

第三步： 令 $Q_3 = \text{diag}(I_{3 \times 3}, H_3)$, 其中 H_3 是对应于向量 $A_2(4:5, 3)$ 的 Householder 矩阵, 则有

$$Q_3 A_2 = \begin{bmatrix} * & * & * & * & * \\ * & * & * & * & * \\ 0 & * & * & * & * \\ 0 & 0 & * & * & * \\ 0 & 0 & 0 & * & * \end{bmatrix} \text{ 和 } A_3 \triangleq Q_3 A_2 Q_3^\top = \begin{bmatrix} * & * & * & * & * \\ * & * & * & * & * \\ 0 & * & * & * & * \\ 0 & 0 & * & * & * \\ 0 & 0 & 0 & * & * \end{bmatrix}$$

这时, 我们就将 A 转化成一个上 Hessenberg 矩阵, 即 $QAQ^\top = A_3$, 其中 $Q = Q_3 Q_2 Q_1$ 是正交矩阵, A_3 是上 Hessenberg 矩阵。

上 Hessenberg 化算法

算法 5.1 上 Hessenberg 化算法 (Upper Hessenberg Reduction)

- 1: set $Q = I$
- 2: for $k = 1$ to $n - 2$ do
- 3: compute Hessenberg matrix H_k with respect to $A(k+1:n, k)$
- 4: $A(k+1:n, k:n) = H_k \cdot A(k+1:n, k:n)$
 $\quad = A(k+1:n, k:n) - \beta_k v_k (v_k^\top A(k+1:n, k:n))$
- 5: $A(1:n, k+1:n) = A(1:n, k+1:n) \cdot H_k^\top$
 $\quad = A(1:n, k+1:n) - \beta_k A(1:n, k+1:n) v_k v_k^\top$
- 6: $Q(k+1:n, k:n) = H_k \cdot Q(k+1:n, k:n)$
 $\quad = Q(k+1:n, k:n) - \beta_k v_k (v_k^\top Q(k+1:n, k:n))$
- 7: end for

说明：

- 在实际计算时, 我们不需要显式地形成 Householder 矩阵 H_k 。
- 上述算法的运算量大约为 $\frac{14}{3}n^3 + \mathcal{O}(n^2)$ 。如果不需要计算特征向量, 则正交矩阵 Q 也不用计算, 此时运算量大约为 $\frac{10}{3}n^3 + \mathcal{O}(n^2)$ 。
- 上 Hessenberg 矩阵的一个很重要的性质就是在 QR 迭代中保持形状不变。

定理 设 $A \in \mathbb{R}^{n \times n}$ 是非奇异上 Hessenberg 矩阵, 其 QR 分解为 $A = QR$, 则 $\tilde{A} \triangleq RQ$ 也是上 Hessenberg 矩阵。

若 A 是奇异的, 也可以通过选取适当的 Q , 使得上述结论成立。

由此可知, 如果 A 是上 Hessenberg 矩阵, 则 QR 迭代中的每一个 A_k 都是上 Hessenberg 矩阵。这样在进行 QR 分解时, 运算量可大大降低。

Hessenberg 矩阵另一重要性质: 在 QR 迭代中保持下次对角线元素非零。

定理 设 $A \in \mathbb{R}^{n \times n}$ 是上 Hessenberg 矩阵且下次对角线元素均非零, 即 $a_{i+1,i} \neq 0, i = 1, 2, \dots, n-1$ 。设其 QR 分解为 $A = QR$, 则 $\tilde{A} \triangleq RQ$ 的下次对角线元素也都非零。

若 A 村咋子某个下次对角线元素为零, 则 A 一定可约。因此, 我们只需考虑下次对角线均非零的情形。

推论 $\tilde{A} \triangleq RQ$ 则在带位移的 QR 迭代中, 所有的 A_k 的下次对角线元素均非零。

0.0.1 隐式 QR 迭代

在 QR 迭代中, 我们要先做 QR 分解 $A_k = Q_k R_k$, 然后计算 $A_{k+1} = Q_k R_k$ 。但事实上, 我们可以直接计算出 A_{k+1} 。这就是**隐式 QR 迭代**。

不失一般性, 我们假定 A 是不可约的上 Hessenberg 矩阵。

隐式 QR 迭代的理论基础就是下面的**隐式 Q 定理**。

定理 (Implicit Q Theorem) 设 $H = Q^T A Q \in \mathbb{R}^{n \times n}$ 是一个不可约上 Hessenberg 矩阵, 其中 $Q \in \mathbb{R}^{n \times n}$ 是正交矩阵, 则 Q 的第 2 至第 n 列均由 Q 的第一列所唯一确定(可相差一个符号)。

