范数, 则由式 (7.66) 迭代一次所得的向量就有很小的残差向量, 从而在特征值问题不是十分病态的条件下, 就得到了很好的近似特征向量. 因此, 只需说明在式 (7.72) 成立的前提下确有 $\widetilde{\boldsymbol{y}}^{(1)}$ 的范数很大.

事实上, 设 $\boldsymbol{A} - \alpha \boldsymbol{I} + \boldsymbol{E}$ 的奇异值分解为

$$\boldsymbol{A} - \alpha \boldsymbol{I} + \boldsymbol{E} = \boldsymbol{U}\boldsymbol{\Sigma}\boldsymbol{V}^{\mathrm{H}}, \tag{7.76}$$

式中: $\boldsymbol{U}=[\boldsymbol{u}_1,\boldsymbol{u}_2,\cdots,\boldsymbol{u}_n]$, $\boldsymbol{V}=[\boldsymbol{v}_1,\boldsymbol{v}_2,\cdots,\boldsymbol{v}_n]$;
$\boldsymbol{\Sigma}=\mathrm{diag}(\sigma_1,\sigma_2,\cdots,\sigma_n)$; $\sigma_1\geqslant\sigma_2\geqslant\cdots\geqslant\sigma_n>0$. 由 $\lambda-\alpha$ 是矩阵 $\boldsymbol{A}-\alpha\boldsymbol{I}$ 的特征值, 得

$$\sigma_n(\boldsymbol{A}-\alpha\boldsymbol{I})\leqslant|\lambda-\alpha|\leqslant\varepsilon_1,$$

式中: $\sigma_n(\boldsymbol{A}-\alpha\boldsymbol{I})$ 为 $\boldsymbol{A}-\alpha\boldsymbol{I}$ 的最小奇异值. 则由特征值的分离理论, 有

$$\sigma_n\leqslant\sigma_n(\boldsymbol{A}-\alpha\boldsymbol{I})+\|\boldsymbol{E}\|_2\leqslant\varepsilon_1+\varepsilon_2. \tag{7.77}$$

将 $\boldsymbol{x}^{(0)}$ 按 $\boldsymbol{u}_1,\boldsymbol{u}_2,\cdots,\boldsymbol{u}_n$ 展开, 有

$$\boldsymbol{x}^{(0)}=\sum_{i=1}^{n}\beta_i\boldsymbol{u}_i,\quad \sum_{i=1}^{n}|\beta_i|^2=\|\boldsymbol{x}^{(0)}\|_2^2=1.$$

从而有

$$\widetilde{\boldsymbol{y}}^{(1)} = (\boldsymbol{A} - \alpha\boldsymbol{I} + \boldsymbol{E})^{-1}\boldsymbol{x}^{(0)}$$
$$= \boldsymbol{V}\boldsymbol{\Sigma}^{-1}\boldsymbol{U}^{\mathrm{H}}\Big(\sum_{i=1}^{n}\beta_i\boldsymbol{u}_i\Big) = \sum_{i=1}^{n}\frac{\beta_i}{\sigma_i}\boldsymbol{v}_i. \tag{7.78}$$

于是有

$$\|\widetilde{\boldsymbol{y}}^{(1)}\|_2 = \left(\sum_{i=1}^{n}\left|\frac{\beta_i}{\sigma_i}\right|^2\right)^{\frac{1}{2}} \geqslant \frac{|\beta_n|}{\sigma_n} \geqslant \frac{|\beta_n|}{\varepsilon_1 + \varepsilon_2}.$$

这样一来, 只要 $|\beta_n|$ 不是很小 (即 $\boldsymbol{x}^{(0)}$ 在 $\boldsymbol{u}_n$ 方向上不是十分亏损), $\|\widetilde{\boldsymbol{y}}^{(1)}\|_2$ 就会很大. 因此, 通常反幂法只需要迭代一次就足够了.

上述分析表明, 利用反幂法求特征向量时, 位移量 α 取为较精确的近似特征值最好. 此时, 一般只需要迭代一次就可以得到很好的近似特征向量. 因此, 通常总是在用某种方法求得 $\boldsymbol{A}$ 的近似特征值之后, 再利用反幂法求对应的特征向量. 连同 QR 方法一起来使用反幂法的基本步骤如下.

算法 7.9

步 1, 输入矩阵 $\boldsymbol{A}=(a_{ij})\in\mathbb{R}^{n\times n}$.

步 2, 上 Hessenberg 化. 用算法 7.4 计算 $\boldsymbol{A}$ 的上 Hessenberg 分解 $\boldsymbol{A}:=\boldsymbol{Q}^{\mathrm{T}}\boldsymbol{A}\boldsymbol{Q}$.

步 3, 用双重步位移隐式 QR 方法 (算法 7.7) 求出 $\boldsymbol{A}$ 的特征值, 而不累积正交变换矩阵.

步 4, 对每个计算得到的近似特征值 $\widetilde{\lambda}$, 在式 (7.66) 中取位移参数 $\alpha=\widetilde{\lambda}$ 进行迭代, 求出特征向量 $\boldsymbol{z}$.

步 5, 计算 $\boldsymbol{x}=\boldsymbol{Q}\boldsymbol{z}$ (则 $\boldsymbol{x}$ 就是对应于 $\widetilde{\lambda}$ 的近似特征向量).

算法 7.9 的 MATLAB 程序如下:

```
function [Lam,V,iter,ki]=ddiqr_eigvec(A,tol)
%用双重步位移隐式QR方法求实方阵的
%全部特征值和相应的特征向量
```

```
%输入:A为n阶实方阵,tol为控制精度(默认是1.e-5)
%输出:Lam为A的全部特征值,V为A的全部特征向量,iter为迭代次数
if nargin<2, tol=1e-5; end
n=size(A,1); x=rand(n,1); %x=ones(n,1);
Lam=zeros(n,1); V=zeros(n);
[A,Q]=mhessen(A);  %调用上Hessenberg化程序
%调用双重步位移隐式QR方法求全部特征值
[iter,lambda]=ddiqr_eig(A,tol);
for i=1:n
    [lam,v,k]=mvpower(A,x,lambda(i));  %调用反幂法程序
    V(:,i)=v; ki(i)=k; Lam(i)=lam;
end
V=Q*V;  %V的每一列为特征向量
```

例 7.9 用算法 7.9 求例 7.7 中矩阵 $\boldsymbol{A}$ 的全部特征值和特征向量.

解 编写脚本 M 文件 ex79.m, 然后在 MATLAB 命令窗口执行之, 得计算结果:

```
>> ex79
双位移QR方法结果      eig函数计算结果
18.4123 + 0.0000i   18.4123 + 0.0000i
11.1805 + 0.0000i   11.1805 + 0.0000i
 1.7099 - 4.2522i    4.4983 + 0.0000i
 1.7099 + 4.2522i    1.7099 + 4.2522i
-2.2327 + 0.0000i    1.7099 - 4.2522i
 4.4983 + 0.0000i   -2.2327 + 0.0000i
-0.2783 + 0.0000i   -0.2783 + 0.0000i
```

```
iter =

     8

ki =

     2     2     2     2     2     2     2

err =

   1.7959e-05
```

§7.5 Givens–Householder 方法

7.3 节中的 Jacobi 方法通过 Givens 变换构造一个正交相似矩阵序列 $\{\boldsymbol{A}_{k+1}\}(\boldsymbol{A}_{k+1}=\boldsymbol{Q}_k^{\mathrm{T}}\boldsymbol{A}_k\boldsymbol{Q}_k)$, 使得 $\boldsymbol{A}_{k+1}$ 趋于一个对角矩阵, 从而得到 $\boldsymbol{A}$ 的全部特征值 $\lambda_i\,(i=1,2,\cdots,n)$. Givens 指出, 如果不要求 $\boldsymbol{A}_k$ 趋于对角矩阵而是三对角矩阵, 则这个过程可以是有限步的. 而 Householder 建议如果 $\boldsymbol{Q}_k$ 取为形如 $\boldsymbol{H}_k=\boldsymbol{I}-2\boldsymbol{u}\boldsymbol{u}^{\mathrm{T}}$ 的 Householder 矩阵, 则可以更有效地实现这个三对角化的过程.

将 $\boldsymbol{A}$ 化为三对角矩阵 $\boldsymbol{T}=\boldsymbol{H}^{\mathrm{T}}\boldsymbol{A}\boldsymbol{H}$ 后, 还需要求 $\boldsymbol{T}$ 的特征值. 为此, Givens 根据 $\boldsymbol{T}-\lambda\boldsymbol{I}$ 的顺序主子式构成的 Sturm 序列这一事实提出了计算 $\boldsymbol{T}$ 的特征值的二分法.

为计算 $\boldsymbol{A}$ 的特征向量, 要先计算 $\boldsymbol{T}$ 的特征向量. 一个有效的方法是反迭代法.

这种先通过 Householder 变换化对称矩阵 $\boldsymbol{A}$ 为对称三对角矩阵 $\boldsymbol{T}$, 然后通过求 $\boldsymbol{T}$ 的特征值和特征向量来求得 $\boldsymbol{A}$ 的特征值和特征向量的方法称为 Givens–Householder 方法. 对称矩阵的三对角化即为上 Hessenberg 化, 这在 节中已经讨论过, 可以执行算法 7.4 得到. 下面只需考虑如何计算对称三对角矩阵 $\boldsymbol{T}$ 的特征值.

§7.5.1 求对称三对角矩阵特征值的二分法

考虑实对称三对角矩阵

$$\boldsymbol{T}=\begin{bmatrix} \alpha_1 & \beta_2 & & & \\ \beta_2 & \alpha_2 & \beta_3 & & \\ & \ddots & \ddots & \ddots & \\ & & \beta_{n-1} & \alpha_{n-1} & \beta_n \\ & & & \beta_n & \alpha_n \end{bmatrix}.$$

假定 $\boldsymbol{T}$ 是不可约的, 即 $\beta_i \neq 0\,(i=2,\cdots,n)$. 否则, 可以将 $\boldsymbol{T}$ 分解成几个阶数更小的不可约三对角矩阵.

令 $p_k(\lambda)$ 表示矩阵 $\boldsymbol{T}-\lambda\boldsymbol{I}$ 的 k 阶顺序主子式, 即

$$p_k(\lambda)=\det(\boldsymbol{T}-\lambda\boldsymbol{I})_k$$

$$=\det\begin{bmatrix}\alpha_1-\lambda & \beta_2 & & & \\ \beta_2 & \alpha_2-\lambda & \beta_3 & & \\ & \ddots & \ddots & \ddots & \\ & & \beta_{k-1} & \alpha_{k-1}-\lambda & \beta_k \\ & & & \beta_k & \alpha_k-\lambda\end{bmatrix},\ k=1,2,\cdots,n.$$

若令 $p_0(\lambda)\equiv 1$, 则有递推式

$$p_0(\lambda)=1,\ p_1(\lambda)=\alpha_1-\lambda,$$

$$p_{k+1}(\lambda)=(\alpha_{k+1}-\lambda)p_k(\lambda)-\beta_{k+1}^2p_{k-1}(\lambda),\quad k=1,\cdots,n-1. \tag{7.79}$$

容易证明, 多项式序列 $\{p_k(\lambda)\}_0^n$ 具有如下四个特征.

引理 7.2 序列 $\{p_k(\lambda)\}_0^n$ 满足

$$p_k(-\infty) > 0, \quad p_k(+\infty) \begin{cases} > 0, & k \text{ 为偶数}, \\ < 0, & k \text{ 为奇数}. \end{cases} \tag{7.80}$$

证明 首先用归纳法证明

$$p_k(\lambda) = (-1)^k \lambda^k + g_{k-1}(\lambda), \tag{7.81}$$

式中: $g_{k-1}(\lambda)$ 为 λ 的 $k-1$ 次多项式.

事实上, 当 $k=0$ 时, 显然式 (7.81) 成立. 假设 $k-1$ 时结论成立. 则对 k, 有

$$\begin{aligned} p_k(\lambda) &= (\alpha_k - \lambda) p_{k-1}(\lambda) - \beta_k^2 p_{k-2}(\lambda) \\ &= (\alpha_k - \lambda)\left[(-1)^{k-1}\lambda^{k-1} + g_{k-2}(\lambda)\right] - \beta_k^2 p_{k-2}(\lambda) \\ &= (-1)^k \lambda^k + (-1)^{k-1}\alpha_k \lambda^{k-1} + (\alpha_k - \lambda) g_{k-2}(\lambda) - \beta_k^2 p_{k-2}(\lambda) \\ &= (-1)^k \lambda^k + g_{k-1}(\lambda). \end{aligned}$$




Back
Close

注意到 k 次多项式 $p_k(\lambda)$ 的符号在 $|\lambda|$ 充分大时决定于其首项的符号, 故由上式立即得到引理的结论. 证毕. □

引理 7.3 $p_0(\lambda)$ 无零点, 相邻两个多项式 $p_k(\lambda)$ 与 $p_{k+1}(\lambda)$ 无公共零点.

证明 $p_0(\lambda)$ 无零点是显然的. 设 $p_k(\lambda)$ 与 $p_{k+1}(\lambda)$ 有公共零点 $\bar{\lambda}$, 则由递推式 (7.79) 可知, $\bar{\lambda}$ 是 p_{k-1} 的零点 (因 $\beta_{k+1} \neq 0$). 依次类推, 可知 $\bar{\lambda}$ 是 $p_{k-2}(\lambda), \cdots, p_1(\lambda), p_0(\lambda)$ 的零点, 与 $p_0(\lambda) = 1$ 矛盾. 证毕. □

引理 7.4 设 $\bar{\lambda}$ 是 $p_k(\lambda)\,(0 < k < n)$ 的零点, 则 $p_{k-1}(\bar{\lambda})p_{k+1}(\bar{\lambda}) < 0$.

证明 因 $\bar{\lambda}$ 是 p_k 的零点, 由引理 7.3 可知, $p_{k-1}(\bar{\lambda}) \neq 0, p_{k+1}(\bar{\lambda}) \neq 0$. 由式 (7.79) 可知

$$p_{k+1}(\bar{\lambda}) = -\beta_{k+1}^2 p_{k-1}(\bar{\lambda}),$$

故 $p_{k-1}(\bar{\lambda})p_{k+1}(\bar{\lambda}) = -\beta_{k+1}^2 p_{k-1}(\bar{\lambda})^2 < 0$. 证毕. □

引理 7.5 $p_k(\lambda)\,(0 < k < n)$ 的零点全是单重的, 并且 $p_k(\lambda)$ 的零点把 $p_{k+1}(\lambda)$ 的零点严格地隔离开来.

证明 对 k 使用归纳法. 事实上, 当 $k = 1$ 时, $p_1(\lambda) = \alpha_1 - \lambda$, 其零点为 α_1. 另外, $p_2(\alpha_1) = -\beta_2^2 < 0$, 且由式 (7.80) 可知, $p_2(-\infty) > 0$, $p_2(+\infty) > 0$. 故 $p_2(\lambda)$ 的根必定在区间 $(-\infty, \alpha_1)$ 与 $(\alpha_1, +\infty)$ 内, 即当 $k = 1$ 时, 引理结论成立.

假设当 $k = i-1$ 时结论成立, 即 $p_{i-1}(\lambda)$ 和 $p_i(\lambda)$ 的根全是单根且 $p_{i-1}(\lambda)$ 的根把 $p_i(\lambda)$ 的根严格隔开. 记 $\lambda_s^{(l)}$ 为 $p_l(\lambda)\,(l = i-1, i)$ 的第 s 个根. 则有

$$\lambda_1^{(i)} < \lambda_1^{(i-1)} < \lambda_2^{(i)} < \lambda_2^{(i-1)} < \cdots < \lambda_{i-1}^{(i)} < \lambda_{i-1}^{(i-1)} < \lambda_i^{(i)}.$$

由于

$$p_{i-1}(-\infty) > 0,\ p_{i-1}(\lambda_1^{(i-1)}) = 0,\ p_{i-1}(\lambda_2^{(i-1)}) = 0, \cdots, p_{i-1}(\lambda_{i-1}^{(i-1)}) = 0,$$

所以

$$p_{i-1}(\lambda_1^{(i)}) > 0,\ \ p_{i-1}(\lambda_2^{(i)}) < 0,\ \ p_{i-1}(\lambda_3^{(i)}) > 0, \cdots,$$

即 $p_{i-1}(\lambda_s^{(i)})$ 的符号为 $(-1)^{s+1}$. 又因

$$p_{i+1}(\lambda_s^{(i)}) = -\beta_{i+1}^2 p_{i-1}(\lambda_s^{(i)}),$$

故

$$p_{i+1}(-\infty) > 0,\ \ p_{i+1}(\lambda_1^{(i)}) < 0,\ \ p_{i+1}(\lambda_2^{(i)}) > 0,\ \ p_{i+1}(\lambda_3^{(i)}) < 0, \cdots.$$

于是由零点存在定理, $p_{i+1}(\lambda)$ 在区间

$$(-\infty, \lambda_1^{(i)}),\ (\lambda_1^{(i)}, \lambda_2^{(i)}), \cdots, (\lambda_{i-1}^{(i)}, \lambda_i^{(i)}),\ (\lambda_i^{(i)}, +\infty)$$

内都有根. 这样的区间共有 $i+1$ 个, 而 $p_{i+1}(\lambda)$ 恰有 $i+1$ 个根, 故在每个区间内有且仅有一个根, 即当 $k=i$ 时结论成立. 由归纳法原理, 引理得证. □

定义 7.5 对于一个多项式序列 $\{p_k(\lambda)\}_0^n$, 如果它具有引理 7.3 至引理 7.5 的性质, 则称这个多项式序列为 Sturm 序列.

对于 Sturm 序列, 任意取定 α, 则

$$p_0(\alpha),\ p_1(\alpha),\ \cdots, p_k(\alpha),\quad k \leqslant n \tag{7.82}$$

是一个有限序列. 用 $s_k(\alpha)$ 表示式 (7.82) 中每相邻两数符号一致的数目, 并称其为该序列的同号数. 若序列中某一项 $p_k(\alpha)=0$, 则约定 $p_k(\alpha)$ 的符号与 $p_{k-1}(\alpha)$ 的符号相同. 根据引理 7.3, 此时必有 $p_{k-1}(\alpha)\neq 0$. 例如, 数列 $\{2,\ 4,\ 8,\ 16,\ -10,\ -12,\ 14\}$ 的同号数 $s_6=4$, 而数列 $\{-1,\ -3,\ -5, 0,\ 6,\ 7\}$ 的同号数 $s_5=4$.




Back
Close

下面的定理是求对称三对角矩阵特征值二分法的理论基础.

定理 7.16 设 $\boldsymbol{T}$ 是不可约对称三对角矩阵, α 是任意实数, 则 $s_k(\alpha)$ 等于 $p_k(\lambda)=0$ 在区间 $[\alpha,+\infty)$ 内根的个数.

证明 用归纳法. 当 $k=0$ 时, 由于 $p_0(\alpha)=1$, 故结论显然成立. 当 $k=1$ 时, $p_1(\lambda)=\alpha_1-\lambda$. 若 $p_1(\alpha)<0$, 则 $s_1(\alpha)=0$. 注意到 $p_1(-\infty)>0$, 可知 $p_1(\lambda)=0$ 的根必在 $(-\infty,\alpha)$ 内, 因此在 $[\alpha,+\infty)$ 无根, 即根的个数为 0 个. 若 $p_1(\alpha)\geqslant 0$, 则 $s_1(\alpha)=1$. 注意到 $p_1(+\infty)<0$, 可知此时 $p_1(\lambda)=0$ 的唯一根必在 $[\alpha,+\infty)$ 内. 故当 $k=1$ 时, 结论为真.

现假定 $k\leqslant l$ 时结论成立. 记 $s_l(\alpha)=r$, $p_l(\lambda)$ 的零点为

$$\xi_l<\xi_{l-1}<\cdots<\xi_1.$$

由归纳法假设, 有

$$\xi_l < \xi_{l-1} < \cdots < \xi_{r+1} < \alpha \leqslant \xi_r < \cdots < \xi_1. \tag{7.83}$$

设 $p_{l+1}(\lambda)$ 的零点为

$$\mu_{l+1} < \mu_l < \cdots < \mu_{r+1} < \mu_r < \cdots < \mu_1,$$

则显然有

$$p_l(\alpha) = \prod_{i=1}^{l}(\xi_i - \alpha), \quad p_{l+1}(\alpha) = \prod_{i=1}^{l+1}(\mu_i - \alpha). \tag{7.84}$$

且据引理 7.5 有 $p_l(\lambda)$ 与 $p_{l+1}(\lambda)$ 的零点相互隔离:

$$\mu_{l+1} < \xi_l < \mu_l < \cdots < \xi_{r+1} < \mu_{r+1} < \xi_r < \mu_r < \cdots < \xi_1 < \mu_1, \tag{7.85}$$

由式 (7.83) 和式 (7.85) 可知, $\xi_{r+1}, \mu_{r+1}, \alpha, \xi_r$ 之间的相互位置只能出现下面四种情况:




(1) $\xi_{r+1} < \alpha < \mu_{r+1}$. 此时 $p_{l+1}(\lambda)$ 在 $[\alpha, +\infty)$ 上恰有 $r+1$ 个零点. 而由式 (7.84) 可知 $p_l(\alpha)$ 的符号为 $(-1)^{l-r}$ 与 $p_{l+1}(\alpha)$ 的符号 $(-1)^{l+1-(r+1)}$ 相同, 故 $s_{l+1}(\alpha) = s_l(\alpha) + 1 = r + 1$.

(2) $\mu_{r+1} < \alpha < \xi_r$. 此时 $p_{l+1}(\lambda)$ 在 $[\alpha, +\infty)$ 上恰有 r 个零点. 而由式 (7.84) 可知 $p_l(\alpha)$ 的符号 $(-1)^{l-r}$ 与 $p_{l+1}(\alpha)$ 的符号 $(-1)^{(l+1)-r}$ 相反, 于是 $s_{l+1}(\alpha) = s_l(\alpha) = r$.

(3) $\mu_{r+1} = \alpha < \xi_r$. 此时 $p_{l+1}(\lambda)$ 在 $[\alpha, +\infty)$ 上恰有 $r+1$ 个零点. 因为此时 $p_{l+1}(\alpha) = 0$, 按约定 $p_{l+1}(\alpha)$ 与 $p_l(\alpha)$ 符号相同, 故仍有 $s_{l+1}(\alpha) = s_l(\alpha) + 1 = r + 1$.

(4) $\mu_{r+1} < \alpha = \xi_r$. 此时 $p_{l+1}(\lambda)$ 在 $[\alpha, +\infty)$ 上恰有 r 个零点. 注意到 $p_l(\alpha) = 0$, 按约定, 它与 $p_{l-1}(\alpha)$ 符号相同. 而由引理 7.4 知, $p_{l+1}(\alpha)$ 与 $p_{l-1}(\alpha)$ 反号, 故 $p_{l+1}(\alpha)$ 与 $p_l(\alpha)$ 反号. 从而 $s_{l+1}(\alpha) = s_l(\alpha) = r$.

这就对 $k=l+1$ 证明了结论成立. 由归纳法原理知定理对一切 $k \leqslant n$ 都成立. □

推论 7.4 设 $\boldsymbol{T}$ 是不可约对称三对角矩阵, $p_k(\lambda)$ 是 $\boldsymbol{T}-\lambda \boldsymbol{I}$ 的前 k 阶主子式, $s_n(\alpha)$ 是序列 $\{p_k(\alpha)\}_0^n$ 的同号数, 则 $s_n(\alpha)$ 等于 $\boldsymbol{T}$ 在区间 $[\alpha,+\infty)$ 中特征值的个数.

利用推论 7.4, 可以确定在任意区间 $(\alpha,\beta]$ 中所含 $\boldsymbol{T}$ 的特征值的数目. 实际上, 它就等于 $s_n(\alpha)-s_n(\beta)$. 而且若 $\boldsymbol{T}$ 的特征值排列为

$$\lambda_n<\lambda_{n-1}<\cdots<\lambda_m<\cdots<\lambda_1,$$

$s_n(\alpha) \geqslant m>s_n(\beta)$, 则 $\lambda_m \in [\alpha,\beta)$.

综上所述, 可以给出计算 $\boldsymbol{T}$ 的特征值 λ_m 的二分法.

算法 7.10 (计算三对角矩阵特征值的二分法)

步 1, 输入包含 λ_m 的初始区间 $[a_0, b_0]$ (如取 $a_0 = -\|\boldsymbol{T}\|, b_0 = \|\boldsymbol{T}\|$). 置 $k := 0$.

步 2, 计算 $[a_k, b_k]$ 的中点 $c_k = \dfrac{a_k + b_k}{2}$ 及 $\{p_i(c_k)\}_{i=0}^{n}$ 的同号数 $s_n(c_k)$.

步 3, 若 $s_n(c_k) \geqslant m$, 则 $a_{k+1} := c_k$, $b_{k+1} := b_k$; 否则, $a_{k+1} := a_k$, $b_{k+1} := c_k$.

步 4, 若 $|b_{k+1} - a_{k+1}| \leqslant \varepsilon$, 则令 $\lambda_m \approx \dfrac{1}{2}\big(a_{k+1} + b_{k+1}\big)$, 停算. 否则, 置 $k := k + 1$, 转步 2.

在算法 7.10 中, λ_m 始终属于区间 $[a_k, b_k]\,(k = 0, 1, \cdots)$. 当 k 充分大时, $[a_k, b_k]$ 的长度 $\dfrac{b_0 - a_0}{2^k}$ 可以小于任意指定的精度, 此时区间中点 c_k 作为 λ_m 的近似值, 其误差不会超过 $\varepsilon/2$. 若二分法的

Back

Close

计算是精确的, 则可以得到任意指定精度的近似特征值. 但实际计算中会有舍入误差的影响. 虽然如此, 利用后验误差分析的方法可以证明二分法是一个数值稳定的方法.

注 7.4 在算法 7.10 中, 每一步都要计算 n 个多项式 $p_i(\lambda)\,(i = 1, 2, \cdots, n)$ 在区间 $[a_k, b_k]$ 的中点 c_k 的值 (因 $p_0(\lambda) = 1$, 故不需要计算), 这很容易发生“上溢”或“下溢”现象.

下面讨论 $s = s_n(\alpha)$ 的计算. 由于

$$p_k(\alpha) = (\alpha_k - \alpha)p_{k-1}(\alpha) - \beta_k^2 p_{k-2}(\alpha),\ \ k = 1, 2, \cdots, n,$$

这里 $p_0(\alpha) = 1$, $\beta_1 = 0$, $p_{-1}(\alpha) = 0$. 令 $s = 0$. 若 $p_{k-1}(\alpha)p_k(\alpha) > 0$, s 加 1. 若 $p_{k-1}(\alpha)p_k(\alpha) < 0$, s 不变. 若 $p_{k-1}(\alpha)p_k(\alpha) = 0$, 由于 $p_{k-1}(\alpha)$ 与 $p_k(\alpha)$ 不能同时为 0, 分为两种情况. 第一种情况是 $p_{k-1}(\alpha) = 0$, $p_k(\alpha) \neq 0$, 此时 $p_{k-1}(\alpha)$ 与 $p_{k-2}(\alpha)\,(\neq 0)$ 同

号. 若 $p_{k-2}(\alpha)p_k(\alpha) > 0$, 则 s 加 1, 否则 s 不变. 第二种情况是 $p_{k-1}(\alpha) \neq 0$, $p_k(\alpha) = 0$. 此时 $p_k(\alpha)$ 与 $p_{k-1}(\alpha)\,(\neq 0)$ 同号, s 加 1.

因此, 如果引入

$$q_k(\alpha) = p_k(\alpha)/p_{k-1}(\alpha), \quad k = 1, 2, \cdots, n,$$

则

$$q_k(\alpha) = (\alpha_k - \alpha) - \beta_k^2/q_{k-1}(\alpha), \quad k = 1, 2, \cdots, n. \tag{7.86}$$

对于任意的 $1 \leqslant k \leqslant n$, ① 若 $q_{k-1}(\alpha) \neq 0$, 当式 (7.86) 的计算结果 $q_k(\alpha) \geqslant 0$ 时, s 加 1, 否则, s 不变; ② 若 $q_{k-1}(\alpha) = 0$, 这表明 $p_{k-1}(\alpha) = 0$, 从而有 $p_k(\alpha) = -\beta_k^2 p_{k-2}(\alpha)$. 由于 $\beta_k \neq 0$, $p_k(\alpha)$ 与 $p_{k-2}(\alpha)$ 异号, 故 $p_k(\alpha)$ 与 $p_{k-1}(\alpha)$ 同号, s 加 1.

为此, 定义一个新的序列 $\{q_k(\lambda)\}_{k=1}^n$ 如下:

$$q_1(\lambda) = \alpha_1 - \lambda,$$

$$q_k(\lambda) = \begin{cases} \alpha_k - \lambda - \dfrac{\beta_k^2}{q_{k-1}(\lambda)}, & q_{k-1}(\lambda) \neq 0, \\ 1, & q_{k-1}(\lambda) = 0, \end{cases}$$

$$k = 2, 3, \cdots, n.$$

则 $\{p_k(\lambda)\}_{k=0}^n$ 的同号数 $s_n(\lambda)$ 等于 $\{q_k(\lambda)\}_{k=1}^n$ 中非负数的数目. 故在二分法的实际计算中, 总是通过计算 $\{q_k(\lambda)\}_{k=1}^n$ 的非负数的数目来代替计算 $s_n(\lambda)$.

§7.5.2 二分法的程序实现

本节考虑算法 7.10 的程序实现. 首先考虑多项式序列 $\{p_k(\lambda)\}_{k=1}^n$ 和 $\{q_k(\lambda)\}_{k=1}^n$. 编制 MATLAB 程序如下:

```
%本函数计算Sturm序列{p_k(t)}的值
function p=pkfun(t,a,b)
n=length(a);
p(1)=a(1)-t; p(2)=(a(2)-t)*p(1)-b(1);
for k=3:n
    p(k)=(a(k)-t)*p(k-1)-b(k-1)^2*p(k-2);
end
%本函数计算防止溢出的多项式序列{q_k(t)}的值
function q=qkfun(t,a,b)
n=length(a);
```

```
q(1)=a(1)-t;
for k=2:n
    if q(k-1)==0
        q(k)=1;
    else
        q(k)=a(k)-t-b(k-1)^2/q(k-1);
    end
end
```

用数据

$$a = [0.8147, 0.9058, 0.1270, 0.9134, 0.6324, 0.0975, 0.2785, 0.5469]^{\mathrm{T}},$$

$$b = [0.9575, 0.9649, 0.1576, 0.9706, 0.9572, 0.4854, 0.8003]^{\mathrm{T}}, \quad t = 6.8,$$

测试两个程序, 得

```
>> a=[0.8147,0.9058,0.1270,0.9134,0.6324,0.0975,0.2785,0.5469]';
>> b=[0.9575,0.9649,0.1576,0.9706,0.9572,0.4854,0.8003]';
>> p=pkfun(6.8,a,b)
p =
    1.0e+06 *
   -0.0000   0.0000 -0.0002   0.0013 -0.0079   0.0517 -0.3355   2.0646
>> q=qkfun(6.8,a,b)
q =
   -5.9853 -5.7410 -6.5108 -5.8828 -6.0075 -6.5500 -6.4855 -6.1543
```

可以看出序列 $\{q_k(\lambda)\}_{k=1}^{n}$ 的计算的确能有效地防止“溢出”.

现在结合算法 7.10 和注 7.4, 编制 MATLAB 程序如下:

```
%Givens-Householder方法程序-givens_househ.m
function [lambda,k]=givens_househ(T,m,tol,max_it)
%用Givens-Householder求实对称三对角矩阵的第m个特征值
%输入:T为n阶实对称方阵,m特征值序号,
%      tol为容许误差,max_it为最大迭代次数
```

```
%输出:lambda为第m个特征值,k为满足精度迭代次数
if nargin<4, max_it=500; end
if nargin<3, tol=1e-6; end
n=size(T,1); k=0;
b=norm(T,inf); a=-b;
alpha=diag(T); beta=diag(T,1);
while(k<max_it)
    k=k+1; c=(a+b)/2;
    q=qkfun(c,alpha,beta);
    tt=find(q>=0);
    sn=length(tt);
    if(sn>=m), a=c; else, b=c; end
    if abs(b-a)<=tol,
```

```
            lambda=c; break;
        end
    end
```

下面看一个计算实例.

例 7.10 用 Givens–Householder 方法计算矩阵

$$\boldsymbol{A}=\begin{bmatrix} 1 & 1 & 1 & \cdots & 1 \\ 1 & 2 & 2 & \cdots & 2 \\ 1 & 2 & 3 & \cdots & 3 \\ \vdots & \vdots & \vdots & \ddots & \vdots \\ 1 & 2 & 3 & \cdots & n \end{bmatrix}$$

的全部特征值. 取 $n=12$.

解 编制 MATLAB 脚本文件 ex710.m, 然后在命令窗口执行之, 即可得到 $\boldsymbol{A}$ 全部特征值以及计算时间.




§7.5.3 特征向量的计算

关于 $\boldsymbol{A}$ 的特征向量的计算, 既可以直接从 $\boldsymbol{A}$ 出发借助已求得的近似特征值利用反迭代法计算, 也可以先用反迭代法求出 $\boldsymbol{T}$ 的相应特征向量, 然后再利用三对角化 (上 Hessenberg 化) 过程所得到的变换矩阵 $\boldsymbol{H}_1, \boldsymbol{H}_2, \cdots, \boldsymbol{H}_{n-2}$ 将其变换为 $\boldsymbol{A}$ 的特征向量. 后者由于其方法简单、计算量小而得到推荐.

设 $(\lambda, \boldsymbol{x})$ 是矩阵 $\boldsymbol{T}$ 的一个特征对, 即 $\boldsymbol{T}\boldsymbol{x} = \lambda\boldsymbol{x}$. 由于

$$\boldsymbol{T} = (\boldsymbol{H}_1\boldsymbol{H}_2\cdots\boldsymbol{H}_{n-2})^{\mathrm{T}}\boldsymbol{A}(\boldsymbol{H}_1\boldsymbol{H}_2\cdots\boldsymbol{H}_{n-2}),$$

故

$$\boldsymbol{A}(\boldsymbol{H}_1\boldsymbol{H}_2\cdots\boldsymbol{H}_{n-2})\boldsymbol{x} = \lambda(\boldsymbol{H}_1\boldsymbol{H}_2\cdots\boldsymbol{H}_{n-2})\boldsymbol{x},$$

于是

$$\boldsymbol{z} = (\boldsymbol{H}_1\boldsymbol{H}_2\cdots\boldsymbol{H}_{n-2})\boldsymbol{x} \tag{7.87}$$

即为 $\boldsymbol{A}$ 的相应于 λ 的特征向量.

实际计算时不是直接按式 (7.87) 进行 $n-2$ 个矩阵乘向量的运算, 而是利用 Householder 矩阵 $\boldsymbol{H}_k = \boldsymbol{I} - 2\boldsymbol{u}_k\boldsymbol{u}_k^{\mathrm{T}}$ 的特殊形式按下列公式计算:

$$\begin{cases} \boldsymbol{y}_{n-1} = \boldsymbol{x}, \\ \boldsymbol{y}_k = \boldsymbol{H}_k\boldsymbol{y}_{k+1} = \boldsymbol{y}_{k+1} - 2(\boldsymbol{u}_k^{\mathrm{T}}\boldsymbol{y}_{k+1})\boldsymbol{u}_k, \\ \boldsymbol{z} = \boldsymbol{y}_1. \end{cases}$$

因此, 问题归结为求不可约对称三对角矩阵 $\boldsymbol{T}$ 的特征向量 $\boldsymbol{x}$. 通常采用所谓的*反迭代法*. 反迭代法本质上是用原点位移的反幂法求矩阵的特征向量, 其基本迭代格式如下:

$$\begin{cases} (\boldsymbol{T} - \mu\boldsymbol{I})\boldsymbol{v}_k = \boldsymbol{z}_{k-1}, \\ \boldsymbol{z}_k = \boldsymbol{v}_k/\|\boldsymbol{v}_k\|_2, \quad k = 1, 2, \cdots, \end{cases} \tag{7.88}$$

式中: μ 为事先取定的位移参数; $\boldsymbol{z}_0$ 为初始向量.

通常用 Givens–Householder 方法 (或其他方法, 如 QR 方法等) 求得某个特征值的近似值后, 即用其作为位移参数, 然后再用式 (7.88) 进行迭代, 会有很快的收敛速度. 反迭代法的基本步骤如下.

算法 7.11 (计算特征向量的反迭代法)

步 1, 计算 $\boldsymbol{A}$ 的上 Hessenberg 分解: $\boldsymbol{Q}^{\mathrm{T}}\boldsymbol{A}\boldsymbol{Q}=\boldsymbol{T}$.

步 2, 使用算法 7.10 计算 $\boldsymbol{T}$ 的近似特征值 $\bar{\lambda}_1>\bar{\lambda}_2>\cdots>\bar{\lambda}_n$.

步 3, 对每个计算得到的特征值 $\bar{\lambda}_i\,(i=1,2,\cdots,n)$, 在式 (7.88) 中取 $\mu:=\bar{\lambda}_i$ 进行迭代, 求出特征向量 $\boldsymbol{z}_i$.

步 4, 计算 $\boldsymbol{x}_i=\boldsymbol{Q}\boldsymbol{z}_i$ (则 $\boldsymbol{x}_i$ 就是对应于 $\bar{\lambda}_i$ 的近似特征向量).




Givens–Householder 方法可以用来计算对称矩阵的全部特征值问题. 但它可以灵活地求解各种部分特征值问题, 如求若干最大 (最小) 特征值, 求位于给定区间 $[\alpha, \beta]$ 上的特征值等. 这一点它比其他部分特征值问题的算法显得更为优越, 所以这一方法通常也用来计算对称矩阵的部分特征值问题.

算法 7.11 的 MATLAB 程序如下:

```
%基于Givens-Householder方法的求特征向量的
%反迭代法程序-ghvector.m
function  [Lam,V,ki]=ghvector(A,tol,max_it)
%用反迭代法求实对称矩阵A的第m大的特征向量
%输入:A为n阶对称方阵,m为按降幂排列特征值序号,
%      tol为容许误差,max_it为最大迭代次数
%输出:lambda为第m大的特征值,x为对应的特征向量,
```

```
%      k为迭代次数
n=size(A,1); Lam=zeros(n,1); x=rand(n,1); %x=ones(n,1);
[T,Q]=mhessen(A);  %调用上Hessenberg化程序
for i=1:n
     [la]=givens_househ(T,i,tol,max_it); Lam(i)=la;
     [lam,v,k2]=mvpower(T,x,Lam(i));  %调用反幂法程序
     Lam(i)=lam; V(:,i)=v; ki(i)=k2;
     norm(T*v-Lam(i)*v),
end
V=Q*V; %V的每一列为特征向量
```

例 7.11 用算法 7.11 计算例 7.11 中矩阵 $\boldsymbol{A}$ 的全部特征值和特征向量. 取 $n = 15$.

解　编制 MATLAB 脚本文件 ex711.m, 然后在命令窗口执行之, 即可得到 $\boldsymbol{A}$ 全部特征值和特征向量以及计算时间.

§7.6 Krylov 子空间方法

由于目前计算机存储空间和运算速度的限制, QR 方法以及由其导出的各类方法只适用于小型矩阵特征值和特征向量的计算, 而并不能用于求解大型矩阵的特征值问题. 对于大型稀疏矩阵的特征值问题而言, 目前较为有效和实用的一种求解方法就是 Krylov 子空间方法. 特别是近几年, 随着重开始技术的不断完善, 使得这类方法的效率越来越高, 适用范围越来越广. 本节介绍与 Krylov 子空间方法有关的一些基本概念和重要理论, 并且详细地阐述目前较为成熟的 Arnoldi 方法和 Lanczos 方法.

设 $\boldsymbol{A} \in \mathbb{R}^{n\times n}$ 是一个大型稀疏的矩阵, 即 $n \gg 1$ (如 $n \geqslant 10^6$), 而且 $\boldsymbol{A}$ 的非零元很少 (如不超过 10%). 对于这样矩阵的特征值问题, 一般来讲用 QR 方法来求解已经是不可能的事情. 例如, 假如希望用 QR 方法来计算 $\boldsymbol{A}$ 的特征值, 则完成这一计算任务所需




Back
Close

的运算量约为 $10n^3$. 如果 $n = 10^7$, 而且假定用一个万亿次的计算机来执行这一计算任务, 则所需的计算时间为

$$\frac{10 \times 10^{21}}{10^{12} \times 3600 \times 24 \times 365} \approx 317 \text{ (年)}$$

此外, QR 方法所需的存储量是 $O(n^2) = O(10^{14}) = O(15\text{G})$, 这也是现在的计算机上无法实现的事情.

那么如何来求解大型的矩阵计算问题呢? 一个直观而朴素的想法是: 先用一个小型的问题“逼近”所考虑的大型问题, 然后用这个小型问题的解近似所求的解. 实现这一想法的一个途径是, 先选择一个适当的 k 维子空间 $\mathcal{X}_k \subset \mathbb{R}^n, k \ll n$, 然后将所考虑的问题在某种特定的意义下“投影”到 $\mathcal{X}_k$ 上变成一个小型的问题.

目前, 常用的子空间 $\mathcal{X}_k$ 是 Krylov 子空间:

$$\mathcal{K}_k(\boldsymbol{A}, \boldsymbol{v}) = \text{span}\{\boldsymbol{v}, \boldsymbol{A}\boldsymbol{v}, \cdots, \boldsymbol{A}^{k-1}\boldsymbol{v}\},$$

式中: $\boldsymbol{v} \in \mathbb{R}^n$ 是人为给定的.

选择这一空间是因为它有一些特有的优点:

(1) 所需信息量少: 只需一个矩阵 $\boldsymbol{A}$, 一个向量 $\boldsymbol{v}$ 和一个正整数 k.

(2) 计算简单: 只涉及矩阵乘向量, 可以充分利用 $\boldsymbol{A}$ 所具有的稀疏性.

(3) 容易扩张: $\mathcal{K}_{k+1}(\boldsymbol{A}, \boldsymbol{v}) = \mathcal{K}_k(\boldsymbol{A}, \boldsymbol{v}) + \mathrm{span}\{\boldsymbol{A}^k \boldsymbol{v}\}$.

当所有的子空间确定之后, 下一步的任务就是如何将所考虑的大型问题 "投影" 到 $\mathcal{X}_k$ 上. 设 $\boldsymbol{V}_k \in \mathbb{R}^{n\times k}$ 是 $\mathcal{X}_k$ 上的一组标准正交基构成的矩阵, 即若记 $\boldsymbol{V}_k = [\boldsymbol{v}_1, \boldsymbol{v}_2, \cdots, \boldsymbol{v}_k]$, 则有

$$\mathcal{X}_k = \mathrm{span}\{\boldsymbol{v}_1, \boldsymbol{v}_2, \cdots, \boldsymbol{v}_k\}, \quad \boldsymbol{V}_k^{\mathrm{T}} \boldsymbol{V}_k = \boldsymbol{I}. \tag{7.89}$$

有了 $\boldsymbol{V}_k$ 之后, 就可以计算 $\boldsymbol{A}$ 在 $\mathcal{X}_k$ 上的投影

$$\boldsymbol{H}_k \;=\; \boldsymbol{V}_k^{\mathrm{T}} \boldsymbol{A} \boldsymbol{V}_k.$$

现在来看 $\boldsymbol{H}_k$ 的几何解释. 令 $\mathcal{A} : \mathbb{R}^n \mapsto \mathbb{R}^n$ 为

$$\boldsymbol{x} \mapsto \boldsymbol{A}\boldsymbol{x}, \quad \forall\, \boldsymbol{x} \in \mathbb{R}^n.$$

则有

$$\begin{array}{ccc} \mathcal{X}_k & \xrightarrow{\mathcal{A}|_{\mathcal{X}_k}} & \mathbb{R}^n \\ & \searrow & \Big\downarrow \mathcal{P}_{\mathcal{X}_k} \\ & \mathcal{P}_{\mathcal{X}_k} \circ \mathcal{A}|_{\mathcal{X}_k} & \mathcal{X}_k \end{array} .$$

而且 $\boldsymbol{H}_k$ 正好是线性算子 $\mathcal{P}_{\mathcal{X}_k} \circ \mathcal{A}|_{\mathcal{X}_k}$ 在基向量 $\{\boldsymbol{v}_1, \boldsymbol{v}_2, \cdots, \boldsymbol{v}_k\}$ 下的矩阵表示, 其中 $\mathcal{A}|_{\mathcal{X}_k}$ 表示算子 $\mathcal{A}$ 在子空间 $\mathcal{X}_k$ 上的限制, $\mathcal{P}_{\mathcal{X}_k}$ 表示 $\mathcal{X}_k$ 上的正交投影算子, $\mathcal{P}_{\mathcal{X}_k} \circ \mathcal{A}|_{\mathcal{X}_k}$ 表示算子 $\mathcal{P}_{\mathcal{X}_k}$ 和 $\mathcal{A}|_{\mathcal{X}_k}$ 的复合.