由于 Q_k 的其他列都由 Q_k 的第一列唯一确定(至多相差一个符号), 所以我们只要找到一个正交矩阵 \tilde{Q}_k 使得其第一列与 Q_k 的第一列相等, 且 $\tilde{Q}_k^T A_k \tilde{Q}_k$ 为上 Hessenberg 矩阵, 则由隐式 Q 定理可知 $\tilde{Q}_k = W Q_k$, 其中 $W = \text{diag}(1, \pm 1, \dots, \pm 1)$, 于是

$$\tilde{Q}_k^T A_k \tilde{Q}_k = W^T Q_k^T A_k Q_k W = W^T A_{k+1} W$$

。又 $W^T A_{k+1} W$ 与 A_{k+1} 相似, 且对角线元素相等, 而其他元素也至多相差一个符号, 所以不会影响 A_{k+1} 的收敛性, 即下三角元素收敛到 0, 对角线元素收敛到 A 的特征值。

在 QR 迭代算法中, 如果我们直接令 $A_{k+1} = \tilde{Q}_k^T A_k \tilde{Q}_k$, 则其收敛性与原 QR 迭代算法没有任何区别! 这就是隐式 QR 迭代的基本思想。

由于 A 是上 Hessenberg 矩阵, 因此在实际计算中, 我们只需 Givens 变换。

下面我们举一个例子, 具体说明如何利用隐式 Q 定理, 由 A_1 得到 A_2 。

设 $A \in \mathbb{R}^{5 \times 5}$ 是一个不可约上 Hessenberg 矩阵, 即

$$A_1 = A = \begin{bmatrix} * & * & * & * & * \\ * & * & * & * & * \\ 0 & * & * & * & * \\ 0 & 0 & * & * & * \\ 0 & 0 & 0 & * & * \end{bmatrix}$$

第一步: 构造一个 Givens 变换

$$G_1^T \triangleq G(1, 2, \theta_1) = \begin{bmatrix} c_1 & s_1 & & & \\ -s_1 & c_1 & & & \\ & & I_3 & & \end{bmatrix} \quad (c_1, s_1 \text{ 待定})$$

于是有

$$G_1^\top A = \begin{bmatrix} * & * & * & * & * \\ * & * & * & * & * \\ 0 & * & * & * & * \\ 0 & 0 & * & * & * \\ 0 & 0 & 0 & * & * \end{bmatrix} \text{ 和 } A^{(1)} \triangleq G_1^\top A G_1 = \begin{bmatrix} * & * & * & * & * \\ * & * & * & * & * \\ + & * & * & * & * \\ 0 & 0 & * & * & * \\ 0 & 0 & 0 & * & * \end{bmatrix}$$

与 A_1 相比较, $A^{(1)}$ 在 $(3,1)$ 位置上多出一个非零元, 我们把它记为 “+”, 并称之为 **bulge**。在下面的计算过程中, 我们的目标就是将其“赶”出矩阵, 从而得到一个新的上 Hessenberg 矩阵, 即 A_2 。

第二步: 为了消去这个 **bulge**, 我们可以构造 Givens 变换

$$G_2^\top \triangleq G(2, 3, \theta_2) = \begin{bmatrix} 1 & & & & \\ & c_2 & s_2 & & \\ & -s_2 & c_2 & & \\ & & & I_2 & \end{bmatrix} \text{ 使得 } G_2^\top A^{(1)} = \begin{bmatrix} * & * & * & * & * \\ * & * & * & * & * \\ 0 & * & * & * & * \\ 0 & 0 & * & * & * \\ 0 & 0 & 0 & * & * \end{bmatrix}$$

为了保持与原矩阵的相似性, 需要再右乘 G_2 , 所以

$$A^{(2)} \triangleq G_2^\top A^{(1)} G_2 = \begin{bmatrix} * & * & * & * & * \\ * & * & * & * & * \\ 0 & * & * & * & * \\ 0 & + & * & * & * \\ 0 & 0 & 0 & * & * \end{bmatrix}$$

此时, **bugle** 从 $(3,1)$ 位置被“赶”到 $(4,2)$ 位置。

第三步: 与第二步类似, 构造 Givens 变换

$$G_3^\top \triangleq G(3, 4, \theta_3) = \begin{bmatrix} I_2 & & & & \\ & c_3 & s_3 & & \\ & -s_3 & c_3 & & \\ & & & 1 & \end{bmatrix} \text{ 使得 } G_3^\top A^{(2)} = \begin{bmatrix} * & * & * & * & * \\ * & * & * & * & * \\ 0 & * & * & * & * \\ 0 & 0 & * & * & * \\ 0 & 0 & 0 & * & * \end{bmatrix}$$

这时

$$A^{(3)} \triangleq G_3^\top A^{(2)} G_3 = \begin{bmatrix} * & * & * & * & * \\ * & * & * & * & * \\ 0 & * & * & * & * \\ 0 & 0 & * & * & * \\ 0 & 0 & + & * & * \end{bmatrix}$$