事实上, 只要注意到: ① $\boldsymbol{x} \in \mathcal{X}_k$ 当且仅当存在 $\boldsymbol{y} \in \mathbb{R}^k$ 使得 $\boldsymbol{x} = \boldsymbol{V}_k \boldsymbol{y}$; ② $\mathcal{P}_{\mathcal{X}_k}$ 的矩阵表示为 $\boldsymbol{V}_k \boldsymbol{V}_k^{\mathrm{T}}$. 有

$$\mathcal{P}_{\mathcal{X}_k} \circ \mathcal{A}|_{\mathcal{X}_k}(\boldsymbol{V}_k \boldsymbol{y}) = \boldsymbol{V}_k \boldsymbol{V}_k^{\mathrm{T}} \boldsymbol{A} \boldsymbol{V}_k \boldsymbol{y}, \quad \forall\, \boldsymbol{y} \in \mathbb{R}^k,$$

即

$$\mathcal{P}_{\mathcal{X}_k} \circ \mathcal{A}|_{\mathcal{X}_k} \boldsymbol{V}_k = \boldsymbol{V}_k \boldsymbol{H}_k.$$

这表明, $\boldsymbol{H}_k$ 正好是线性算子 $\mathcal{A}|_{\mathcal{X}_k}$ 在基向量 $\{\boldsymbol{v}_1, \boldsymbol{v}_2, \cdots, \boldsymbol{v}_k\}$ 下的矩阵表示.

只要有了 $\boldsymbol{H}_k$ 和 $\boldsymbol{V}_k$, 涉及大矩阵 $\boldsymbol{A}$ 的计算问题就可用相应的小矩阵 $\boldsymbol{H}_k$ 对应的问题来解决.

设 $(\lambda, \boldsymbol{x})$ 是矩阵 $\boldsymbol{A}$ 的一个特征对, 即

$$\boldsymbol{A}\boldsymbol{x} = \lambda \boldsymbol{x}. \tag{7.90}$$




Back
Close

再假定

$$\boldsymbol{x} = \boldsymbol{V}_k \boldsymbol{y} + \boldsymbol{v} \approx \boldsymbol{V}_k \boldsymbol{y}.$$

于是, 由式 (7.90), 有

$$\boldsymbol{A}\boldsymbol{V}_k \boldsymbol{y} \approx \lambda \boldsymbol{V}_k \boldsymbol{y}.$$

从而有

$$\boldsymbol{H}_k \boldsymbol{y} = \boldsymbol{V}_k^{\mathrm{T}} \boldsymbol{A} \boldsymbol{V}_k \boldsymbol{y} \approx \lambda \boldsymbol{y}.$$

因此, 若 $(\lambda, \boldsymbol{y})$ 是 $\boldsymbol{H}_k = \boldsymbol{V}_k^{\mathrm{T}} \boldsymbol{A} \boldsymbol{V}_k$ 的特征对, 则可用 $(\lambda, \boldsymbol{V}_k \boldsymbol{y})$ 近似 $\boldsymbol{A}$ 的特征对. 这样就将大型特征值问题 (7.90) 转化为一个小型特征值问题:

$$\boldsymbol{H}_k \boldsymbol{y} = \lambda \boldsymbol{y}.$$

这就是用 Krylov 子空间方法求解大规模特征值问题的基本思想.

§7.6.1 Rayleigh–Ritz 投影方法

基于 Krylov 子空间来设计特征值问题数值解法的最基本的技术就是投影. 下面介绍 Rayleigh–Ritz 投影方法及其基本性质.

设 $\boldsymbol{A} \in \mathbb{R}^{n\times n}, \mathcal{V} \subset \mathbb{C}^n$ 是一个 m 维子空间 (不一定是 Krylov 子空间), 借助 $\mathcal{V}$ 给出 $\boldsymbol{A}$ 的某些近似特征对. 达到这一目的的一种途径就是正交投影法, 其基本思想是选择 $\mu \in \mathbb{C}$ 和 $\boldsymbol{u} \in \mathcal{V}$ 使得

$$(\boldsymbol{A}\boldsymbol{u} - \mu\boldsymbol{u}) \perp \mathcal{V}. \tag{7.91}$$

通常称条件 (7.91) 为 Galerkin 条件. 这一条件可以等价地表述为

$$(\boldsymbol{A}\boldsymbol{u} - \mu\boldsymbol{u}, \boldsymbol{v}) = 0, \quad \forall\, \boldsymbol{v} \in \mathcal{V}. \tag{7.92}$$

设 $\mathcal{V}$ 的一组标准正交基为 $\boldsymbol{v}_1, \boldsymbol{v}_2, \cdots, \boldsymbol{v}_m$, 并令 $\boldsymbol{V}_m = [\boldsymbol{v}_1, \boldsymbol{v}_2, \cdots, \boldsymbol{v}_m]$, 则 $\boldsymbol{u}$ 可表示为 $\boldsymbol{u} = \boldsymbol{V}_m\boldsymbol{y}$, 其中 $\boldsymbol{y} \in \mathbb{C}^m$. 这样式 (7.92) 可以等价地表示为

$$\boldsymbol{V}_m^{\mathrm{H}}(\boldsymbol{A}\boldsymbol{V}_m\boldsymbol{y} - \mu\boldsymbol{V}_m\boldsymbol{y}) = \boldsymbol{0},$$

即

$$\boldsymbol{A}_m\boldsymbol{y} = \mu\boldsymbol{y}, \tag{7.93}$$

其中 $\boldsymbol{A}_m = \boldsymbol{V}_m^{\mathrm{H}}\boldsymbol{A}\boldsymbol{V}_m$ 正好是 $\boldsymbol{A}$ 关于 $\boldsymbol{V}_m$ 的 Rayleigh 商. 这里, 对 $\boldsymbol{A}_m$ 的每一个特征对 $(\mu, \boldsymbol{y})$, 称 μ 为 Ritz 值, 而称 $\boldsymbol{u} = \boldsymbol{V}_m\boldsymbol{y}$ 为 Ritz 向量, 称 $(\mu, \boldsymbol{V}_m\boldsymbol{y})$ 为 Ritz 对. 在一定条件下, Ritz 对 $(\mu, \boldsymbol{V}_m\boldsymbol{y})$ 将是 $\boldsymbol{A}$ 的一个很好的近似特征对.

上述的这一求 $\boldsymbol{A}$ 的某些近似特征值和近似特征向量的方法就称为 Rayleigh–Ritz 投影方法, 其基本步骤如下.

算法 7.12 (Rayleigh–Ritz 投影方法)

步 1, 求 $\mathcal{V}$ 的一组标准正交基 $\{\boldsymbol{v}_1, \boldsymbol{v}_2, \cdots, \boldsymbol{v}_m\}$, 并记 $\boldsymbol{V}_m = [\boldsymbol{v}_1, \boldsymbol{v}_2, \cdots, \boldsymbol{v}_m]$.

步 2, 计算 Rayleigh 商 $\boldsymbol{A}_m = \boldsymbol{V}_m^{\mathrm{H}}\boldsymbol{A}\boldsymbol{V}_m$.

步 3, 计算 $\boldsymbol{A}_m$ 的特征值, 并选择其中若干所需的作为 $\boldsymbol{A}$ 的近似值: $\mu_1, \mu_2, \cdots, \mu_k$.

步 4, 计算 μ_i 所对应的特征向量 $\boldsymbol{y}_i$, 并形成 Ritz 向量 $\boldsymbol{u}_i = \boldsymbol{V}_m \boldsymbol{y}_i, i = 1, 2, \cdots, k$.

一般来讲, 有 $m \ll n$. 因此, $\boldsymbol{A}_m$ 将是一个阶数很小的矩阵. 这样, 就有很多成熟的数值方法用来完成上述方法中步 3 和步 4 的计算任务. 例如, 可先用 QR 方法来计算 $\boldsymbol{A}_m$ 特征值, 然后再用反幂法来求其对应的特征向量. 另外需要说明的一点是, 由于特征向量比 Schur 向量对扰动敏感得多, 因此, 在某些情况下, 步 4 中的“求 Ritz 向量”可以用“求 Schur 向量”来代替 (所谓 Schur 向量是指实矩阵 $\boldsymbol{A} \in \mathbb{R}^{n \times n}$ 的实 Schur 分解 $\boldsymbol{A} = \boldsymbol{U}^{\mathrm{T}} \boldsymbol{T} \boldsymbol{U}$ 中正交矩阵 $\boldsymbol{U}$ 的列向量, 其中 $\boldsymbol{T}$ 为对角块为 1×1 或 2×2 子矩阵的拟上三角矩阵).

下面给出用 Ritz 对来近似特征对的一个理论依据. 这里假定 $\boldsymbol{A} \in \mathbb{R}^{n\times n}$ 且 $\boldsymbol{A}^{\mathrm{T}} = \boldsymbol{A}$, 即 $\boldsymbol{A}$ 是 n 阶实对称矩阵.

设 $\boldsymbol{V} = [\boldsymbol{V}_k, \boldsymbol{V}_u]$ 是一个 n 阶正交矩阵, 其中 $\boldsymbol{V}_k \in \mathbb{R}^{n\times k}$, $\boldsymbol{V}_u \in \mathbb{R}^{n\times(n-k)}$. 在实际应用中, 通常 $\boldsymbol{V}_k$ 是已知的, 是由 Lanczos 方法产生的 Krylov 子空间的一组标准正交基构成的, 而 $\boldsymbol{V}_u$ 通常是不知道的. 在下面的讨论中, 将使用如下的符号:

$$\boldsymbol{T} = \boldsymbol{V}^{\mathrm{T}}\boldsymbol{A}\boldsymbol{V} = \begin{bmatrix} \boldsymbol{V}_k^{\mathrm{T}}\boldsymbol{A}\boldsymbol{V}_k & \boldsymbol{V}_k^{\mathrm{T}}\boldsymbol{A}\boldsymbol{V}_u \\ \boldsymbol{V}_u^{\mathrm{T}}\boldsymbol{A}\boldsymbol{V}_k & \boldsymbol{V}_u^{\mathrm{T}}\boldsymbol{A}\boldsymbol{V}_u \end{bmatrix} := \begin{bmatrix} \boldsymbol{T}_k & \boldsymbol{T}_{uk} \\ \boldsymbol{T}_{ku} & \boldsymbol{T}_u \end{bmatrix}. \quad (7.94)$$

Ritz 对作为 $\boldsymbol{A}$ 的特征对的“最佳”逼近有多种理由. 其中之一可以从上面的矩阵看出, 由于 $\boldsymbol{T}_{uk}$ 和 $\boldsymbol{T}_u$ 是不知道的, 而 $\boldsymbol{T}_k$ 是唯一知道的, 因此自然只能借助 $\boldsymbol{T}_k$ 的特征对来构造出 $\boldsymbol{A}$ 的近似特征对. 另一个理由是下面的定理 7.17 所描述的 Rayleigh 商的最佳逼近性.

如所周知, 若对某个 k 阶矩阵 $\boldsymbol{R}$ 有 $\boldsymbol{A}\boldsymbol{V}_k = \boldsymbol{V}_k\boldsymbol{R}$, 则 $\mathcal{R}(\boldsymbol{V}_k)$ 就是 $\boldsymbol{A}$ 的一个不变子空间, 从而 $\boldsymbol{R}$ 的特征值全都是 $\boldsymbol{A}$ 的特征值, 即 $\lambda(\boldsymbol{R}) \subset \lambda(\boldsymbol{A})$. 如果上述等式不能成立, 自然希望找到一个 k 阶矩阵 $\boldsymbol{R}$, 使得

$$\|\boldsymbol{A}\boldsymbol{V}_k - \boldsymbol{V}_k\boldsymbol{R}\|_2 = \min,$$

然后用 $\boldsymbol{R}$ 的特征值来近似 $\boldsymbol{A}$ 的特征值. 下面的定理表明 Rayleigh 商 $\boldsymbol{T}_k = \boldsymbol{V}_k^{\mathrm{T}}\boldsymbol{A}\boldsymbol{V}_k$ 就是满足此条件的矩阵.

定理 7.17 设 $\boldsymbol{A} \in \mathbb{R}^{n\times n}$ 满足 $\boldsymbol{A}^{\mathrm{T}} = \boldsymbol{A}$, $\boldsymbol{V}_k \in \mathbb{R}^{n\times k}$ 满足 $\boldsymbol{V}_k^{\mathrm{T}}\boldsymbol{V}_k = \boldsymbol{I}$, 则有

(1) $\boldsymbol{T}_k$ 满足

$$\begin{aligned}\|\boldsymbol{A}\boldsymbol{V}_k - \boldsymbol{V}_k\boldsymbol{T}_k\|_2 &= \min\{\|\boldsymbol{A}\boldsymbol{V}_k - \boldsymbol{V}_k\boldsymbol{S}\|_2 : \boldsymbol{S}^{\mathrm{T}} = \boldsymbol{S} \in \mathbb{R}^{k\times k}\} \\ &= \|\boldsymbol{T}_{ku}\|_2. \end{aligned} \tag{7.95}$$




(2) 若 $\boldsymbol{T}_k$ 的谱分解为 $\boldsymbol{T}_k = \boldsymbol{Y}\boldsymbol{M}\boldsymbol{Y}^{\mathrm{T}}$, 其中 $\boldsymbol{Y}$ 是正交矩阵, $\boldsymbol{M}$ 是对角矩阵, 则

$$\|\boldsymbol{A}(\boldsymbol{V}_k\boldsymbol{Y}) - (\boldsymbol{V}_k\boldsymbol{Y})\boldsymbol{M}\|_2 = \min\{\|\boldsymbol{A}\boldsymbol{P}_k - \boldsymbol{P}_k\boldsymbol{D}\|_2 : \boldsymbol{P}_k \in \mathcal{P},\ \boldsymbol{D} \in \mathcal{D}\}$$

$$= \|\boldsymbol{T}_{ku}\|_2, \tag{7.96}$$

式中:

$$\mathcal{P} = \{\boldsymbol{P}_k \in \mathbb{R}^{n\times k} : \boldsymbol{P}_k^{\mathrm{T}}\boldsymbol{P}_k = \boldsymbol{I},\ \ \mathcal{R}(\boldsymbol{P}_k) = \mathcal{R}(\boldsymbol{V}_k)\},$$

$$\mathcal{D} = \{\boldsymbol{D} \in \mathbb{R}^{k\times k} : \boldsymbol{D} \text{ 是对角矩阵}\}.$$

证明 (1) 对任意的 $\boldsymbol{S}^{\mathrm{T}}=\boldsymbol{S}\in\mathbb{R}^{k\times k}$, 有

$$\begin{aligned}\|\boldsymbol{AV}_k-\boldsymbol{V}_k\boldsymbol{S}\|_2&=\|\boldsymbol{V}^{\mathrm{T}}(\boldsymbol{AV}_k-\boldsymbol{V}_k\boldsymbol{S})\|_2\\&=\|[\boldsymbol{V}_k,\boldsymbol{V}_u]^{\mathrm{T}}(\boldsymbol{AV}_k-\boldsymbol{V}_k\boldsymbol{S})\|_2\\&=\left\|\begin{bmatrix}\boldsymbol{V}_k^{\mathrm{T}}\boldsymbol{AV}_k-\boldsymbol{V}_k^{\mathrm{T}}\boldsymbol{V}_k\boldsymbol{S}\\\boldsymbol{V}_u^{\mathrm{T}}\boldsymbol{AV}_k-\boldsymbol{V}_u^{\mathrm{T}}\boldsymbol{V}_k\boldsymbol{S}\end{bmatrix}\right\|_2=\left\|\begin{bmatrix}\boldsymbol{T}_k-\boldsymbol{S}\\\boldsymbol{T}_{ku}\end{bmatrix}\right\|_2\\&\geqslant\|\boldsymbol{T}_{ku}\|_2,\end{aligned}$$

而且当 $\boldsymbol{S}=\boldsymbol{T}_k$ 时上述不等式等号成立. 因此, 定理的结论 (1) 成立.

(2) 任意取 $\boldsymbol{P}_k\in\mathcal{P}$ 和 $\boldsymbol{D}\in\mathcal{D}$, 由 $\mathcal{P}$ 的定义可知, 必存在一个 k 阶正交矩阵 $\boldsymbol{U}$ 使得 $\boldsymbol{P}_k=\boldsymbol{V}_k\boldsymbol{U}$. 再由结论 (1) 所证和谱范数的酉不变性, 有

$$\begin{aligned}\|\boldsymbol{AP}_k - \boldsymbol{P}_k\boldsymbol{D}\|_2 &= \|\boldsymbol{AV}_k\boldsymbol{U} - \boldsymbol{V}_k\boldsymbol{U}\boldsymbol{D}\|_2 = \|\boldsymbol{AV}_k - \boldsymbol{V}_k\boldsymbol{U}\boldsymbol{D}\boldsymbol{U}^{\mathrm{T}}\|_2 \\ &\geqslant \|\boldsymbol{AV}_k - \boldsymbol{V}_k\boldsymbol{T}_k\|_2 = \|\boldsymbol{AV}_k - \boldsymbol{V}_k\boldsymbol{Y}\boldsymbol{M}\boldsymbol{Y}^{\mathrm{T}}\|_2 \\ &= \|\boldsymbol{A}(\boldsymbol{V}_k\boldsymbol{Y}) - (\boldsymbol{V}_k\boldsymbol{Y})\boldsymbol{M}\|_2 = \|\boldsymbol{T}_{ku}\|_2.\end{aligned}$$

注意到 $\boldsymbol{P}_k$ 和 $\boldsymbol{D}$ 的任意性, 可知式 (7.96) 成立. 证毕. □

注 7.5 从定理 7.17 的证明可以看出, 将谱范数换为 Frobenius 范数定理的结论仍然成立. 定理的结论 (1) 说明, 对于实对称矩阵 $\boldsymbol{A}$ 来说, Rayleigh 商 $\boldsymbol{T}_k = \boldsymbol{V}_k^{\mathrm{T}}\boldsymbol{A}\boldsymbol{V}_k$ 具有最佳逼近性; 而定理的结论 (2) 又说明用 Ritz 对作为其近似特征对是最优的.

一般来讲, $\boldsymbol{T}_{ku}$ 是未知的, 但如果 $\boldsymbol{V}_k$ 是由 Lanczos 过程 (即算法 2.13) 产生的, 则有

$$\boldsymbol{T}=\left[\begin{array}{c|c}\boldsymbol{T}_k & \boldsymbol{T}_{uk}\\ \hline \boldsymbol{T}_{ku} & \boldsymbol{T}_u\end{array}\right]=\left[\begin{array}{cccc|ccccc}\alpha_1 & \beta_1 & & & & & & & \\ \beta_1 & \ddots & \ddots & & & & & & \\ & \ddots & \ddots & \beta_{k-1} & & & & & \\ & & \beta_{k-1} & \alpha_k & \beta_k & & & & \\ \hline & & & \beta_k & \alpha_{k+1} & \beta_{k+1} & & & \\ & & & & \beta_{k+1} & \ddots & \ddots & & \\ & & & & & \ddots & \ddots & \beta_{n-1} & \\ & & & & & & \beta_{n-1} & \alpha_n & \end{array}\right] \tag{7.97}$$

从而 $\boldsymbol{T}_{ku}=\beta_k\boldsymbol{e}_1\boldsymbol{e}_k^{\mathrm{T}}$, 其中 β_k 已由算法 2.13 算出, 于是此时有 $\|\boldsymbol{T}_{ku}\|_2=\beta_k$. 此外, 利用 $\boldsymbol{T}_{ku}$ 还可以给出 Ritz 对作为近似特征对的绝对误差.

定理 7.18 假设符号同前, 并设 $\boldsymbol{T}_k$ 的谱分解为

$$\boldsymbol{T}_k=\boldsymbol{Y}\boldsymbol{M}\boldsymbol{Y}^{\mathrm{T}},$$

式中: $\boldsymbol{Y}=[\boldsymbol{y}_1,\boldsymbol{y}_2,\cdots,\boldsymbol{y}_k]$ 为正交矩阵; $\boldsymbol{M}=\mathrm{diag}(\mu_1,\mu_2,\cdots,\mu_k)$.

则有

(1) 存在 $\boldsymbol{A}$ 的 k 个特征值 $\lambda_1, \cdots, \lambda_k$, 使得

$$|\lambda_i - \mu_i| \leqslant \|\boldsymbol{T}_{ku}\|_2, \quad i = 1, 2, \cdots, k. \tag{7.98}$$

(2) 令 $\boldsymbol{z}_i = \boldsymbol{V}_k \boldsymbol{y}_i$, 则

$$\|\boldsymbol{A}\boldsymbol{z}_i - \mu_i \boldsymbol{z}_i\|_2 = \|\boldsymbol{T}_{ku}\boldsymbol{y}_i\|_2, \quad i = 1, 2, \cdots, k. \tag{7.99}$$

(3) 若 $\boldsymbol{V}_k$ 是由 Lanczos 过程 (算法 2.13) 得到的, 则有

$$\|\boldsymbol{T}_{ku}\|_2 = |\beta_k|, \quad \|\boldsymbol{T}_{ku}\boldsymbol{y}_i\|_2 = |\beta_k (\boldsymbol{y}_i)_k|, \tag{7.100}$$

式中: $(\boldsymbol{y}_i)_k$ 为 $\boldsymbol{y}_i$ 的最后一个分量.

证明 (1) 令 $\widehat{\boldsymbol{T}} = \mathrm{diag}(\boldsymbol{T}_k, \boldsymbol{T}_u)$, 则

$$\|\boldsymbol{T} - \widehat{\boldsymbol{T}}\|_2 = \left\| \begin{bmatrix} \boldsymbol{O} & \boldsymbol{T}_{uk} \\ \boldsymbol{T}_{ku} & \boldsymbol{O} \end{bmatrix} \right\|_2 = \|\boldsymbol{T}_{ku}\|_2,$$

然后应用 Weyl 定理即知式 (7.98) 成立.

(2) 直接计算, 有

$$\begin{aligned}\|\boldsymbol{A}\boldsymbol{z}_i-\mu_i\boldsymbol{z}_i\|_2 &= \|\boldsymbol{V}^{\mathrm{T}}(\boldsymbol{A}\boldsymbol{V}_k\boldsymbol{y}_i)-\mu_i\boldsymbol{V}^{\mathrm{T}}(\boldsymbol{V}_k\boldsymbol{y}_i)\|_2\\ &= \left\|\begin{bmatrix}\boldsymbol{T}_k\boldsymbol{y}_i\\ \boldsymbol{T}_{ku}\boldsymbol{y}_i\end{bmatrix}-\begin{bmatrix}\mu_i\boldsymbol{y}_i\\ \boldsymbol{0}\end{bmatrix}\right\|_2\\ &= \left\|\begin{bmatrix}\boldsymbol{0}\\ \boldsymbol{T}_{ku}\boldsymbol{y}_i\end{bmatrix}\right\|_2 = \|\boldsymbol{T}_{ku}\boldsymbol{y}_i\|_2,\end{aligned}$$

故式 (7.99) 成立.

(3) 由式 (7.97) 可得式 (7.100). 证毕. □

§7.6.2 Lanczos 方法

Lanczos 方法是目前计算大型稀疏实对称矩阵少数几个特征值和对应特征向量最常用的 Krylov 子空间方法之一. 本节介绍 Lanczos 方法的详细算法及其相关理论.

1. 经典 Lanczos 方法

设 $\boldsymbol{A}^{\mathrm{T}} = \boldsymbol{A} \in \mathbb{R}^{n\times n}$ 已经给定, 希望求 $\boldsymbol{A}$ 的几个最大特征对或最小特征对. 将算法 2.13 与 7.6.1 节所介绍的 Rayleigh–Ritz 方法相结合就可得到完成这一计算任务的经典 Lanczos 方法.

算法 7.13 (经典 Lanczos 方法) 给定一个 n 阶实对称矩阵 $\boldsymbol{A}$. 本算法计算 $\boldsymbol{A}$ 的少数几个两端特征对.

步 1, 选择初始向量 $\boldsymbol{v}$, 并令 $\boldsymbol{v}_1 = \boldsymbol{v}/\|\boldsymbol{v}\|_2$.

步 2, 应用算法 2.13 产生一个长度为 k 的 Lanczos 分解:

$$\boldsymbol{A}\boldsymbol{V}_k = \boldsymbol{V}_k\boldsymbol{T}_k + \beta_k\boldsymbol{v}_{k+1}\boldsymbol{e}_k^{\mathrm{T}}.$$

步 3, 计算 $\boldsymbol{T}_k$ 的特征值, 并选择其中满足要求的记为 μ_i, $i = 1, 2, \cdots, \ell$.

步 4, 计算 $\boldsymbol{T}_k$ 对应于 μ_i 的特征向量 $\boldsymbol{y}_i$, 并形成 Ritz 向量 $\boldsymbol{u}_i = \boldsymbol{V}_k\boldsymbol{y}_i$, $i = 1, 2, \cdots, \ell$.

步 5, 检验是否已经满足要求. 如果不满足要求, 则增加 k, 再返回到步 2, 重复以上各步.

注 7.6 定理 7.18 说明, 若 Rayleigh 商 $\boldsymbol{T}_k$ 的一个特征对 $(\mu, \boldsymbol{y})$ 使得 $|\beta_k| \cdot |\boldsymbol{e}_k^{\mathrm{T}}\boldsymbol{y}|$ 很小, 则 μ 就是 $\boldsymbol{A}$ 的某个特征值的很好近似. 因此, 通常在算法 7.13 中就是以 $|\beta_k| \cdot |\boldsymbol{e}_k^{\mathrm{T}}\boldsymbol{y}|$ 是否已经足够小来判断 Ritz 对 $(\mu, \boldsymbol{V}_k\boldsymbol{y})$ 是否可以作为 $\boldsymbol{A}$ 的近似特征对.




Back
Close

经典 Lanzcos 方法的 MATLAB 程序如下:

```
function [mu,U,lambda]=Class_Lanczos(A,v,k)
%输入:A为对称矩阵,v为初始向量,k为子空间的维数
%输出:lambda和U分别是A的k个特征值及相应的特征向量
tol=1.e-12; v=v/norm(v); V(:,1)=v;
for i=1:k
    [V,T,beta]=Lanczos2(A,v,i);
    [Y,Mu]=eig(T);
    mu{i}=diag(Mu); U=V*Y;
    if beta*abs(Y(i,i))<tol
       k=i; break;
    end
end
lambda=mu{k};
```

例 7.12 令 $\boldsymbol{A} \in \mathbb{R}^{500\times 500}$ 是一个随机产生的对角矩阵, 其对角元服从正态分布. 对于 k 从 1 到 40, 利用算法 7.13 计算出 $\boldsymbol{T}_k$ 的特征值, 并标注在图 7.2 中. 图中第 k 列 "+" 表示 $\boldsymbol{T}_k$ 的所有特征值, 最后一列 "+" 表示 $\boldsymbol{A}$ 的精确特征值.

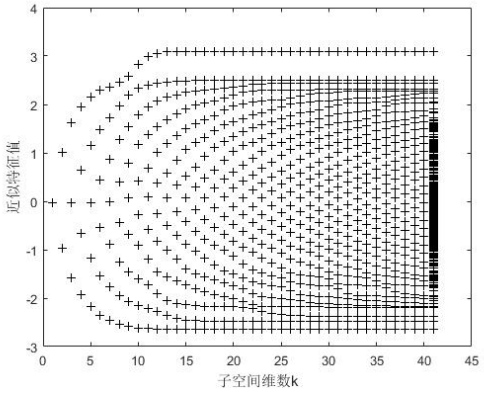

图 7.2 Lanczos 算法的收敛特性

从这一数值例子的计算结果可以看出:

(1) 两端特征值 (即最大和最小特征值) 收敛最快, 而内部特征值收敛较慢.

(2) $\boldsymbol{T}_k$ 的第 i 个最大 (或最小) 特征值单调增加 (或减少) 地收敛到 $\boldsymbol{A}$ 的第 i 个最大 (或最小) 特征值.

2. 收敛性理论

下面对 Lanczos 方法及其收敛性进行一些理论上的分析. 为此, 先引进两个基本概念.

设 $\boldsymbol{x}, \boldsymbol{y} \in \mathbb{R}^n$ 是两个非零向量, 定义 $\boldsymbol{x}$ 与 $\boldsymbol{y}$ 之间的夹角为

$$\theta := \theta(\boldsymbol{x}, \boldsymbol{y}) = \arccos \frac{|\boldsymbol{x}^{\mathrm{T}}\boldsymbol{y}|}{\|\boldsymbol{x}\|_2 \|\boldsymbol{y}\|_2}, \tag{7.101}$$

其几何解释如图 7.3 所示.

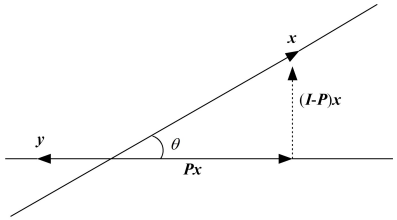

图 7.3 向量 $\boldsymbol{x}$ 与 $\boldsymbol{y}$ 之间的夹角

令

$$\boldsymbol{P} = \frac{1}{\|\boldsymbol{y}\|_2^2}\boldsymbol{y}\boldsymbol{y}^{\mathrm{T}},$$

即 $\boldsymbol{P}$ 是到 $\mathrm{span}\{\boldsymbol{y}\}$ 上的正交投影, 则有

$$\tan\theta = \frac{\|(\boldsymbol{I}-\boldsymbol{P})\boldsymbol{x}\|_2}{\|\boldsymbol{P}\boldsymbol{x}\|_2}, \quad \sin\theta = \frac{\|(\boldsymbol{I}-\boldsymbol{P})\boldsymbol{x}\|_2}{\|\boldsymbol{x}\|_2}, \quad \cos\theta = \frac{\|\boldsymbol{P}\boldsymbol{x}\|_2}{\|\boldsymbol{x}\|_2}.$$

设 $\boldsymbol{x} \in \mathbb{R}^n$ 是一个非零向量, $\mathcal{V}$ 是一个子空间, 则定义 $\boldsymbol{x}$ 与 $\mathcal{V}$ 之间的夹角为

$$\theta(\boldsymbol{x}, \mathcal{V}) = \min\{\theta(\boldsymbol{x}, \boldsymbol{u}) : \boldsymbol{0} \neq \boldsymbol{u} \in \mathcal{V}\}. \tag{7.102}$$




Back
Close

再假设 $\boldsymbol{P}$ 是 $\mathcal{V}$ 上的正交投影, 则有

$$\tan\theta(\boldsymbol{x},\mathcal{V}) = \frac{\|(\boldsymbol{I}-\boldsymbol{P})\boldsymbol{x}\|_2}{\|\boldsymbol{P}\boldsymbol{x}\|_2}. \tag{7.103}$$

式 (7.103) 的几何解释如图 7.4 所示.

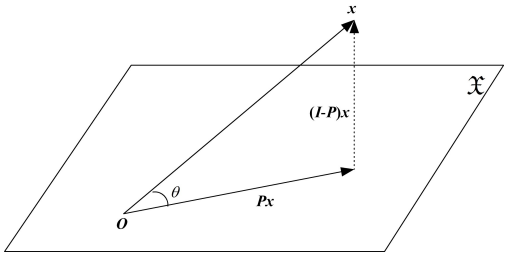

图 7.4 向量 $\boldsymbol{x}$ 与子空间 $\mathcal{X}$ 之间的夹角

事实上, 可以证明 $\theta(\boldsymbol{x},\mathcal{V}) = \theta$, 其中 θ 如图 7.4 所示. 记 $\boldsymbol{x}_1 = \boldsymbol{P}\boldsymbol{x}, \boldsymbol{x}_2 = (\boldsymbol{I}-\boldsymbol{P})\boldsymbol{x}$, 则有

$$\boldsymbol{x} = \boldsymbol{x}_1 + \boldsymbol{x}_2, \quad \boldsymbol{x}_1^{\mathrm{T}}\boldsymbol{x}_2 = 0, \quad \boldsymbol{x}_2 \perp \mathcal{V},$$

于是对任意的 $\boldsymbol{u} \in \mathcal{V}$, 有

$$|\boldsymbol{x}^{\mathrm{T}}\boldsymbol{u}| = |\boldsymbol{x}_1^{\mathrm{T}}\boldsymbol{u}| \leqslant \|\boldsymbol{x}_1\|_2\|\boldsymbol{u}\|_2,$$

从而

$$\frac{|\boldsymbol{x}^{\mathrm{T}}\boldsymbol{u}|}{\|\boldsymbol{x}\|_2\|\boldsymbol{u}\|_2} \leqslant \frac{\|\boldsymbol{x}_1\|_2}{\|\boldsymbol{x}\|_2} = \frac{|\boldsymbol{x}_1^{\mathrm{T}}\boldsymbol{x}_1|}{\|\boldsymbol{x}\|_2\|\boldsymbol{x}_1\|_2} = \frac{|\boldsymbol{x}^{\mathrm{T}}\boldsymbol{x}_1|}{\|\boldsymbol{x}\|_2\|\boldsymbol{x}_1\|_2}.$$

于是有

$$\theta(\boldsymbol{x}, \boldsymbol{u}) \geqslant \theta(\boldsymbol{x}, \boldsymbol{x}_1) = \theta.$$

注意到 $\boldsymbol{u}$ 的任意性, 便有 $\theta = \min\limits_{\boldsymbol{u}\in\mathcal{X}} \theta(\boldsymbol{x}, \boldsymbol{u}) = \theta(\boldsymbol{x}, \mathcal{V})$.

为了叙述简洁, 先将下面所要使用的符号作一说明. 对给定的 $\boldsymbol{A}^{\mathrm{T}} = \boldsymbol{A} \in \mathbb{R}^{n\times n}$ 和 $\boldsymbol{0} \neq \boldsymbol{v} \in \mathbb{R}^n$, 简记

$$\mathcal{K}_k = \mathcal{K}_k(\boldsymbol{A}, \boldsymbol{v}),$$

Back

Close

并假定与其对应的长度为 k 的 Lanczos 分解为

$$\boldsymbol{A}\boldsymbol{V}_k = \boldsymbol{V}_k\boldsymbol{T}_k + \beta_k \boldsymbol{v}_{k+1}\boldsymbol{e}_k^{\mathrm{T}},$$

式中: $\boldsymbol{v}_1 = \boldsymbol{v}/\|\boldsymbol{v}\|_2$. 再假定 $\boldsymbol{A}$ 和 $\boldsymbol{T}_k$ 的谱分解分别为

$$\boldsymbol{A} = \boldsymbol{U}\boldsymbol{\Lambda}\boldsymbol{U}^{\mathrm{T}} \quad 和 \quad \boldsymbol{T}_k = \boldsymbol{Y}\boldsymbol{M}\boldsymbol{Y}^{\mathrm{T}},$$

式中:

$$\begin{aligned}
&\boldsymbol{\Lambda} = \mathrm{diag}(\lambda_1, \lambda_2, \cdots, \lambda_n), \quad \lambda_1 \geqslant \lambda_2 \geqslant \cdots \geqslant \lambda_n,\\
&\boldsymbol{M} = \mathrm{diag}(\mu_1, \mu_2, \cdots, \mu_k), \quad \mu_1 \geqslant \mu_2 \geqslant \cdots \geqslant \mu_k,\\
&\boldsymbol{U} = [\boldsymbol{u}_1, \boldsymbol{u}_2, \cdots, \boldsymbol{u}_n], \quad \boldsymbol{Y} = [\boldsymbol{y}_1, \boldsymbol{y}_2, \cdots, \boldsymbol{y}_k].
\end{aligned}$$

如果没有特别说明, 在本节后续部分中所遇到的这些符号均为上面所述的含义.




Back
Close

首先给出一个简单的结果. 若 $(\lambda, \boldsymbol{u})$ 是 $\boldsymbol{A}$ 的一个特征对, 而且如果 $\boldsymbol{u} \in \mathcal{K}_k$, 则 λ 必是 $\boldsymbol{T}_k$ 的特征值. 事实上, $\boldsymbol{u} \in \mathcal{K}_k$ 蕴涵着必存在 $\boldsymbol{y}$ 使得 $\boldsymbol{u} = \boldsymbol{V}_k \boldsymbol{y}$, 这样利用 Lanczos 分解有

$$\lambda \boldsymbol{V}_k \boldsymbol{y} = \lambda \boldsymbol{u} = \boldsymbol{A}\boldsymbol{u} = \boldsymbol{A}\boldsymbol{V}_k \boldsymbol{y} = \boldsymbol{V}_k \boldsymbol{T}_k \boldsymbol{y} + \beta_k \boldsymbol{v}_{k+1} \boldsymbol{e}_k^{\mathrm{T}} \boldsymbol{y}.$$

在上面的等式两边左乘 $\boldsymbol{V}_k^{\mathrm{T}}$, 并注意到 $\boldsymbol{V}_k^{\mathrm{T}} \boldsymbol{v}_{k+1} = \boldsymbol{0}$, 有

$$\lambda \boldsymbol{y} = \boldsymbol{T}_k \boldsymbol{y},$$

即 λ 是 $\boldsymbol{T}_k$ 的特征值 (这里 $\boldsymbol{u} \neq \boldsymbol{0}$ 蕴涵着 $\boldsymbol{y} \neq \boldsymbol{0}$). 因此, 如果 $\boldsymbol{u} \notin \mathcal{K}_k$, 但若 $\boldsymbol{u}$ 与 $\mathcal{K}_k$ 很靠近的话, 可望 $\boldsymbol{T}_k$ 有特征值是 λ 的很好近似. 进一步, 可以证明如下结果.

定理 7.19 令 $\boldsymbol{A}\boldsymbol{u} = \lambda \boldsymbol{u}$, $\|\boldsymbol{u}\|_2 = 1$, 则必存在 $\boldsymbol{T}_k$ 的一个特征值 μ, 使得

$$|\mu - \lambda| \leqslant \|\boldsymbol{A}\|_2 \tan \theta(\boldsymbol{u}, \mathcal{K}_k), \tag{7.104}$$

式中: $\mathcal{K}_k = \mathcal{K}_k(\boldsymbol{A}, \boldsymbol{v})$.




证明 令 $\boldsymbol{P}=\boldsymbol{V}_k\boldsymbol{V}_k^{\mathrm{T}}$ 是 $\mathcal{K}_k$ 上的正交投影, 并记

$$\boldsymbol{u}_1=\boldsymbol{P}\boldsymbol{u},\quad \boldsymbol{u}_2=(\boldsymbol{I}-\boldsymbol{P})\boldsymbol{u},$$

则有

$$\tan\theta(\boldsymbol{u},\mathcal{K}_k)=\frac{\|\boldsymbol{u}_2\|_2}{\|\boldsymbol{u}_1\|_2}, \tag{7.105}$$

而且

$$\begin{aligned}\lambda\boldsymbol{u}_1+\lambda\boldsymbol{u}_2=\lambda\boldsymbol{u}&=\boldsymbol{A}\boldsymbol{u}=\boldsymbol{A}\boldsymbol{u}_1+\boldsymbol{A}\boldsymbol{u}_2\\&=\boldsymbol{A}\boldsymbol{V}_k\boldsymbol{V}_k^{\mathrm{T}}\boldsymbol{u}_1+\boldsymbol{A}\boldsymbol{u}_2\\&=\boldsymbol{V}_k\boldsymbol{T}_k\boldsymbol{V}_k^{\mathrm{T}}\boldsymbol{u}_1+\beta_k\boldsymbol{v}_{k+1}\boldsymbol{e}_k^{\mathrm{T}}\boldsymbol{V}_k^{\mathrm{T}}\boldsymbol{u}_1+\boldsymbol{A}\boldsymbol{u}_2.\end{aligned}$$

在上式两边左乘 $\boldsymbol{V}_k^{\mathrm{T}}$, 得

$$\lambda\boldsymbol{V}_k^{\mathrm{T}}\boldsymbol{u}_1=\boldsymbol{T}_k\boldsymbol{V}_k^{\mathrm{T}}\boldsymbol{u}_1+\boldsymbol{V}_k^{\mathrm{T}}\boldsymbol{A}\boldsymbol{u}_2,$$

从而有

$$(\lambda \boldsymbol{I} - \boldsymbol{T}_k)\boldsymbol{V}_k^{\mathrm{T}}\boldsymbol{u}_1 = \boldsymbol{V}_k^{\mathrm{T}}\boldsymbol{A}\boldsymbol{u}_2.$$

于是

$$\begin{aligned}
\min_{\mu\in\lambda(\boldsymbol{T}_k)} |\lambda-\mu| \cdot \|\boldsymbol{V}_k^{\mathrm{T}}\boldsymbol{u}_1\|_2 = \min_{\mu\in\lambda(\boldsymbol{T}_k)} |\lambda-\mu| \cdot \|\boldsymbol{Y}^{\mathrm{T}}\boldsymbol{V}_k^{\mathrm{T}}\boldsymbol{u}_1\|_2 \\
\leqslant \|(\lambda \boldsymbol{I} - \boldsymbol{M})\boldsymbol{Y}^{\mathrm{T}}\boldsymbol{V}_k^{\mathrm{T}}\boldsymbol{u}_1\|_2 \qquad (7.106) \\
= \|(\lambda \boldsymbol{I} - \boldsymbol{T}_k)\boldsymbol{V}_k^{\mathrm{T}}\boldsymbol{u}_1\|_2 = \|\boldsymbol{V}_k^{\mathrm{T}}\boldsymbol{A}\boldsymbol{u}_2\|_2 \\
\leqslant \|\boldsymbol{V}_k^{\mathrm{T}}\|_2\|\boldsymbol{A}\|_2\|\boldsymbol{u}_2\|_2 = \|\boldsymbol{A}\|_2\|\boldsymbol{u}_2\|_2.
\end{aligned}$$

此外, 由于 $\boldsymbol{u}_1 \in \mathcal{K}_k$, 故存在 $\boldsymbol{y}$ 使得 $\boldsymbol{u}_1 = \boldsymbol{V}_k\boldsymbol{y}$, 从而有

$$\|\boldsymbol{V}_k^{\mathrm{T}}\boldsymbol{u}_1\|_2 = \|\boldsymbol{V}_k^{\mathrm{T}}\boldsymbol{V}_k\boldsymbol{y}\|_2 = \|\boldsymbol{y}\|_2 = \|\boldsymbol{u}_1\|_2, \qquad (7.107)$$

将式 (7.105)、式 (7.106) 与式 (7.107) 相结合便得到所要证的不等式 (7.104). 证毕. □




注 7.7 定理 7.19 是说, 如果对应于特征值 λ 的特征向量 $\boldsymbol{u}$ 与 Krylov 子空间 $\mathcal{K}_k$ 之间的距离 $\tan\theta(\boldsymbol{u},\mathcal{K}_k)$ 很小, 则 $\boldsymbol{T}_k$ 就必有一个特征值 μ 和 λ 很靠近. 因此, 可望由 Lanczos 算法得到 λ 的一个很好的近似值.

下面的定理给出了第 i 个特征向量 $\boldsymbol{u}_i$ 与 $\mathcal{K}_k$ 之间的距离 $\tan\theta(\boldsymbol{u}_i,\mathcal{K}_k)$ 的一个可计算的上界估计.

定理 7.20 对给定的 $i\,(1\leqslant i<k)$, 若

$$\lambda_{i-1}>\lambda_i,\qquad \lambda_{i+1}>\lambda_n,\qquad \boldsymbol{v}^{\mathrm{T}}\boldsymbol{u}_i\neq 0,$$

则有

$$\tan\theta(\boldsymbol{u}_i,\mathcal{K}_k)\leqslant\frac{\xi_i}{C_{k-i}(1+2\delta_i)}\tan\theta(\boldsymbol{u}_i,\boldsymbol{v}),\tag{7.108}$$

式中: $\mathcal{K}_k=\mathcal{K}_k(\boldsymbol{A},\boldsymbol{v})$; $\delta_i=\dfrac{\lambda_i-\lambda_{i+1}}{\lambda_{i+1}-\lambda_n}$; $C_{k-i}(t)$ 为 $k-i$ 次 Chebyshev

多项式; 而

$$\xi_i = \begin{cases} 1, & i = 1, \\ \prod_{s=1}^{i-1} \frac{\lambda_s - \lambda_n}{\lambda_s - \lambda_i}, & i > 1. \end{cases} \tag{7.109}$$

证明 记 $\boldsymbol{P}_i = \boldsymbol{u}_i \boldsymbol{u}_i^{\mathrm{T}}$, 即 $\boldsymbol{P}_i$ 是到子空间 $\mathrm{span}\{\boldsymbol{u}_i\}$ 上的正交投影, 因此, 有

$$\tan\theta(\boldsymbol{u}_i, \boldsymbol{v}) = \frac{\|(\boldsymbol{I} - \boldsymbol{P}_i)\boldsymbol{v}\|_2}{\|\boldsymbol{P}_i \boldsymbol{v}\|_2} \quad (\boldsymbol{P}_i \boldsymbol{v} \neq \boldsymbol{0}). \tag{7.110}$$

此外, 任取 $\boldsymbol{x} \in \mathcal{K}_k$, 则必存在一个 $p \in \mathcal{P}_{k-1}$, 使得 $\boldsymbol{x} = p(\boldsymbol{A})\boldsymbol{v}$. 当然这里假定 $\boldsymbol{x} \neq \boldsymbol{0}$. 注意到 $\boldsymbol{P}_i \boldsymbol{A} = \boldsymbol{A}\boldsymbol{P}_i$, 便有

$$\boldsymbol{P}_i \boldsymbol{x} = p(\boldsymbol{A})\boldsymbol{P}_i \boldsymbol{v} = p(\lambda_i)\boldsymbol{P}_i \boldsymbol{v},$$

$$(\boldsymbol{I} - \boldsymbol{P}_i)\boldsymbol{x} = p(\boldsymbol{A})(\boldsymbol{I} - \boldsymbol{P}_i)\boldsymbol{v} = p(\boldsymbol{A})\boldsymbol{v}_i,$$

式中: $\boldsymbol{v}_i = (\boldsymbol{I} - \boldsymbol{P}_i)\boldsymbol{v}$. 这样

$$\tan\theta(\boldsymbol{u}_i,\boldsymbol{x}) = \frac{\|(\boldsymbol{I}-\boldsymbol{P}_i)\boldsymbol{x}\|_2}{\|\boldsymbol{P}_i\boldsymbol{x}\|_2} = \frac{\|p(\boldsymbol{A})\boldsymbol{v}_i\|_2}{|p(\lambda_i)|\cdot\|\boldsymbol{P}_i\boldsymbol{v}\|_2}$$
$$= \left\|\frac{p(\boldsymbol{A})}{p(\lambda_i)}\frac{\boldsymbol{v}_i}{\|\boldsymbol{v}_i\|_2}\right\|_2 \frac{\|(\boldsymbol{I}-\boldsymbol{P}_i)\boldsymbol{v}\|_2}{\|\boldsymbol{P}_i\boldsymbol{v}\|_2}$$
$$= \|\widehat{p}(\boldsymbol{A})\widehat{\boldsymbol{v}}_i\|_2 \tan\theta(\boldsymbol{u}_i,\boldsymbol{v}), \qquad (7.111)$$

式中: $\widehat{p}(t) = p(t)/p(\lambda_i)$; $\widehat{\boldsymbol{v}}_i = \boldsymbol{v}_i/\|\boldsymbol{v}_i\|_2$. 这里最后一个等式用到了式 (7.110).

现将 $\widehat{\boldsymbol{v}}_i$ 用 $\boldsymbol{A}$ 的特征向量展开, 即令

$$\widehat{\boldsymbol{v}}_i = \alpha_1\boldsymbol{u}_1 + \alpha_2\boldsymbol{u}_2 + \cdots + \alpha_n\boldsymbol{u}_n.$$

而 $\widehat{\boldsymbol{v}}_i \perp \boldsymbol{u}_i$, 故必有 $\alpha_i = 0$. 此外, $\alpha_1^2 + \cdots + \alpha_n^2 = \|\widehat{\boldsymbol{v}}_i\|_2^2 = 1$. 这样, 有




$$\|\widehat{p}(\boldsymbol{A})\widehat{\boldsymbol{v}}_i\|_2^2 = \sum_{s=1}^{n} \alpha_s^2 \widehat{p}(\lambda_s)^2 \leqslant \max_{1\leqslant s\leqslant n, s\neq i} |\widehat{p}(\lambda_s)|^2. \tag{7.112}$$

现在取一个特殊的多项式 $p_* \in \mathcal{P}_{k-1}$:

$$p_*(t) = \frac{(\lambda_1 - t)(\lambda_2 - t)\cdots(\lambda_{i-1} - t)}{(\lambda_1 - \lambda_i)(\lambda_2 - \lambda_i)\cdots(\lambda_{i-1} - \lambda_i)} v(t),$$

这里

$$v(t) = C_{k-i}\Big(\frac{2t - (\lambda_{i+1} + \lambda_n)}{\lambda_{i+1} - \lambda_n}\Big) / C_{k-i}(1 + 2\delta_i).$$

容易验证, 对于这个多项式 p_*, 有

$$p_*(\lambda_i) = 1, \quad p_*(\lambda_s) = 0, \quad s = 1, 2, \cdots, i-1.$$




从而有 $p_*(t) \equiv \widehat{p}_*(t)$, 而且

$$\max_{1\leqslant s\leqslant n,\, s\neq i} |p_*(\lambda_s)| = \max_{i+1\leqslant s\leqslant n} |p_*(\lambda_s)| \leqslant \xi_i \max_{i+1\leqslant s\leqslant n} |v(\lambda_s)|$$

$$\leqslant \frac{\xi_i}{C_{k-i}(1+2\delta_i)}, \tag{7.113}$$

这里最后一个等式用到 Chebyshev 多项式的性质. 现将式 (7.111)、式 (7.112) 和式 (7.113) 相结合, 有

$$\begin{aligned}\tan\theta(\boldsymbol{u}_i, \mathcal{K}_k) &= \min_{x\in\mathcal{K}_k} \tan\theta(\boldsymbol{u}_i, \boldsymbol{x}) \\ &\leqslant \max_{1\leqslant s\leqslant n,\, s\neq i} |p_*(\lambda_s)| \tan\theta(\boldsymbol{u}_i, \boldsymbol{v}) \\ &\leqslant \frac{\xi_i}{C_{k-i}(1+2\delta_i)} \tan\theta(\boldsymbol{u}_i, \boldsymbol{v}).\end{aligned}$$

这正是要证的结论. 证毕. □

定理 7.21 (Kaniel–Saad 定理) 对于任意给定的 $i(1 \leqslant i < k)$, 若

$$\lambda_i \neq \mu_s, \quad s = 1, 2, \cdots, i-1, \quad \lambda_{i+1} > \lambda_n,$$

而且 $\boldsymbol{v}^{\mathrm{T}}\boldsymbol{u}_i \neq 0$, 则有

$$0 \leqslant \lambda_i - \mu_i \leqslant (\lambda_i - \lambda_n)\left[\frac{\zeta_i}{C_{k-i}(1+2\delta_i)}\tan\theta(\boldsymbol{u}_i, \boldsymbol{v})\right]^2, \tag{7.114}$$

其中 δ_i 由定理 7.20 给出, 而

$$\zeta_i = \begin{cases} 1, & i = 1, \\ \max\limits_{i+1 \leqslant \ell \leqslant n} \prod\limits_{s=1}^{i-1} \frac{\mu_s - \lambda_\ell}{\mu_s - \lambda_i}, & i > 1. \end{cases} \tag{7.115}$$

注 7.8 不等式 (7.114) 右端可分为三部分: 第一部分为 $(\lambda_i - \lambda_n)\zeta_i^2$, 主要是由 λ_i 与 $\mu_1, \cdots, \mu_{i-1}$ 的分离程度确定; 第二部分为 $\tan^2\theta(\boldsymbol{u}_i, \boldsymbol{v})$, 是由初始向量 $\boldsymbol{v}$ 所确定, 反映了 $\boldsymbol{v}$ 中含有特征向量

$\boldsymbol{u}_i$ 的成分的大小; 第三部分为 $[C_{k-i}(1+2\delta_i)]^{-2}$, 这是关键的一部分, 反映了 Lanczos 方法的收敛性, 当 i 较小且 $0<\delta_i<0.1$ 时, 由 Chebyshev 多项式的性质可知, 此时随着 k 的增加, 这个值减少得非常快.

注 7.9 定理 7.21 实质上表明, 随着 Lanczos 迭代次数 k 的增加, $\boldsymbol{T}_k$ 的几个最大特征值 $\mu_1,\cdots,\mu_s$ (s 较小) 将非常快地收敛到 $\boldsymbol{A}$ 的前几个最大的特征值 $\lambda_1,\cdots,\lambda_s$. 特别地, 对于 $i=1$, 不等式 (7.114) 就变为

$$0\leqslant\lambda_1-\mu_1\leqslant(\lambda_1-\lambda_n)\tan^2\theta(\boldsymbol{u}_1,\boldsymbol{v})[C_{k-1}(1+2\delta_1)]^{-2},$$

由此更容易看出, 随着 k 的增加, μ_1 与 λ_1 之间的差距迅速地缩小.

注 7.10 对 $-\boldsymbol{A}$ 应用定理 7.21 可知, $\boldsymbol{T}_k$ 的几个最小的特征值 $\mu_{k-s},\mu_{k-s+1},\cdots,\mu_k$ 将随着 k 的增加很快地收敛到 $\boldsymbol{A}$ 的几个




最小的特征值 $\lambda_{n-s}, \lambda_{n-s+1}, \cdots, \lambda_n$. 特别地, 有

$$0 \leqslant \mu_k - \lambda_n \leqslant (\lambda_1 - \lambda_n) \tan^2 \theta(\boldsymbol{u}_n, \boldsymbol{v}) [C_{k-1}(1 + 2\widetilde{\delta}_1)]^{-2},$$

式中: $\widetilde{\delta}_1 = \frac{\lambda_{n-1} - \lambda_n}{\lambda_1 - \lambda_{n-1}}$. 随着 k 的增加, $\boldsymbol{T}_k$ 的最小特征值 μ_k 将很快地收敛到 $\boldsymbol{A}$ 的最小特征值 λ_n.

为了证明定理 7.21, 先将证明中需要的几个基本结果总结在下面的引理中.

引理 7.6 符号如前所述, 则有

(1) $\boldsymbol{T}_k$ 的每个特征值 μ_i 可表示为

$$\mu_i = \max_{\boldsymbol{0} \neq \boldsymbol{u} \in \mathcal{U}_{k-i+1}} \frac{\boldsymbol{u}^{\mathrm{T}} \boldsymbol{A} \boldsymbol{u}}{\boldsymbol{u}^{\mathrm{T}} \boldsymbol{u}}, \quad i = 1, 2, \cdots, k, \tag{7.116}$$

式中:

$$\mathcal{U}_{k-i+1} = \mathrm{span}\{\boldsymbol{V}_k \boldsymbol{y}_i, \boldsymbol{V}_k \boldsymbol{y}_{i+1}, \cdots, \boldsymbol{V}_k \boldsymbol{y}_k\}. \tag{7.117}$$

(2) 由式 (7.117) 所定义的子空间 $\mathcal{U}_{k-i+1}$ 可表示为

$$\mathcal{U}_{k-i+1} = \{p(\boldsymbol{A})\boldsymbol{v} : p \in \mathcal{P}_{k-1},\ p(\mu_s) = 0,\ s = 1, 2, \cdots, i-1\}. \tag{7.118}$$

证明 (1) 由 $\boldsymbol{T}_k$ 的谱分解易证

$$\mu_i = \max_{\boldsymbol{0} \neq \boldsymbol{y} \in \mathcal{Y}_{k-i+1}} \frac{\boldsymbol{y}^{\mathrm{T}} \boldsymbol{T}_k \boldsymbol{y}}{\boldsymbol{y}^{\mathrm{T}} \boldsymbol{y}}, \tag{7.119}$$

式中: $\mathcal{Y}_{k-i+1} = \mathrm{span}\{\boldsymbol{y}_i, \boldsymbol{y}_{i+1}, \cdots, \boldsymbol{y}_k\}$. 注意到

$$\frac{\boldsymbol{y}^{\mathrm{T}} \boldsymbol{T}_k \boldsymbol{y}}{\boldsymbol{y}^{\mathrm{T}} \boldsymbol{y}} = \frac{\boldsymbol{y}^{\mathrm{T}} \boldsymbol{V}_k^{\mathrm{T}} \boldsymbol{A} \boldsymbol{V}_k \boldsymbol{y}}{\boldsymbol{y}^{\mathrm{T}} \boldsymbol{V}_k^{\mathrm{T}} \boldsymbol{V}_k \boldsymbol{y}} = \frac{\boldsymbol{u}^{\mathrm{T}} \boldsymbol{A} \boldsymbol{u}}{\boldsymbol{u}^{\mathrm{T}} \boldsymbol{u}}, \qquad \boldsymbol{u} = \boldsymbol{V}_k \boldsymbol{y},$$

以及 $\boldsymbol{V}_k \mathcal{Y}_{k-i+1} = \mathcal{U}_{k-i+1}$, 由式 (7.119) 可知式 (7.116) 成立.

(2) 由于

$$\mathcal{U}_{k-i+1} \subset \mathcal{R}(\boldsymbol{V}_k) = \mathcal{K}_k = \{p(\boldsymbol{A})\boldsymbol{v} : p \in \mathcal{P}_{k-1}\},$$

$$\mathcal{U}_{k-i+1}^{\perp} = \mathrm{span}\{\boldsymbol{V}_k \boldsymbol{y}_1, \boldsymbol{V}_k \boldsymbol{y}_2, \cdots, \boldsymbol{V}_k \boldsymbol{y}_{i-1}\},$$

故 $\boldsymbol{u} \in \mathcal{U}_{k-i+1}$ 的充分必要条件是存在 $p \in \mathcal{P}_{k-1}$ 满足

$$\boldsymbol{v}^{\mathrm{T}} p(\boldsymbol{A}) \boldsymbol{V}_k \boldsymbol{y}_s = 0, \quad s = 1, 2, \cdots, i-1, \tag{7.120}$$

使得 $\boldsymbol{u} = p(\boldsymbol{A})\boldsymbol{v}$.

注意到 $\boldsymbol{v}_{k+1} \perp \mathcal{K}_k$, 从 Lanczos 分解出发, 归纳地可以证明

$$\boldsymbol{v}^{\mathrm{T}} \boldsymbol{A}^s \boldsymbol{V}_k = \boldsymbol{v}^{\mathrm{T}} \boldsymbol{V}_k \boldsymbol{T}_k^s, \quad s = 0, 1, \cdots, k-1.$$

从而, 对任意的 $p \in \mathcal{P}_{k-1}$, 有

$$\boldsymbol{v}^{\mathrm{T}} p(\boldsymbol{A}) \boldsymbol{V}_k = \boldsymbol{v}^{\mathrm{T}} \boldsymbol{V}_k p(\boldsymbol{T}_k). \tag{7.121}$$

上式两边右乘 $\boldsymbol{y}_s$, 得

$$\boldsymbol{v}^{\mathrm{T}} p(\boldsymbol{A}) \boldsymbol{V}_k \boldsymbol{y}_s = \boldsymbol{v}^{\mathrm{T}} \boldsymbol{V}_k p(\mu_s) \boldsymbol{y}_j = \|\boldsymbol{v}\|_2 p(\mu_j) \boldsymbol{e}_1^{\mathrm{T}} \boldsymbol{y}_s,$$

这里用到了 $\boldsymbol{v}^{\mathrm{T}} \boldsymbol{V}_k = \|\boldsymbol{v}\|_2 \boldsymbol{v}_1^{\mathrm{T}} \boldsymbol{V}_k = \|\boldsymbol{v}\|_2 \boldsymbol{e}_1^{\mathrm{T}}$. 再注意到 $\boldsymbol{T}_k$ 是不可约的对称三对角矩阵蕴含着 $\boldsymbol{e}_1^{\mathrm{T}} \boldsymbol{y}_s \neq 0$, 便知式 (7.120) 成立的充分必

Back

Close

要条件是

$$p(\mu_s) = 0, \quad s = 1, 2, \cdots, i-1.$$

由此可知式 (7.118) 成立. 证毕. □

有了前面的准备工作, 下面证明定理 7.21.

定理 7.21 的证明 注意到式 (7.94), 并且应用推论 7.2, 即知 $\lambda_i - \mu_i \geqslant 0$. 再应用引理 7.6 的结论 (1), 有

$$\begin{aligned} 0 \leqslant \lambda_i - \mu_i &= \lambda_i - \max_{\boldsymbol{0} \neq \boldsymbol{u} \in \mathcal{U}_{k-i+1}} \frac{\boldsymbol{u}^{\mathrm{T}} \boldsymbol{A} \boldsymbol{u}}{\boldsymbol{u}^{\mathrm{T}} \boldsymbol{u}} \\ &= \min_{\boldsymbol{0} \neq \boldsymbol{u} \in \mathcal{U}_{k-i+1}} \frac{\boldsymbol{u}^{\mathrm{T}} (\lambda_i \boldsymbol{I} - \boldsymbol{A}) \boldsymbol{u}}{\boldsymbol{u}^{\mathrm{T}} \boldsymbol{u}}. \end{aligned} \tag{7.122}$$

现在定义 $\widehat{p}(t)=g(t)h(t)$, 其中

$$g(t)=\begin{cases}1, & i=1,\\ \dfrac{(\mu_1-t)\cdots(\mu_{i-1}-t)}{(\mu_1-\lambda_i)\cdots(\mu_{i-1}-\lambda_i)}, & i>1,\end{cases}$$

$$h(t)=C_{k-i}\left(\frac{2t-\lambda_{i+1}-\lambda_n}{\lambda_{i+1}-\lambda_n}\right)\Big/C_{k-i}(1+2\delta_i).$$

容易验证, 这样定义的多项式 $\widehat{p}(t)$ 满足 $\widehat{p}(t)\in\mathcal{P}_{k-1}$, 且

$$\widehat{p}(\lambda_i)=1,\quad \widehat{p}(\mu_s)=0,\quad s=1,2,\cdots,i-1.$$

再利用引理 7.6 的结论 (2), 即有 $\widehat{\boldsymbol{u}}=\widehat{p}(\boldsymbol{A})\boldsymbol{v}\in\mathcal{U}_{k-i+1}$.

设 $\boldsymbol{v}=\gamma_1\boldsymbol{u}_1+\cdots+\gamma_n\boldsymbol{u}_n$. 直接计算有

$$\widehat{\boldsymbol{u}}^{\mathrm{T}}\widehat{\boldsymbol{u}}=\boldsymbol{v}^{\mathrm{T}}\widehat{p}(\boldsymbol{A})^2\boldsymbol{v}=\gamma_1^2\,\widehat{p}(\lambda_1)^2+\cdots+\gamma_n^2\,\widehat{p}(\lambda_n)^2\geqslant\gamma_i^2 \qquad (7.123)$$

和

$$\begin{aligned}
\widehat{\boldsymbol{u}}^{\mathrm{T}}(\lambda_i \boldsymbol{I} - \boldsymbol{A})\widehat{\boldsymbol{u}} &= \boldsymbol{v}^{\mathrm{T}}\widehat{p}(\boldsymbol{A})(\lambda_i \boldsymbol{I} - \boldsymbol{A})\widehat{p}(\boldsymbol{A})\boldsymbol{v} \\
&= \sum_{s=1}^{n} \gamma_s^2\, \widehat{p}(\lambda_s)^2(\lambda_i - \lambda_s) \\
&= \sum_{s=1}^{i-1} \gamma_s^2\, \widehat{p}(\lambda_s)^2(\lambda_i - \lambda_s) + \sum_{s=i+1}^{n} \gamma_s^2\, \widehat{p}(\lambda_s)^2)(\lambda_i - \lambda_s) \\
&\leqslant \sum_{s=i+1}^{n} \gamma_s^2\, \widehat{p}(\lambda_s)^2(\lambda_i - \lambda_s) \\
&\leqslant (\lambda_i - \lambda_n) \sum_{s=i+1}^{n} \gamma_s^2\, \widehat{p}(\lambda_s)^2, \qquad (7.124)
\end{aligned}$$

其中不等式 (7.123) 用到了 $\widehat{p}(\lambda_i) = 1$.

将式 (7.122)、式 (7.123) 和式 (7.124) 结合, 有

$$
\begin{aligned}
0 \leqslant \lambda_i - \mu_i &\leqslant \frac{\widehat{\boldsymbol{u}}^{\mathrm{T}}(\lambda_i \boldsymbol{I} - \boldsymbol{A})\widehat{\boldsymbol{u}}}{\widehat{\boldsymbol{u}}^{\mathrm{T}}\widehat{\boldsymbol{u}}^{\mathrm{T}}} \\
&\leqslant \frac{\lambda_i - \lambda_n}{\gamma_i^2} \sum_{s=i+1}^{n} \gamma_s^2 \, \widehat{p}(\lambda_s)^2 \\
&\leqslant \frac{\lambda_i - \lambda_n}{\gamma_i^2} \max_{i+1 \leqslant s \leqslant n} \widehat{p}(\lambda_s)^2 \sum_{s=i+1}^{n} \gamma_s^2. \qquad (7.125)
\end{aligned}
$$

此外, 由 $\widehat{p}(t)$ 的定义, 有

$$
\begin{aligned}
\max_{i+1 \leqslant s \leqslant n} |\widehat{p}(\lambda_s)| &\leqslant \max_{i+1 \leqslant s \leqslant n} |g(\lambda_s)| \max_{i+1 \leqslant s \leqslant n} |h(\lambda_s)| \\
&\leqslant \frac{\zeta_i}{C_{k-i}(1 + 2\delta_i)}. \qquad (7.126)
\end{aligned}
$$

将式 (7.126) 代入式 (7.125), 并注意到

$$\frac{1}{\gamma_i^2}\sum_{s=i+1}^{n}\gamma_s^2 \leqslant \frac{1}{\gamma_i^2}\left(\sum_{s=1}^{i-1}\gamma_s^2+\sum_{s=i+1}^{n}\gamma_s^2\right)$$

$$= \frac{\|(\boldsymbol{I}-\boldsymbol{P}_i)\boldsymbol{v}\|_2^2}{\|\boldsymbol{P}_i\boldsymbol{v}\|_2^2} = \tan^2\theta(\boldsymbol{u}_i,\boldsymbol{v}),$$

便知式 (7.114) 成立. 证毕. □

3. 重开始 Lanczos 方法

从理论上来讲, 由 Lanczos 算法产生的矩阵 $\boldsymbol{V}_k$ 的列向量是相互正交的. 然而, 当其在计算机上运行时, 产生的 $\boldsymbol{V}_k$ 很快就失去了其正交性. 正交性的损失会导致 Ritz 值的收敛发生紊乱现象: 会有多个 Ritz 值收敛到同一个特征值, 或者有的特征值迟迟没有逼近它的 Ritz 值出现. 如果并不关心特征值的重数, 又能忍受长时间的等待, 则正交性的损失并不影响所要达到的目的. 但如果对特

征值精确的重数很感兴趣, 或者对计算时间要求很高, 则就必须去弥补 $\boldsymbol{V}_k$ 的正交性损失. 目前已有各种各样的重正交化技术来达到这一目标, 最直接的方法就是采用算法 ?? 后所述的完全重正交化技术. 这样做的结果是 $\boldsymbol{V}_k$ 的正交性可以达到机器精度, 但为此付出的代价是:

(1) 存储量从 $O(n)$ 增加到 $O(kn)$.

(2) 计算量从 $O(kn)$ 增加到 $O(k^2n)$.

由于计算机存储空间的限制, k 不能无限制地增加, 这样就会出现 k 已经达到了最大的可能, 但 $\boldsymbol{T}_k$ 中还没有足够的信息来解决希望解决的问题.

解决这一问题的方法主要有两种: 一种是放弃完全重正交化, 而采用有选择性的重正交化技术, 从而减少存储量的要求; 另一种是使用重开始技术.

假定已经得到了一个长度为 k 的 Lanczos 分解

$$\boldsymbol{A}\boldsymbol{V}_k = \boldsymbol{V}_k\boldsymbol{T}_k + \beta_k \boldsymbol{v}_{k+1}\boldsymbol{e}_k^{\mathrm{T}}, \tag{7.127}$$

而希望求的特征值是 $\widehat{m}$ 个 (如 $\widehat{m}$ 个最大特征值, 或者 $\widehat{m}$ 个最小特征值). 现在取一个正整数 m 略大于 $\widehat{m}$, 如 $m = \widehat{m} + 2$. k 常取作 $2m$ 或 $5m$ 等.

下面介绍最近由 Stewart 给出的 Krylov–Schur 重开始方法, 其基本步骤如下.

第 1 步, 计算 $\boldsymbol{T}_k$ 的谱分解 $\boldsymbol{T}_k = \boldsymbol{Y}\boldsymbol{M}\boldsymbol{Y}^{\mathrm{T}}$, 其中

$$\boldsymbol{Y} = [\boldsymbol{y}_1, \boldsymbol{y}_2, \cdots, \boldsymbol{y}_k], \quad \boldsymbol{M} = \mathrm{diag}(\mu_1, \mu_2, \cdots, \mu_k),$$

然后适当地调整特征值 μ_i 的次序, 使得 $\mu_{m+1}, \cdots, \mu_k$ 是与要计算的特征值差距较大者, 希望将其剔除. 例如, 要计算 $\boldsymbol{A}$ 的几个最大




特征值, 则将 μ_i 排序为

$$\mu_1 \geqslant \mu_2 \geqslant \cdots \geqslant \mu_m \geqslant \cdots \geqslant \mu_k$$

即可.

第 2 步, 将 $\boldsymbol{T}_k$ 的谱分解代入式 (7.127), 并右乘 $\boldsymbol{Y}$, 得

$$\boldsymbol{A}(\boldsymbol{V}_k\boldsymbol{Y}) = (\boldsymbol{V}_k\boldsymbol{Y})\boldsymbol{M} + \beta_k\boldsymbol{v}_{k+1}\boldsymbol{e}_k^{\mathrm{T}}\boldsymbol{Y}.$$

比较上式两边的前 m 列, 得

$$\boldsymbol{A}\widetilde{\boldsymbol{V}}_m = \widetilde{\boldsymbol{V}}_m\boldsymbol{M}_1 + \beta_k\boldsymbol{v}_{k+1}\boldsymbol{z}_m^{\mathrm{T}}, \tag{7.128}$$

式中:

$$\widetilde{\boldsymbol{V}}_m = \boldsymbol{V}_k\boldsymbol{Y}(:, 1:m), \quad \boldsymbol{z}_m^{\mathrm{T}} = \boldsymbol{Y}(k, 1:m), \quad \boldsymbol{M}_1 = \mathrm{diag}(\mu_1, \cdots, \mu_m).$$

第 3 步, 计算一个正交矩阵 $\boldsymbol{Q} \in \mathbb{R}^{m\times m}$, 使得

$$\boldsymbol{Q}\boldsymbol{M}_1\boldsymbol{Q}^{\mathrm{T}} = \widehat{\boldsymbol{T}}_m, \quad \boldsymbol{Q}\boldsymbol{z}_m = \alpha\boldsymbol{e}_m, \tag{7.129}$$

式中: $\widehat{\boldsymbol{T}}_m$ 为对称三对角矩阵; $\alpha = \|\boldsymbol{z}_m\|_2$. 则在式 (7.128) 两边右乘 $\boldsymbol{Q}^{\mathrm{T}}$, 可得如下长度为 m 的 Lanczos 分解:

$$\boldsymbol{A}\widehat{\boldsymbol{V}}_m = \widehat{\boldsymbol{V}}_m\widehat{\boldsymbol{T}}_m + \widehat{\beta}_m\boldsymbol{v}_{m+1}\boldsymbol{e}_m^{\mathrm{T}}, \tag{7.130}$$

式中:

$$\widehat{\boldsymbol{V}}_m = \widetilde{\boldsymbol{V}}_m\boldsymbol{Q}^{\mathrm{T}}, \quad \widehat{\beta}_m = \alpha\beta_k, \quad \boldsymbol{v}_{m+1} = \boldsymbol{v}_{k+1}.$$

然后从式 (7.130) 出发, 再应用重正交化的 Lanczos 迭代, 使其扩展为一个长度为 k 的 Lanczos 分解. 如果仍不满足要求, 则可再用上面介绍的方法将其收缩为一个长度为 m 的 Lanczos 分解. 如此反复进行, 最终会得到一个 $\widehat{\boldsymbol{T}}_m$, 它的特征值含有所需要的特征值的很好的近似.

现在的问题是如何实现式 (7.129). 这一计算任务的具体计算过程可从下面的示例中明白. 例如 $m = 5$, 且已经将 $\boldsymbol{M}_1$ 和 $\boldsymbol{z}_m$ 约




化为如下形式, 即

$$\boldsymbol{M}_1 \to \widetilde{\boldsymbol{M}}_1 = \begin{bmatrix} \alpha_1 & \beta_1 & & & \\ \beta_1 & \alpha_2 & \beta_2 & & \\ & \beta_2 & \alpha_3 & & \\ & & & \alpha_4 & \\ & & & & \alpha_5 \end{bmatrix}, \quad \boldsymbol{z}_m \to \widetilde{\boldsymbol{z}}_m = \begin{bmatrix} 0 \\ 0 \\ \widetilde{z}_3 \\ z_4 \\ z_5 \end{bmatrix}.$$

下一步, 先确定一个 Givens 变换 $\boldsymbol{G}_3 = \boldsymbol{G}(3, 4; c_3, s_3)$, 使得 $\widetilde{\boldsymbol{z}}_m$ 的第 3 个分量为零, 然后计算 $\widehat{\boldsymbol{z}}_m = \boldsymbol{G}_3 \widetilde{\boldsymbol{z}}_m$ 和 $\widehat{\boldsymbol{M}}_1 = \boldsymbol{G}_3 \widetilde{\boldsymbol{M}}_1 \boldsymbol{G}_3^{\mathrm{T}}$, 则 $\widehat{\boldsymbol{z}}_m$ 和 $\widehat{\boldsymbol{M}}_1$ 有如下形式, 即

$$\widehat{\boldsymbol{M}}_1 = \begin{bmatrix} \alpha_1 & \beta_1 & & & \\ \beta_1 & \alpha_2 & \beta_2 & \gamma & \\ & \beta_2 & \alpha_3 & \beta_3 & \\ & \gamma & \beta_3 & \alpha_4 & \\ & & & & \alpha_5 \end{bmatrix}, \quad \widehat{\boldsymbol{z}}_m = \begin{bmatrix} 0 \\ 0 \\ 0 \\ \widetilde{z}_4 \\ z_5 \end{bmatrix}.$$

在 $\widehat{\boldsymbol{M}}_1$ 的 (2,4) 和 (4,2) 位置上出现了两个不希望有的可能非零元 "γ". 然后, 就可以采用类似于隐式对称 QR 方法中使用的技术,




计算 (2,3) 平面和 (1,2) 平面的两个 Givens 变换

$$\boldsymbol{G}_2 = \boldsymbol{G}(2,3;c_2,s_2) \quad 和 \quad \boldsymbol{G}_1 = \boldsymbol{G}(1,2;c_1,s_1),$$

将这两个不需要的元素消去, 即

$$\widehat{M}_1 \xrightarrow{G_2} \begin{bmatrix} \alpha_1 & \beta_1 & \gamma & & \\ \beta_1 & \alpha_2 & \beta_2 & & \\ \gamma & \beta_2 & \alpha_3 & \beta_3 & \\ & & \beta_3 & \alpha_4 & \\ & & & & \alpha_5 \end{bmatrix} \xrightarrow{G_1} \begin{bmatrix} \alpha_1 & \beta_1 & & & \\ \beta_1 & \alpha_2 & \beta_2 & & \\ & \beta_2 & \alpha_3 & \beta_3 & \\ & & \beta_3 & \alpha_4 & \\ & & & & \alpha_5 \end{bmatrix}.$$

再注意到 $\boldsymbol{G}_1\boldsymbol{G}_2\widehat{\boldsymbol{z}}_m = \widehat{\boldsymbol{z}}_m = \widehat{\boldsymbol{z}}_m$, 就完成了一步约化, 即在 $\boldsymbol{z}_m$ 上又多引进了一个零元.

具体的算法可总结如下.

算法 7.14 给定 m 阶对角矩阵 $\boldsymbol{M} = \mathrm{diag}(\mu_1, \cdots, \mu_m)$ 和 m 维向量 $\boldsymbol{z}$. 本算法计算正交矩阵 $\boldsymbol{Q}$, 对称三对角矩阵 $\boldsymbol{T}$ 和实数 α, 使得 $\boldsymbol{Q}\boldsymbol{z} = \tau \boldsymbol{e}_m$, $\boldsymbol{Q}\boldsymbol{M}\boldsymbol{Q}^{\mathrm{T}} = \boldsymbol{T}$.

function $[\boldsymbol{Q}, \boldsymbol{T}, \tau] = \mathbf{TriReduce}(\boldsymbol{M}, \boldsymbol{z})$

$\alpha_i = \mu_i,\ i = 1, 2, \cdots, m;$

$\beta_i = 0,\ i = 1, 2, \cdots, m-1;$

$\boldsymbol{Q} = \boldsymbol{I}_m;$

for $i = 1 : m-1$

确定 $c = \cos\theta$ 和 $s = \sin\theta$, 使得

$$\begin{bmatrix} c & s \\ -s & c \end{bmatrix} \begin{bmatrix} z_i \\ z_{i+1} \end{bmatrix} = \begin{bmatrix} 0 \\ \delta \end{bmatrix}; \quad \begin{bmatrix} z_i \\ z_{i+1} \end{bmatrix} := \begin{bmatrix} 0 \\ \delta \end{bmatrix};$$

Back

Close

$$\begin{bmatrix} \alpha_i & \beta_i \\ \beta_i & \alpha_{i+1} \end{bmatrix} := \begin{bmatrix} c & s \\ -s & c \end{bmatrix} \begin{bmatrix} \alpha_i & \beta_i \\ \beta_i & \alpha_{i+1} \end{bmatrix} \begin{bmatrix} c & s \\ -s & c \end{bmatrix}^{\mathrm{T}};$$

$\gamma := -s\beta_{i-1};\ \beta_{i-1} := c\beta_{i-1};$

$\boldsymbol{Q} := \boldsymbol{G}(i, i+1; c, s)\boldsymbol{Q};$

if $i > 1$

for $r = i : -1 : 2$

确定 $c = \cos\theta$ 和 $s = \sin\theta$, 使得

$$\begin{bmatrix} c & s \\ -s & c \end{bmatrix} \begin{bmatrix} \gamma \\ \beta_r \end{bmatrix} = \begin{bmatrix} 0 \\ \sigma \end{bmatrix};\ \beta_r := \sigma;$$

$$\begin{bmatrix} \alpha_{r-1} & \beta_{r-1} \\ \beta_{r-1} & \alpha_r \end{bmatrix} := \begin{bmatrix} c & s \\ -s & c \end{bmatrix} \begin{bmatrix} \alpha_{r-1} & \beta_{r-1} \\ \beta_{r-1} & \alpha_r \end{bmatrix} \begin{bmatrix} c & s \\ -s & c \end{bmatrix}^{\mathrm{T}};$$

Back

Close

if $r > 2$

$\gamma := -s\beta_{r-2};\ \beta_{r-2} := c\beta_{r-2};$

end

$\boldsymbol{Q} := \boldsymbol{G}(r-1, r; c, s)\boldsymbol{Q};$

end

end

end

$\tau = z_m;$

注 7.11 需要注意的是, 这里对于对称三对角矩阵 $\boldsymbol{T}$, 只存储其对角元和次对角元: $\alpha_1, \alpha_2, \cdots, \alpha_m;\ \beta_1, \beta_2, \cdots, \beta_{m-1}$.

算法 7.14 **的** MATLAB **程序如下**:

```
function [Q,alpha,beta,tau]=trireduce(M,z)
```

```
%输入:M为m阶对角矩阵,z为m维向量
%输出:Q为正交矩阵,alpha为对称三对角矩阵T的对角元,
%      beta为对称三对角矩阵T的次对角元,
%      使得Qz=tau*e_m,QMQ'=T
m=size(M,1);
for i=1:m
    alpha(i)=M(i,i);
end
beta=zeros(m-1,1); Q=eye(m);
for i=1:m-1
    delta=sqrt(z(i)^2+z(i+1)^2);
    c=z(i+1)/delta; s=-z(i)/delta;
    z(i+1)=delta; z(i)=0;
```

Back

Close

```
a=[c,s;-s,c]*[alpha(i),beta(i);
beta(i),alpha(i+1)]*[c,s;-s,c]';
alpha(i)=a(1,1); beta(i)=a(1,2); alpha(i+1)=a(2,2);
Q(i:i+1,:)=[c,s;-s,c]*Q(i:i+1,:);
if i>1
    gamma=-s*beta(i-1); beta(i-1)=c*beta(i-1);
    for r=i:-1:2
        sigma=sqrt(gamma^2+beta(r)^2);
        c=beta(r)/sigma; s=-gamma/sigma;
        a=[c,s;-s,c]*[alpha(r-1),beta(r-1);
        beta(r-1),alpha(r)]*[c,s;-s,c]';
        alpha(r-1)=a(1,1); beta(r-1)=a(1,2);
        alpha(r)=a(2,2); beta(r)=sigma;
```

```
            if r>2
                gamma=-s*beta(r-2);
                beta(r-2)=c*beta(r-2);
            end
            Q(r-1:r,:)=[c,s;-s,c]*Q(r-1:r,:);
        end
    end
end
tau=z(m);
```

在上面的讨论中, 已经提到了用 Lanczos 过程将长度为 m 的 Lanczos 分解扩展到长度为 k 的 Lanczos 分解, 下面算法给出了一种实现方法.

算法 7.15 给定对称矩阵 $\boldsymbol{A}$ 的长度为 m 的 Lanczos 分解

$$\boldsymbol{A}\boldsymbol{V}_m = \boldsymbol{V}_m\boldsymbol{T}_m + \beta_m\boldsymbol{v}_{m+1}\boldsymbol{e}_m^{\mathrm{T}} = \boldsymbol{V}_{m+1}\widehat{\boldsymbol{T}}_m,$$

本算法将其扩展为一个长度为 k 的 Lanczos 分解.

function $[\boldsymbol{V}_{k+1}, \widehat{\boldsymbol{T}}_k] = \mathbf{ExpandLan}(\boldsymbol{A}, \boldsymbol{V}_{m+1}, \widehat{\boldsymbol{T}}_m, m, k)$

for $i = m+1 : k$

$\boldsymbol{u} = \boldsymbol{A}\boldsymbol{v}_i - \beta_{i-1}\boldsymbol{v}_{i-1};\ \alpha_i = \boldsymbol{u}^{\mathrm{T}}\boldsymbol{v}_i;$

$\boldsymbol{u} = \boldsymbol{u} - \alpha_i\boldsymbol{v}_i;$

$\boldsymbol{u} = \boldsymbol{u} - (\boldsymbol{v}_{i-1}^{\mathrm{T}}\boldsymbol{u})\boldsymbol{v}_{i-1};$ (局部重正交化)

$\boldsymbol{u} = \boldsymbol{u} - (\boldsymbol{v}_i^{\mathrm{T}}\boldsymbol{u})\boldsymbol{v}_i;$

$\boldsymbol{u} = \boldsymbol{u} - \boldsymbol{V}_i(\boldsymbol{V}_i^{\mathrm{T}}\boldsymbol{u});$ (完全重正交化)

$\beta_i = \|\boldsymbol{u}\|_2;$

if $\beta_i = 0$

stop; (已经得到了 $\boldsymbol{A}$ 的一个不变子空间)

else

$\boldsymbol{v}_{i+1} = \boldsymbol{u}/\beta_i$;

$\boldsymbol{V}_{i+1} = [\boldsymbol{V}_i, \boldsymbol{v}_{i+1}]$;

end

end

算法 7.15 **的** MATLAB **程序如下**:

```
function [V,alpha,beta]=expand_lanczos(A,V,alpha,beta,m,k)
%给定对称矩阵A的长度为m的Lanczos分解,
%本算法将其扩展为一个长度为k的Lanczos分解
for i=m+1:k
    u=A*V(:,i)-beta(i-1)*V(:,i-1);
    alpha(i)=u'*V(:,i); u=u-alpha(i)*V(:,i);
```

```
        u=u-(u'*V(:,i-1))*V(:,i-1); %局部重正交化
        u=u-(u'*V(:,i))*V(:,i);
        u=u-V(:,1:i)*(V(:,1:i)'*u); %完全重正交化
        beta(i)=norm(u);
        if (beta(i)==0)
            return;
        else
            V(:,i+1)=u/beta(i);
            V=[V, V(:,i+1)];
        end
    end
```

综合上面的讨论, 就得到了如下的隐式重开始完全重正交化 Lanczos 方法.




算法 7.16 (隐式重开始 Lanczos 方法) 给定一个大型稀疏的对称矩阵 $\boldsymbol{A} \in \mathbb{R}^{n\times n}$. 本算法计算 $\boldsymbol{A}$ 的 $\widehat{m}$ 个两端特征对 $(\widehat{m} \ll n)$.

步 1, 初始化

(1) $m = \widehat{m} + 2$; $k = 2m$ (或者 $5m$, 或者取存储所能允许的最大整数).

(2) 选取一个适当的非零向量 $\boldsymbol{v} \in \mathbb{R}^n$ (一般随机选取).

(3) 计算

$$\boldsymbol{v}_1 = \boldsymbol{v}/\|\boldsymbol{v}\|_2; \quad \boldsymbol{u} = \boldsymbol{A}\boldsymbol{v}_1; \quad \alpha_1 = \boldsymbol{u}^{\mathrm{T}}\boldsymbol{v}_1;$$

$$\boldsymbol{u} = \boldsymbol{u} - \alpha_1\boldsymbol{v}_1; \quad \boldsymbol{u} = \boldsymbol{u} - (\boldsymbol{u}^{\mathrm{T}}\boldsymbol{v}_1)\boldsymbol{v}_1;$$

$$\beta_1 = \|\boldsymbol{u}\|_2; \quad \boldsymbol{v}_2 = \boldsymbol{u}/\beta_1; \quad \boldsymbol{V}_2 = [\boldsymbol{v}_1, \boldsymbol{v}_2].$$

(4) $[\boldsymbol{V}_{m+1}, \widehat{\boldsymbol{T}}_m] = \mathbf{ExpandLan}(\boldsymbol{A}, \boldsymbol{V}_2, \widehat{\boldsymbol{T}}_1, 1, m)$.

步 2, 迭代

(1) $[\boldsymbol{V}_{k+1}, \widehat{\boldsymbol{T}}_k] = \mathbf{ExpandLan}(\boldsymbol{A}, \boldsymbol{V}_{m+1}, \widehat{\boldsymbol{T}}_m, m, k)$.

(2) 用对称 QR 方法 (或其他方法) 计算 $\boldsymbol{T}_k$ 的谱分解:

$$\boldsymbol{T}_k = \boldsymbol{Y}\boldsymbol{M}\boldsymbol{Y}^{\mathrm{T}}, \quad \boldsymbol{M} = \mathrm{diag}(\mu_1, \cdots, \mu_k), \quad \boldsymbol{Y}^{\mathrm{T}}\boldsymbol{Y} = \boldsymbol{I}_k.$$

(3) 若 $\boldsymbol{T}_k$ 的特征值中已含有所求特征值的很好近似, 则输出有关消息, 结束;

否则适当调整 μ_i 的顺序, 使得 $\boldsymbol{T}_k$ 的两端特征值位于前面, 而后 $k-m$ 个特征值

为需要剔除的对象.

(4) $[\boldsymbol{Q}, \boldsymbol{T}_m, \alpha] = \mathbf{TriReduce}(\boldsymbol{M}_1, \boldsymbol{z})$, 其中

$$\boldsymbol{M}_1 = \mathrm{diag}(\mu_1, \cdots, \mu_k), \quad \boldsymbol{z}^{\mathrm{T}} = \boldsymbol{Y}(k, 1:m).$$

(5) $\boldsymbol{V}_{m+1} = [\boldsymbol{V}_k \boldsymbol{Y}(1:k,1;m)\boldsymbol{Q}^{\mathrm{T}}, \boldsymbol{v}_{k+1}]$; $\beta_m = \alpha\beta_k$. 然后, 转步 2.

注 7.12 假如希望计算的是 $\boldsymbol{A}$ 之位于某一实数 μ 附近的几个特征对, 则需要应用上一节最后所介绍的位移求逆技术, 即应用算法 7.16 到矩阵 $(\boldsymbol{A}-\mu\boldsymbol{I})^{-1}$ 上.

算法 7.16 的 MATLAB 程序如下:

```
function [M,V,iter]=irest_Lanczos(A,m)
%给定一个大型稀疏的对称矩阵A,
%本算法计算A的m个两端特征对(m<<n).
m=m+2; k=2*m; V=[ ]; tol=1.0e-30; iter=0;
v=rand(size(A,1),1); v=v/norm(v); V=[V,v];
u=A*v; alpha(1)=u'*v;
u=u-alpha(1)*v; u=u-(u'*v)*v;
```

```
beta(1)=norm(u); v=u/beta(1); V=[V,v];
[V,alpha,beta]=expand_lanczos(A,V,alpha,beta,1,m);
while(iter<1000)
  iter=iter+1;
  [V,alpha,beta]=expand_lanczos(A,V,alpha,beta,m,k);
  betak=beta(end);  %将beta的最后一个分量存起来
  vk1=V(:,end);  %将V的最后一列存起来
  T=diag(alpha)+diag(beta(1:end-1),1)+diag(beta(1:end-1),-1)
  [Y,M]=eig(T);  %M的对角元按升序按列
  [M,I]=sort(diag(M),'descend');  %对M的对角元按降序按列
  M=diag(M); Y=Y(:,I);  %第k列到了第1列
  V=V(:,1:k)*Y(:,1:m); z=Y(k,1:m)'; M=M(1:m,1:m);
  if abs(beta(end))*abs(Y(k,1))<tol
```

```
    break;
  end
  [V,alpha,beta,theta]=trireduce2(M,z);
  V=V*V'; V=[V,vk1];
  beta(m)=theta*betak;
end
```

例 7.13 用算法 7.16 计算矩阵

$$\boldsymbol{A}=\begin{bmatrix} 1 & 1 & 1 & \cdots & 1 \\ 1 & 2 & 2 & \cdots & 2 \\ 1 & 2 & 3 & \cdots & 3 \\ \vdots & \vdots & \vdots & \ddots & \vdots \\ 1 & 2 & 3 & \cdots & n \end{bmatrix} \tag{7.131}$$

的 m 个模最大的特征值. 取 $n=2000$, $m=5$.

解 利用算法 7.16, 编写 MATLAB 脚本文件 ex713.m,

```
%例7.13-ex713.m
clear all
n=2000; m=5; A=zeros(n,n);
for i=1:n
    for j=1:i
        A(i,j)=j; A(j,i)=A(i,j);
    end
end
ev=eig(A); ev=sort(ev,'descend'); lam=ev(1:m);
[M,V,iter]=irest_Lanczos(A,m); iter
mu=diag(M(1:m,1:m))
err1=norm(mu-lam)
```

```
err2=norm(A*V(:,1)-M(1,1)*V(:,1))
```

然后在 MATLAB 命令窗口运行之, 即得所要求的结果.

§7.6.3 Arnoldi 方法

Arnoldi 方法是最经典的一类 Krylov 子空间法, 它主要用于计算一个大型稀疏非对称矩阵的少数几个特征值和对应的特征向量. 随着重开始技术的不断完善, Arnoldi 方法的应用范围变得越来越广. 本节将阐述这些重开始技术以及与此相结合所产生的重开始 Arnoldi 方法.

1. *经典 Arnoldi 方法*

设 $\boldsymbol{A} \in \mathbb{R}^{n\times n}$ 已经给定, 希望计算的是 $\boldsymbol{A}$ 在某一指定范围内的少数几个特征值及对应的特征向量, 其中 $\boldsymbol{A}$ 是大型的稀疏矩阵.




将算法 2.12 与 Rayleigh–Ritz 方法相结合就可得到完成这一任务的 Arnoldi 算法.

算法 7.17 (经典 Arnoldi 方法) 给定一个 n 阶实矩阵 $\boldsymbol{A}$. 本算法计算 $\boldsymbol{A}$ 的少数几个特征对.

步 1, 选择初始向量 $\boldsymbol{v}$, 并令 $\boldsymbol{v}_1 = \boldsymbol{v}/\|\boldsymbol{v}\|_2$.

步 2, 利用算法 2.12 产生一个长度为 k 的 Arnoldi 分解, 即

$$\boldsymbol{A}\boldsymbol{V}_k = \boldsymbol{V}_k\boldsymbol{H}_k + \beta_k\boldsymbol{v}_{k+1}\boldsymbol{e}_k^{\mathrm{T}}.$$

步 3, 计算 $\boldsymbol{H}_k$ 的特征值, 并选择其中若干个满足要求的记为 $\mu_1, \cdots, \mu_l$.

步 4, 计算 μ_i 所对应的特征向量 $\boldsymbol{y}_i$, 并形成 Ritz 向量 $\boldsymbol{u}_i = \boldsymbol{V}_k\boldsymbol{y}_i, i = 1, 2, \cdots, l$.

步 5, 如果不满足要求, 增加 k, 再返回到步 2, 重复以上各步.




注 7.13 算法 7.17 只涉及矩阵乘向量 $\boldsymbol{y}=\boldsymbol{A}\boldsymbol{x}$, 因此可充分利用 $\boldsymbol{A}$ 的稀疏性 $\boldsymbol{A}$ 所具有的特殊结构.

注 7.14 设 $\boldsymbol{H}_k\boldsymbol{y}=\mu\boldsymbol{y}$, 在 Arnoldi 分解的两边右乘 $\boldsymbol{y}$, 得

$$\boldsymbol{A}\boldsymbol{V}_k\boldsymbol{y}=\mu\boldsymbol{V}_k\boldsymbol{y}+\beta_k\boldsymbol{v}_{k+1}\boldsymbol{e}_k^{\mathrm{T}}\boldsymbol{y},$$

于是有

$$\|\boldsymbol{A}(\boldsymbol{V}_k\boldsymbol{y})-\mu(\boldsymbol{V}_k\boldsymbol{y})\|_2=|\beta_k|\cdot|\boldsymbol{e}_k^{\mathrm{T}}\boldsymbol{y}|. \tag{7.132}$$

由此可知, 可以由 $|\beta_k|\cdot|\boldsymbol{e}_k^{\mathrm{T}}\boldsymbol{y}|$ 的大小来决定 $(\mu,\boldsymbol{V}_k\boldsymbol{y})$ 是否可以作为 $\boldsymbol{A}$ 的近似特征对.

为了对算法 7.17 的数值特性有个直观的了解, 下面看一个简单的数值例子.

例 7.14 令

$$\boldsymbol{A}=\boldsymbol{P}^{-1}\mathrm{diag}(-24,-23,\cdots,24,25)\boldsymbol{P}\in\mathbb{R}^{50\times 50},$$




式中: $\boldsymbol{P}$ 为随机产生的可逆矩阵.

对于 k 从 1 到 30, 应用算法 7.17 于给定的矩阵 $\boldsymbol{A}$ 上, 并且将计算结果标注在了图 7.5 中, 其中横坐标表示迭代步数 k, 第 k 列的点表示 $\boldsymbol{H}_k$ 的所有实特征值, 最后一列的点表示 $\boldsymbol{A}$ 的精确特征值.

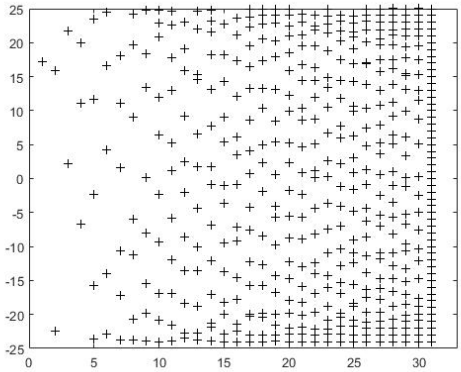

图 7.5 Arnoldi 算法的收敛特性

请注意观察该例所展示出的算法 7.17 的收敛特性: $k=10$ 时, $\boldsymbol{H}_{10}$ 就有两个最大特征值收敛到了 $\boldsymbol{A}$ 的两个最大特征值; 而到了 $k=21$ 时, $\boldsymbol{H}_{21}$ 就有 3 个最小特征值和 2 个最大特征值分别收敛到了 $\boldsymbol{A}$ 的 3 个最小特征值和 2 个最大特征值; 而位于区间内部的有些特征值, 直到 $k=30$ 还没有 Ritz 值收敛到它们.

虽然例 7.14 仅仅是一个可对角化的特殊矩阵, 但其所展示出的算法 7.17 的收敛特性对一般情形也是成立的, 即外围特征值 (exterior eigenvalues) 最先收敛, 而内部特征值收敛特别缓慢. 事实上, 这一结果也可以从理论上解释如下.

为了叙述简洁起见, 假设 $\boldsymbol{A}$ 是可对角化的. 设 $\boldsymbol{A}$ 的特征对为

$$(\lambda_j, \boldsymbol{x}_j), \quad j=1,2,\cdots,n,$$

则 $\boldsymbol{A}$ 可对角化的假定蕴含着 $\boldsymbol{x}_1, \boldsymbol{x}_2, \cdots, \boldsymbol{x}_n$ 是线性无关的. 现将初始向量 $\boldsymbol{v}$ 按特征向量展开:

$$\boldsymbol{v} = \gamma_1 \boldsymbol{x}_1 + \gamma_2 \boldsymbol{x}_2 + \cdots + \gamma_n \boldsymbol{x}_n.$$

根据定理 2.10 的结论 (5), $\boldsymbol{u} \in \mathcal{K}_k(\boldsymbol{A}, \boldsymbol{v})$ 的充分必要条件是存在 $p \in \mathcal{P}_{k-1}$, 使得 $\boldsymbol{u} = p(\boldsymbol{A})\boldsymbol{v}$, 从而有

$$\boldsymbol{u} = \gamma_1 p(\lambda_1)\boldsymbol{x}_1 + \gamma_2 p(\lambda_2)\boldsymbol{x}_2 + \cdots + \gamma_n p(\lambda_n)\boldsymbol{x}_n.$$

由此可知, 欲使 $\boldsymbol{u}$ 是某一特征向量 $\boldsymbol{x}_j$ 的很好近似, 则必须有 $|p(\lambda_j)|$ 相对要比其他的 $|p(\lambda_i)|$ 大得多. 而由复变函数的最大模原理可知, 这只有 λ_j 位于谱集的边缘才能达到. 这也就是 Arnoldi 方法为什么先收敛到外围特征值的缘故.

前面已经列举了 Arnoldi 算法的优点, 但它也有致命的不足之处. 这一算法随着 k 的增加其存储空间需求越来越大, 因此 k 的大小受到了计算机存储量的限制. 这样一来就会出现这样的问题: 希望计算的特征值还没有得到, 而 k 就不能再增大了. 解决这一问题的方法就是现在已经逐渐成熟的重开始技术.

Back

Close

2. 隐式重开始 Arnoldi 方法

假设希望计算 $\boldsymbol{A}$ 的 $\widehat{m}$ 个模最大的特征值 (其他情况可通过注 7.15 所述的“位移求逆技术”归结为这种情形). 再假定已得到一个长度为 k 的 Arnoldi 分解

$$\boldsymbol{A}\boldsymbol{V}_k = \boldsymbol{V}_k\boldsymbol{H}_k + \beta_k\boldsymbol{v}_{k+1}\boldsymbol{e}_k^{\mathrm{T}}, \qquad \beta_k = h_{k+1,k}. \tag{7.133}$$

由于计算机存储空间的限制, k 已经不能再增加了. 但由 $\boldsymbol{H}_k$ 所提供的 Ritz 值还不满足精度要求. 面临这样的处境, 下一步没有别的选择, 只有重新开始, 即再选择一个新的初始向量 $\widehat{\boldsymbol{v}}$, 重新再来计算新的 Arnoldi 分解. 但是并不想把前面所做的一切全部扔掉, 最好能够对重新选择 $\widehat{\boldsymbol{v}}$ 提供一些有用的信息, 使新选择的 $\widehat{\boldsymbol{v}}$ 比 $\boldsymbol{v}$ 更好一些. 这正是发展重开始技术的最原始想法.

目前存在两种隐式重开始 Arnoldi 方法: 一种是 Sorensen (索仑森) 提出的基于“过滤多项式”技术的隐式重开始方法; 另一种是 Stewart (史都瓦) 提出的基于实 Schur 分解的所谓 Krylov–Schur 重开始方法. 在此着重介绍后者, 其基本步骤有如下 3 步:

第 1 步, 计算 $\boldsymbol{H}_k$ 的实 Schur 分解: $\boldsymbol{H}_k = \boldsymbol{U}_1 \boldsymbol{R} \boldsymbol{U}_1^{\mathrm{T}}$, 其中 $\boldsymbol{U}_1 \in \mathbb{R}^{k\times k}$ 是正交矩阵, $\boldsymbol{R} \in \mathbb{R}^{k\times k}$ 是拟上三角矩阵 (即 $\boldsymbol{R}$ 是分块上三角矩阵, 其对角块是 1×1 或 2×2). 这一步可由著名的 QR 方法实现.

第 2 步, 重排 $\boldsymbol{R}$ 的对角块, 即计算一个正交矩阵 $\boldsymbol{U}_2 \in \mathbb{R}^{k\times k}$, 使得

$$\boldsymbol{U}_2^{\mathrm{T}} \boldsymbol{R} \boldsymbol{U}_2 = \begin{bmatrix} \boldsymbol{R}_1 & * \\ \boldsymbol{O} & \boldsymbol{R}_2 \end{bmatrix},$$

式中: $\boldsymbol{R}_1$ 和 $\boldsymbol{R}_2$ 均为实 Schur 标准形, 而且

$$\lambda(\boldsymbol{R}_1) = \{\mu_1, \cdots, \mu_m\}, \quad \lambda(\boldsymbol{R}_2) = \{\mu_{m+1}, \cdots, \mu_k\}.$$

第 1 步和第 2 步计算完成之后, 有

$$\boldsymbol{H}_k = \boldsymbol{U}_1\boldsymbol{U}_2 \begin{bmatrix} \boldsymbol{R}_1 & * \\ \boldsymbol{O} & \boldsymbol{R}_2 \end{bmatrix} (\boldsymbol{U}_1\boldsymbol{U}_2)^{\mathrm{T}}.$$

将其代入式 (7.133) 并且右乘 $\boldsymbol{U}_1\boldsymbol{U}_2$, 得

$$\boldsymbol{A}(\boldsymbol{V}_k\boldsymbol{U}_1\boldsymbol{U}_2) = \boldsymbol{V}_k\boldsymbol{U}_1\boldsymbol{U}_2 \begin{bmatrix} \boldsymbol{R}_1 & * \\ \boldsymbol{O} & \boldsymbol{R}_2 \end{bmatrix} + \beta_k\boldsymbol{v}_{k+1}\boldsymbol{e}_k^{\mathrm{T}}(\boldsymbol{U}_1\boldsymbol{U}_2).$$

比较上式两边的前 m 列, 得

$$\boldsymbol{A}\overline{\boldsymbol{V}}_m = \overline{\boldsymbol{V}}_m\boldsymbol{R}_1 + \beta_k\boldsymbol{v}_{k+1}\boldsymbol{z}_m^{\mathrm{T}}, \tag{7.134}$$




式中: $\overline{\boldsymbol{V}}_m$ 为 $\boldsymbol{V}_k\boldsymbol{U}_1\boldsymbol{U}_2$ 的前 m 列构成的矩阵; $\boldsymbol{z}_m^{\mathrm{T}}$ 为 $\boldsymbol{U}_1\boldsymbol{U}_2$ 的最后一行的前 m 个元素构成的向量.

第 3 步, 计算一个正交矩阵 $\boldsymbol{Q} \in \mathbb{R}^{m\times m}$, 使得

$$\boldsymbol{Q}^{\mathrm{T}}\boldsymbol{R}_1\boldsymbol{Q} = \widehat{\boldsymbol{H}}_m, \quad \boldsymbol{z}_m^{\mathrm{T}}\boldsymbol{Q} = \alpha\boldsymbol{e}_m^{\mathrm{T}}, \tag{7.135}$$

式中: $\widehat{\boldsymbol{H}}_m$ 为上 Hessenberg 矩阵.

事实上, 式 (7.135) 可按如下方式实现:

(1) 首先计算一个 Householder 变换 $\boldsymbol{P}_1$, 使得

$$\boldsymbol{z}_m^{\mathrm{T}}\boldsymbol{P}_1 = \alpha\boldsymbol{e}_m^{\mathrm{T}}.$$

(2) 然后依次计算 $m-2$ 个 Householder 变换

$$\boldsymbol{P}_i \in \mathbb{R}^{(m-i-1)\times(m-i+1)}, \quad i = 2, \cdots, m-1,$$

使得

$$\boldsymbol{Q}_{m-1}^{\mathrm{T}}\cdots\boldsymbol{Q}_2^{\mathrm{T}}\boldsymbol{P}_1^{\mathrm{T}}\boldsymbol{R}_1\boldsymbol{P}_1\boldsymbol{Q}_2\cdots\boldsymbol{Q}_{m-1}=\widehat{\boldsymbol{H}}_m$$

为上 Hessenberg 矩阵, 其中 $\boldsymbol{Q}_i=\mathrm{diag}(\boldsymbol{P}_i,\boldsymbol{I}_{i-1})$.

其实, 这一约化过程本质上就是在 节中介绍的约化一个矩阵为上 Hessenberg 形的方法. 只是此处是由矩阵的行向量来确定约化所需的 Householder 变换 $\boldsymbol{P}_i$, 而且是从最后一行开始约化的; 而那里是由其列向量来确定, 而且是从第 1 列开始约化的. 显然, 这样得到的 $\boldsymbol{Q}_i$ 满足 $\boldsymbol{e}_m^{\mathrm{T}}(\boldsymbol{Q}_2\cdots\boldsymbol{Q}_{m-1})=\boldsymbol{e}_m^{\mathrm{T}}$, 故令 $\boldsymbol{Q}=\boldsymbol{P}_1\boldsymbol{Q}_2\cdots\boldsymbol{Q}_{m-1}$, 则该正交矩阵就是式 (7.135) 中所需的正交矩阵.

一旦式 (7.135) 已经得到, 则在式 (7.134) 两边右乘 $\boldsymbol{Q}$, 得

$$\boldsymbol{A}\widehat{\boldsymbol{V}}_m=\widehat{\boldsymbol{V}}_m\widehat{\boldsymbol{H}}_m+\widehat{\beta}_m\widehat{\boldsymbol{v}}_{m+1}\boldsymbol{e}_m^{\mathrm{T}},\tag{7.136}$$

式中: $\widehat{\boldsymbol{V}}_m=\overline{\boldsymbol{V}}_m\boldsymbol{Q}$, $\widehat{\boldsymbol{v}}_{m+1}=\boldsymbol{v}_{k+1}$, $\widehat{\beta}_m=\alpha\beta_k$.

这就是 Stewart 的 Krylov–Schur 重开始方法最终计算得到的结果: 将一个长度为 k 的 Arnoldi 分解收缩为一个长度为 m 的 Arnoldi 分解.

下面考虑 Krylov–Schur 重开始方法的收敛性问题. 为了符号简单, 将式 (7.134) 改写为

$$\boldsymbol{A}\boldsymbol{V}_m = \boldsymbol{V}_m\boldsymbol{R} + \beta_k \boldsymbol{v}_{k+1}\boldsymbol{z}^{\mathrm{T}}. \tag{7.137}$$

设

$$\boldsymbol{z} = (0, \cdots, 0, z_{\ell+1}, \cdots, z_m)^{\mathrm{T}}. \tag{7.138}$$

实际计算时, 要给定一个收敛准则, 可将 $\boldsymbol{z}$ 之小到一定程度的分量均置零. 现假定对此 ℓ, 矩阵 $\boldsymbol{R}$ 有如下的分块表示:

$$\boldsymbol{R} = \begin{bmatrix} \boldsymbol{R}_{11} & \boldsymbol{R}_{12} \\ \boldsymbol{O} & \boldsymbol{R}_{22} \end{bmatrix} \begin{matrix} \ell \\ m-\ell \end{matrix}.$$
$$\begin{matrix} \ell & m-\ell \end{matrix}$$




Back
Close

事实上, 若 $\boldsymbol{R}$ 的第 ℓ 行正好对应于它的一个 2×2 阶对角块的第 1 行, 则上述分块就不存在, 此时可取 $\ell-1$ 作为 ℓ.

再对 $\boldsymbol{V}_m$ 作相应的分块, 即

$$\boldsymbol{V}_m = \begin{bmatrix} \boldsymbol{V}_{m1} & \boldsymbol{V}_{m2} \\ \ell & m-\ell \end{bmatrix}.$$

由于 $\boldsymbol{z}$ 有式 (7.138) 的形状, 故式 (7.135) 中的正交矩阵就可取作如下形式:

$$\boldsymbol{Q} = \mathrm{diag}(\boldsymbol{I}_\ell, \boldsymbol{Q}_1),$$

式中: $\boldsymbol{Q}_1$ 为 $m-\ell$ 阶正交矩阵. 对于这样的 $\boldsymbol{Q}$, 有

$$\widehat{\boldsymbol{V}}_m = \boldsymbol{V}_m\boldsymbol{Q} = \begin{bmatrix} \boldsymbol{V}_{m1} & \boldsymbol{V}_{m2}\boldsymbol{Q}_1 \end{bmatrix},$$

$$\widehat{\boldsymbol{H}}_m = \boldsymbol{Q}^{\mathrm{T}}\boldsymbol{R}\boldsymbol{Q} = \begin{bmatrix} \boldsymbol{R}_{11} & \boldsymbol{R}_{12}\boldsymbol{Q}_1 \\ \boldsymbol{O} & \boldsymbol{Q}_1^{\mathrm{T}}\boldsymbol{R}_{22}\boldsymbol{Q}_1 \end{bmatrix}.$$

这表明, 在进行这一步变换时, $\boldsymbol{V}_m$ 的前 ℓ 列和 $\boldsymbol{R}$ 的 ℓ 阶顺序主子阵并没有改变. 在 Arnoldi 分解式 (7.136) 中 $\widehat{\boldsymbol{V}}_m$ 的前 ℓ 列 $\boldsymbol{V}_{m1}$ 就是 $\boldsymbol{A}$ 的一个不变子空间的一组标准正交基, $\widehat{\boldsymbol{H}}_m$ 的 ℓ 阶顺序主子阵 $\boldsymbol{R}_{11}$ 的特征值就全部是 $\boldsymbol{A}$ 的特征值, 而且后继部分的计算再也不会涉及 $\boldsymbol{V}_{m1}$ 和 $\boldsymbol{R}_{11}$, 因此就称它们已经被锁定. 随着迭代的进行, 锁定的部分会越来越大, 最终达成收敛.

综合上面的讨论, 可得如下的隐式重开始 Arnoldi 算法.

算法 7.18 (隐式重开始 Arnoldi 方法) 给定 n 阶实矩阵 $\boldsymbol{A}$, 正整数 k 和 $\widehat{m}\,(\widehat{m} < k)$. 本算法计算 $\boldsymbol{A}$ 的 $\widehat{m}$ 个模最大的特征值和对应的特征向量.

步 1, 初始化

(1) 选择初始向量 $\boldsymbol{v}$, 并令 $\boldsymbol{v}_1 = \boldsymbol{v}/\|\boldsymbol{v}\|_2$; $m = \widehat{m} + 2$.

(2) 应用算法 2.12 于数据 $(\boldsymbol{A}, \boldsymbol{v}_1, k)$ 上, 产生一个长度为

k 的 Arnoldi 分解

$$\boldsymbol{A}\boldsymbol{V}_k = \boldsymbol{V}_k\boldsymbol{H}_k + \beta_k\boldsymbol{v}_{k+1}\boldsymbol{e}_k^{\mathrm{T}}.$$

步 2, 收敛性判定

(1) 计算 $\boldsymbol{H}_k$ 的实 Schur 分解: $\boldsymbol{H}_k = \boldsymbol{U}\boldsymbol{R}\boldsymbol{U}^{\mathrm{T}}$, 其中 $\boldsymbol{U}$ 是正交矩阵, $\boldsymbol{R}$ 是

实 Schur 标准形.

(2) 若 $\boldsymbol{H}_k^{\mathrm{T}}$ 的 $\widehat{m}$ 个模最大的特征值已经收敛, 则计算相应 Ritz 向量, 停

算; 否则, 进行下一步.

步 3, 重开始 Arnoldi 过程

(1) 利用 Krylov-Schur 重开始方法产生一个长度为 m 的 Arnoldi 分解

$$\boldsymbol{A}\boldsymbol{V}_m = \boldsymbol{V}_m\boldsymbol{H}_m + \beta_m\boldsymbol{v}_{m+1}\boldsymbol{e}_m^{\mathrm{T}}.$$

Back

Close

(2) 利用 Arnoldi 过程将该分解扩展为一个长度为 k 的 Arnoldi 分解

$$\boldsymbol{A}\boldsymbol{V}_k = \boldsymbol{V}_k\boldsymbol{H}_k + \beta_k \boldsymbol{v}_{k+1}\boldsymbol{e}_k^{\mathrm{T}}.$$

然后转步 2.

注 7.15 注意到算法 7.18 只能计算 $\boldsymbol{A}$ 的几个模最大的特征值和对应的特征向量. 若要计算 $\boldsymbol{A}$ 在某个数 μ 附近的几个特征值和对应的特征向量, 通常是采取位移求逆的方法, 即将算法 7.18 应用到矩阵 $(\boldsymbol{A}-\mu\boldsymbol{I})^{-1}$ 上. 一旦 $(\boldsymbol{A}-\mu\boldsymbol{I})^{-1}$ 的几个模最大的特征值 $\widehat{\lambda}_1, \cdots, \widehat{\lambda}_m$ 已经求得, 则

$$\lambda_i = \mu + 1/\widehat{\lambda}_i, \quad i = 1, 2, \cdots, m$$

就是 $\boldsymbol{A}$ 的最靠近 μ 的几个特征值.

需注意的是, 应用算法 7.18 于 $(\boldsymbol{A}-\mu\boldsymbol{I})^{-1}$ 上时, 需要计算形

如

$$\boldsymbol{y} = (\boldsymbol{A} - \mu \boldsymbol{I})^{-1} \boldsymbol{x}$$

的矩阵乘向量, 即需要求解形如

$$(\boldsymbol{A} - \mu \boldsymbol{I}) \boldsymbol{y} = \boldsymbol{x} \tag{7.139}$$

的线性方程组. 由于 Arnoldi 方法对 $\boldsymbol{y}$ 的误差十分敏感, 因此通常采用稀疏列主元的 LU 分解来求解方程组 (7.139). 当然, 在整个迭代过程中只需要作一次分解就够了.

§7.6.4 Jacobi–Davidson 方法

7.6.1 节中的 Rayleigh–Ritz 投影方法其实只是子空间迭代方法的一般框架, 它并没有给出子空间的具体选取方法. 节的 Lanzcos 方法 (矩阵 $\boldsymbol{A}$ 对称) 和 节的 Arnoldi 方法 (矩阵 $\boldsymbol{A}$ 非对称) 其实




都是将投影子空间取为 Krylov 子空间 $\mathcal{K}_k(\boldsymbol{A}, \boldsymbol{v})$ 的 Rayleigh–Ritz 投影法. 本节将给出投影子空间的另一个取法, 相应的方法称为 Jacobi–Davidson 方法.

设 $\boldsymbol{A} \in \mathbb{C}^{n\times n}$, $\mathcal{K} = \mathrm{span}\{\boldsymbol{v}_1, \boldsymbol{v}_2, \cdots, \boldsymbol{v}_m\} \subset \mathbb{C}^n$, $\{\boldsymbol{v}_i\}_{i=1}^m$ 为 $\mathcal{K}$ 中的标准正交基. 记 $\boldsymbol{V}_m = [\boldsymbol{v}_1, \boldsymbol{v}_2, \cdots, \boldsymbol{v}_m]$. 选择 $\mu \in \mathbb{C}$ 和 $\boldsymbol{x} \in \mathcal{K}$ 使得

$$(\boldsymbol{A}\boldsymbol{x} - \mu\boldsymbol{x}) \perp \mathcal{K}.$$

上述条件可以等价地表述为

$$(\boldsymbol{A}\boldsymbol{x} - \mu\boldsymbol{x}, \boldsymbol{v}) = 0, \quad \forall\, \boldsymbol{v} \in \mathcal{K}.$$

注意到 $\boldsymbol{x}$ 可表示为 $\boldsymbol{x} = \boldsymbol{V}_m\boldsymbol{y}$, 其中 $\boldsymbol{y} \in \mathbb{C}^m$, 故上述条件又可以等价地表示为

$$\boldsymbol{V}_m^{\mathrm{H}}(\boldsymbol{A}\boldsymbol{V}_m\boldsymbol{y} - \mu\boldsymbol{V}_m\boldsymbol{y}) = \boldsymbol{0},$$

即




$$\boldsymbol{A}_m\boldsymbol{y} = \mu\boldsymbol{y},$$

其中 $\boldsymbol{A}_m = \boldsymbol{V}_m^{\mathrm{H}}\boldsymbol{A}\boldsymbol{V}_m$ 正好是 $\boldsymbol{A}$ 关于 $\boldsymbol{V}_m$ 的 Rayleigh 商. 设 $(\mu, \boldsymbol{y})$ $(\|\boldsymbol{y}\|_2 = 1)$ 是 $\boldsymbol{A}_m$ 的特征对, $(\mu, \boldsymbol{V}_m\boldsymbol{y})$ 是 $\boldsymbol{A}$ 的 Ritz 对, 它是 $\boldsymbol{A}$ 的某个特征对的近似. 若记 $\boldsymbol{u} = \boldsymbol{V}_m\boldsymbol{y}$, 则显然有 $\|\boldsymbol{u}\|_2 = 1$. 现取向量 $\boldsymbol{z} \in \mathbb{C}^n$ 满足 $\boldsymbol{z}^{\mathrm{H}}\boldsymbol{u} = 0$, 且

$$\boldsymbol{A}(\boldsymbol{u} + \boldsymbol{z}) = \lambda(\boldsymbol{u} + \boldsymbol{z}) \Longleftrightarrow (\boldsymbol{A} - \lambda\boldsymbol{I})\boldsymbol{z} = -(\boldsymbol{A} - \lambda\boldsymbol{I})\boldsymbol{u}. \quad (7.140)$$

上式表明 $\boldsymbol{A}$ 的近似特征向量 $\boldsymbol{u}$ 加上一个和它正交的向量 $\boldsymbol{z}$ 后变为 $\boldsymbol{A}$ 的特征向量. 这启发把上述方程限制在与 $\boldsymbol{u}$ 正交的子空间上, 得

$$(\boldsymbol{I} - \boldsymbol{u}\boldsymbol{u}^{\mathrm{H}})(\boldsymbol{A} - \lambda\boldsymbol{I})(\boldsymbol{I} - \boldsymbol{u}\boldsymbol{u}^{\mathrm{H}})\boldsymbol{z} = -(\boldsymbol{A} - \mu\boldsymbol{I})\boldsymbol{u}. \quad (7.141)$$

事实上, 考虑到

$$\boldsymbol{u}^{\mathrm{H}}\boldsymbol{A}\boldsymbol{u} = (\boldsymbol{V}_m\boldsymbol{y})^{\mathrm{H}}\boldsymbol{A}(\boldsymbol{V}_m\boldsymbol{y}) = \boldsymbol{y}^{\mathrm{H}}\boldsymbol{A}_m\boldsymbol{y} = \mu, \quad (7.142)$$

$\boldsymbol{u}$ 与 $\boldsymbol{z}$ 正交以及式 (7.140), 有

$$\begin{aligned}
&(\boldsymbol{I}-\boldsymbol{u}\boldsymbol{u}^{\mathrm{H}})(\boldsymbol{A}-\lambda\boldsymbol{I})(\boldsymbol{I}-\boldsymbol{u}\boldsymbol{u}^{\mathrm{H}})\boldsymbol{z}\\
&\quad=(\boldsymbol{I}-\boldsymbol{u}\boldsymbol{u}^{\mathrm{H}})(\boldsymbol{A}-\lambda\boldsymbol{I})\boldsymbol{z}=-(\boldsymbol{I}-\boldsymbol{u}\boldsymbol{u}^{\mathrm{H}})(\boldsymbol{A}-\lambda\boldsymbol{I})\boldsymbol{u}\\
&\quad=-(\boldsymbol{A}-\lambda\boldsymbol{I})\boldsymbol{u}+(\boldsymbol{u}\boldsymbol{u}^{\mathrm{H}}\boldsymbol{A}\boldsymbol{u}-\lambda\boldsymbol{u})=-(\boldsymbol{A}-\mu\boldsymbol{I})\boldsymbol{u}.
\end{aligned}$$

由于式 (7.141) 中的 λ 未知, 用 μ 近似它, 得

$$\widetilde{\boldsymbol{A}}\boldsymbol{z}=-\widetilde{\boldsymbol{r}},\quad \boldsymbol{z}^{\mathrm{H}}\boldsymbol{u}=0, \tag{7.143}$$

式中:

$$\widetilde{\boldsymbol{A}}=(\boldsymbol{I}-\boldsymbol{u}\boldsymbol{u}^{\mathrm{H}})(\boldsymbol{A}-\mu\boldsymbol{I})(\boldsymbol{I}-\boldsymbol{u}\boldsymbol{u}^{\mathrm{H}})$$

为 $\boldsymbol{A}$ 在和 $\boldsymbol{u}$ 正交的子空间上的限制; $\widetilde{\boldsymbol{r}}=(\boldsymbol{A}-\mu\boldsymbol{I})\boldsymbol{u}$ 表示 Ritz 对 $(\mu,\boldsymbol{u})$ 的残差. 由式 (7.142) 可知

$$\boldsymbol{u}^{\mathrm{H}}\widetilde{\boldsymbol{r}}=\boldsymbol{u}^{\mathrm{H}}(\boldsymbol{A}-\mu\boldsymbol{I})\boldsymbol{u}=0,$$

即 $\widetilde{\boldsymbol{r}} \in \mathrm{span}\{\boldsymbol{u}\}^{\perp} = \mathcal{R}(\boldsymbol{I} - \boldsymbol{u}\boldsymbol{u}^{\mathrm{H}})$. 式 (7.143) 也可表示为

$$(\boldsymbol{A} - \mu\boldsymbol{I})\boldsymbol{z} = -\widetilde{\boldsymbol{r}} + \alpha\boldsymbol{u}, \quad \alpha = \boldsymbol{u}^{\mathrm{H}}(\boldsymbol{A} - \mu\boldsymbol{I})\boldsymbol{z}, \quad \boldsymbol{z}^{\mathrm{H}}\boldsymbol{u} = 0.$$

设 μ 不是 $\boldsymbol{A}$ 的特征值, 则有

$$\boldsymbol{z} = (\boldsymbol{A} - \mu\boldsymbol{I})^{-1}(-\widetilde{\boldsymbol{r}} + \alpha\boldsymbol{u}). \tag{7.144}$$

根据求出的 $\boldsymbol{z}$, 得

$$\boldsymbol{u}^{\mathrm{H}}(\boldsymbol{A} - \mu\boldsymbol{I})\boldsymbol{z} = \boldsymbol{u}^{\mathrm{H}}(-\widetilde{\boldsymbol{r}} + \alpha\boldsymbol{u}) = \alpha.$$

注意到 $\boldsymbol{u}$ 与 $\boldsymbol{z}$ 正交, 则由式 (7.144), 有

$$\alpha = \frac{\boldsymbol{u}^{\mathrm{H}}(\boldsymbol{A} - \mu\boldsymbol{I})^{-1}\widetilde{\boldsymbol{r}}}{\boldsymbol{u}^{\mathrm{H}}(\boldsymbol{A} - \mu\boldsymbol{I})^{-1}\boldsymbol{u}}.$$

这表明, 当 μ 不是 $\boldsymbol{A}$ 的特征值时, 方程组 (7.143) 有唯一解.




Back
Close

上面讨论了如何从一个子空间 $\mathrm{span}\{\boldsymbol{v}_1, \boldsymbol{v}_2, \cdots, \boldsymbol{v}_m\}$ 出发求 Ritz 向量 $\boldsymbol{u}$, 并把它进行校正写为 $\boldsymbol{u}+\boldsymbol{z}$, 这里 $\boldsymbol{u}$ 在该子空间中. 现在利用 $\boldsymbol{z}$ 构造一个“更大”的子空间, 取

$$\boldsymbol{v}_{m+1} = \boldsymbol{z} - \sum_{i=1}^{m} (\boldsymbol{z}^{\mathrm{H}} \boldsymbol{v}_i) \boldsymbol{v}_i$$

的单位化向量. 下面给出 Jacobi–Davidson 方法的详细算法步骤.

算法 7.19 (Jacobi–Davidson 方法) 给定 $\boldsymbol{A} \in \mathbb{C}^{n\times n}$, $\boldsymbol{v}_1 \in \mathbb{C}^n$ 满足 $\|\boldsymbol{v}_1\|_2 = 1$.

For $m = 1, 2, \cdots$ **Do**

(1) 根据 $\boldsymbol{V}_m = [\boldsymbol{v}_1, \boldsymbol{v}_2, \cdots, \boldsymbol{v}_m]$ 产生 $\boldsymbol{A}_m = \boldsymbol{V}_m^{\mathrm{H}} \boldsymbol{A} \boldsymbol{V}_m \in \mathbb{C}^{m\times m}$.

(2) 求 $\boldsymbol{A}_m$ 的某个特征对 $(\mu, \boldsymbol{y})$. 令 $\boldsymbol{u} = \boldsymbol{V}_m \boldsymbol{y}$.

Back

Close

(3) 计算 $\widetilde{\boldsymbol{r}} = \boldsymbol{A}\boldsymbol{u} - \mu\boldsymbol{u}$. 若 $\|\widetilde{\boldsymbol{r}}\|_2 \leqslant \varepsilon$, 停算.

(4) 解方程组 (7.143) 得到 $\boldsymbol{z}$.

(5) 取 $\widetilde{\boldsymbol{v}}_{m+1} := \boldsymbol{z} - \sum_{i=1}^{m}(\boldsymbol{z}^{\mathrm{H}}\boldsymbol{v}_i)\boldsymbol{v}_i$, $\boldsymbol{v}_{m+1} = \widetilde{\boldsymbol{v}}_{m+1}/\|\widetilde{\boldsymbol{v}}_{m+1}\|_2$.

Enddo

注 7.16 注意到

$$\boldsymbol{A}_{m+1} = \boldsymbol{V}_{m+1}^{\mathrm{H}}\boldsymbol{A}\boldsymbol{V}_{m+1} = \begin{bmatrix} \boldsymbol{V}_m^{\mathrm{H}} \\ \boldsymbol{v}_{m+1}^{\mathrm{H}} \end{bmatrix} \boldsymbol{A}\big[\boldsymbol{V}_m,\ \boldsymbol{v}_{m+1}\big]$$

$$= \begin{bmatrix} \boldsymbol{A}_m & \boldsymbol{V}_m^{\mathrm{H}}\boldsymbol{A}\boldsymbol{v}_{m+1} \\ \boldsymbol{v}_{m+1}^{\mathrm{H}}\boldsymbol{A}\boldsymbol{V}_m & \boldsymbol{v}_{m+1}^{\mathrm{H}}\boldsymbol{A}\boldsymbol{v}_{m+1} \end{bmatrix},$$

故算法 7.19 在第 $m+1$ 次循环时, 只需计算 $\boldsymbol{V}_m^{\mathrm{H}}\boldsymbol{A}\boldsymbol{v}_{m+1}$, $\boldsymbol{v}_{m+1}^{\mathrm{H}}\boldsymbol{A}\boldsymbol{V}_m$ 和 $\boldsymbol{v}_{m+1}^{\mathrm{H}}\boldsymbol{A}\boldsymbol{v}_{m+1}$ 便产生了 $\boldsymbol{A}_{m+1}$.

注 7.17 算法 7.19 的计算量主要集中在第 4 步解方程组 (7.143). 可以使用子空间迭代法 GMRES 方法或预处理 GMRES 方法来求解.

算法 7.19 的 MATLAB 程序如下:

```
%Jacobi-Davidson方法程序-jacobidavidson.m
function [mu,u,Vm]=jacobidavidson(A,v,tol,max_it)
%用Jacobi-Davidson求矩阵A的模最大的特征值
%及相应的特征向量
%输入:A为n阶实方阵,v为初始向量,tol为容许误差,
%      max_it为子空间的最大维数
%输出:mu返回A的模最大特征值,u为相应的特征向量,
%      Vm为子空间基矩阵
if nargin<4, max_it=ceil(size(A,1)/2); end
```

```
if nargin<3, tol=1e-6; end
n=size(A,1); m=0; Vm=[ ]; I=eye(n);
x0=rand(n,1); v=v/norm(v);
Vm=[Vm,v]; Am=Vm'*A*Vm;
while(m<max_it)
    m=m+1;
    [Y,D]=eig(Am); [s,t]=max(abs(diag(D)));
    mu=D(t,t); u=Vm*Y(:,t); r=A*u-mu*u;
    if (norm(r)<=tol), break; end
    At=(I-u*u')*(A-mu*I)*(I-u*u');
    [z]=gmres(At,-r);  %系统自带的GMRES函数
    v=z;  %正交化
    for i=1:m
```

```
        v=v-(z'*Vm(:,i))*Vm(:,i);
    end
    v=v/norm(v);
    Am=[Am,Vm'*A*v;v'*A*Vm,v'*A*v];
    Vm=[Vm,v];
end
```

例 7.15 用算法 7.19 计算矩阵

$$\boldsymbol{A}=\begin{bmatrix} 1 & 1 & 1 & \cdots & 1 \\ 1 & 2 & 2 & \cdots & 2 \\ 1 & 2 & 3 & \cdots & 3 \\ \vdots & \vdots & \vdots & \ddots & \vdots \\ 1 & 2 & 3 & \cdots & n \end{bmatrix}$$

的模最大特征值. 取 $n=1000$.

解 编写 MATLAB 脚本程序 ex715.m, 然后命令窗口运行该程序可得到相应的结果. 当投影子空间的维数增加到 $m = 4$ 时, 迭代已满足终止条件, 得到近似特征值 $\widetilde{\lambda}$ 和对应的特征向量 $\boldsymbol{u}$ 满足 $\|\boldsymbol{A}\boldsymbol{u} - \widetilde{\lambda}\boldsymbol{u}\|_2 = 5.3620 \times 10^{-9}$.

假设希望计算矩阵 $\boldsymbol{A}$ 的某个与 α 最接近的特征值, 可以考虑求 $(\boldsymbol{A} - \alpha\boldsymbol{I})^{-1}$ 模最大的特征值 μ, 那么 $\widetilde{\lambda} = \dfrac{1}{\mu} + \alpha$ 即为 $\boldsymbol{A}$ 的某个与 α 最接近的特征值. 也就是说, 只需用 $\boldsymbol{B} = (\boldsymbol{A} - \alpha\boldsymbol{I})^{-1}$ 去调用算法 7.19 的程序 jacobidavidson.m 即可. 例如, 要计算例 7.15 中的矩阵与 $\alpha = 18$ 最接近的特征值和相应的特征向量, 可以编写如下 MATLAB 脚本程序:

```
tic; n=1000; A=zeros(n,n);
for i=1:n,
    for j=1:i
```

```
            A(i,j)=j; A(j,i)=A(i,j);
        end
    end
    v=rand(n,1); I=eye(n);
    alpha=18; B=(A-alpha*I)\I;
    [mu,u,Vm]=jacobidavidson(B,v);
    lambda=alpha+1/mu,
    err1=norm(A*u-lambda*u),d=eig(A);
    err2=norm(lambda-d(925)),toc
```

执行上述程序语句, 得到 $\lambda = 17.8762$, $\|\boldsymbol{A}\boldsymbol{u} - \lambda\boldsymbol{u}\|_2 = 8.1153 \times 10^{-9}$.