于是,bugle 又从 (4, 2) 位置被“赶”到 (5, 3) 位置。

第四步： 再次构造 Givens 变换

$$G_4^\top \triangleq G(4, 5, \theta_4) = \begin{bmatrix} 1 & & & & \\ & 1 & & & \\ & & 1 & & \\ & & & c_4 & s_4 \\ & & & -s_4 & c_4 \end{bmatrix} \quad \text{使得 } G_4^\top A^{(3)} = \begin{bmatrix} * & * & * & * & * \\ * & * & * & * & * \\ 0 & * & * & * & * \\ 0 & 0 & * & * & * \\ 0 & 0 & 0 & * & * \end{bmatrix}$$

这时

$$A^{(4)} \triangleq G_4^\top A^{(3)} G_4 = \begin{bmatrix} * & * & * & * & * \\ * & * & * & * & * \\ 0 & * & * & * & * \\ 0 & 0 & * & * & * \\ 0 & 0 & 0 & * & * \end{bmatrix}$$

现在, bulge 被“赶”出矩阵, $A^{(4)}$ 就是我们所要的矩阵!

算法分析, 以及 c_1, s_1 的取值

常规 QR 迭代: $A_1 = Q_1 R_1, A_2 = R_1 Q_1 \Rightarrow A_2 = Q_1^\top A_1 Q_1$

根据前面的计算过程, 有

$$A^{(4)} = G_4^\top G_3^\top G_2^\top G_1^\top A_1 G_1 G_2 G_3 G_4 = \tilde{Q}_1^\top A_1 \tilde{Q}_1$$

, 其中 $\tilde{Q}_1 = G_1 G_2 G_3 G_4 \Rightarrow A^{(4)} = \tilde{Q}_1^\top A_1 \tilde{Q}_1$

通过直接计算可知, \tilde{Q}_1 的第一列为

$$[c_1, s_1, 0, 0, 0]^\top$$

如果将其取为 A_1 的第一列 $[a_{11}, a_{21}, 0, \dots, 0]^\top$ 单位化后的向量, 则 \tilde{Q}_1 的第一列与 Q_1 的第一列相同! $\Rightarrow A^{(4)} = W^\top A_2 W$

针对带位移的 QR 方法, 我们取 $A_1 - \sigma_1 I$ 的第一列

$$[a_{11} - \sigma_1, a_{21}, 0, \dots, 0]^\top$$

单位化后的向量作为 G_1 的第一列即可。

运算量:

如果 $A \in \mathbb{R}^{n \times n}$ 是上 Hessenberg 矩阵, 则使用上面的算法, 带位移 QR 迭代中每一步的运算量为 $6n^2 + O(n)$ 。

0.0.2 位移的选取

通常, 位移越离某个特征值越近, 则收敛速度就越快。

由习题 4.10 可知,如果位移 σ 与某个特征值非常接近,则 $A_k(n, n) - \sigma$ 就非常接近于 0。

这说明 $A_k(n, n)$ 通常会首先收敛到 A 的一个特征值。所以 $\sigma = A_k(n, n)$ 是一个不错的选择。但是,如果这个特征值是复数,这种唯一选取方法就可能失效。

双位移策略

设 $\sigma \in \mathbb{C}$ 是 A 的某个复特征值 λ 的一个很好的近似,则其共轭 $\bar{\sigma}$ 也应该是 $\bar{\lambda}$ 的一个很好的近似。因此我们可以考虑双位移策略,即先以 λ 为位移迭代一次,然后再以 $\bar{\sigma}$ 为位移迭代一次,如此不断交替进行迭代。

这样就有

$$\begin{aligned} A_1 - \sigma I &= Q_1 R_1 \\ A_2 &= R_1 Q_1 + \sigma I \\ A_2 - \bar{\sigma} I &= Q_2 R_2 \\ A_3 &= R_2 Q_2 + \bar{\sigma} I \end{aligned}$$

容易验证

$$A_3 = Q_2^\top A_2 Q_2 = Q_2^* Q_1^* A_1 Q_1 Q_2 = Q^* A_1 Q$$

其中 $Q = Q_1 Q_2$

我们注意到 σ 可能是复的,所以 Q_1 和 Q_2 都可能是复矩阵。但我们却可以选取适当的 Q_1 和 Q_2 ,使得 $Q = Q_1 Q_2$ 是实矩阵。

双位移策略的实现

由前面的结论可知,存在 Q_1 和 Q_2 ,使得 $Q = Q_1 Q_2$ 是实矩阵,从而

$$A_3 = Q^\top A_1 Q$$

也是实矩阵。因此我们希望不计算 A_2 ,而是直接从 A_1 得到 A_3
实现方式:

根据隐式 Q 定理:只要找到一个实正交矩阵 Q ,使得其第一列与

$$A_1^2 - 2 \operatorname{Re}(\sigma) A_1 + |\sigma|^2 I$$

的第一列平行,并且 $A_3 = Q^\top A_1 Q$ 是上 Hessenberg 矩阵即可。

易知, $A_1^2 - 2 \operatorname{Re}(\sigma) A_1 + |\sigma|^2 I$ 的第一列为

$$\begin{bmatrix} a_{11}^2 + a_{12}a_{21} - 2 \operatorname{Re}(\sigma)a_{11} + |\sigma|^2 \\ a_{21}(a_{11} + a_{22} - 2 \operatorname{Re}(\sigma)) \\ a_{21}a_{32} \\ 0 \\ \vdots \end{bmatrix} \quad (1)$$

所以 Q 的第一列是上述向量的单位化。

其他过程可以通过隐式 QR 迭代来实现。但此时的“bulge”是一个 2×2 的小矩阵。因此,在双位移隐式 R 迭代过程中,需要使用 Householder 变换。

需要指出的是,双位移 QR 迭代算法中的运算都是实数运算。

下面通过一个例子来说明如何在实数运算下实现双位移隐式 QR 迭代。

设 $A \in \mathbb{R}^{6 \times 6}$ 是一个不可约上 Hessenberg 矩阵, 即

$$A_1 = A = \begin{bmatrix} * & * & * & * & * & * \\ * & * & * & * & * & * \\ 0 & * & * & * & * & * \\ 0 & 0 & * & * & * & * \\ 0 & 0 & 0 & * & * & * \\ 0 & 0 & 0 & 0 & * & * \end{bmatrix}$$

第一步: 构造一个正交矩阵 $H_1 = \begin{bmatrix} \tilde{H}_1^\top & 0 \\ 0 & I_3 \end{bmatrix}$, 其中 $\tilde{H}_1 \in \mathbb{R}^{3 \times 3}$, 使得第一列与 $A_1^2 - 2\operatorname{Re}(\sigma)A_1 + |\sigma|^2I$ 的第一列平行。于是有

$$H_1^\top A = \begin{bmatrix} * & * & * & * & * & * \\ * & * & * & * & * & * \\ + & * & * & * & * & * \\ 0 & 0 & * & * & * & * \\ 0 & 0 & 0 & * & * & * \\ 0 & 0 & 0 & 0 & * & * \end{bmatrix} \text{ 和 } A^{(1)} \triangleq H_1^\top A H_1 = \begin{bmatrix} * & * & * & * & * & * \\ * & * & * & * & * & * \\ + & * & * & * & * & * \\ + & + & * & * & * & * \\ 0 & 0 & 0 & * & * & * \\ 0 & 0 & 0 & 0 & * & * \end{bmatrix}$$

与 A_1 相比较, $A^{(1)}$ 在 $(3, 1), (4, 1)$ 和 $(4, 2)$ 位置上出现 **bulge**。在下面的计算过程中, 我们的目标就是把它们”赶“出矩阵, 从而得到一个新的上 Hessenberg 矩阵。

第二步: 令 $H_2 = \begin{bmatrix} 1 & 0 & 0 \\ 0 & \tilde{H}_2^\top & 0 \\ 0 & 0 & I_2 \end{bmatrix}$, 其中 $\tilde{H}_2 \in \mathbb{R}^{3 \times 3}$ 是对应于 $A(2 : 4, 1)$ 的

Householder 变换, 使得

$$H_2^\top A^{(1)} = \begin{bmatrix} * & * & * & * & * & * \\ * & * & * & * & * & * \\ 0 & * & * & * & * & * \\ 0 & + & * & * & * & * \\ 0 & 0 & 0 & * & * & * \\ 0 & 0 & 0 & 0 & * & * \end{bmatrix} \text{ 和 } A^{(2)} \triangleq H_2^\top A^{(1)} H_2 = \begin{bmatrix} * & * & * & * & * & * \\ * & * & * & * & * & * \\ 0 & * & * & * & * & * \\ 0 & + & * & * & * & * \\ 0 & + & + & * & * & * \\ 0 & 0 & 0 & 0 & * & * \end{bmatrix}$$

这时, 我们将 **bugle** 向右下角方向”赶“了一个位置。

第三步 与第二步类似, 令 $H_3 = \begin{bmatrix} I_2 & 0 & 0 \\ 0 & \tilde{H}_3^\top & 0 \\ 0 & 0 & 1 \end{bmatrix}$, 其中 $\tilde{H}_3 \in \mathbb{R}^{3 \times 3}$ 是对应于

$A(3:5, 2)$ 的 **Householder** 变换, 使得

$$H_3^\top A^{(2)} = \begin{bmatrix} * & * & * & * & * & * \\ * & * & * & * & * & * \\ 0 & * & * & * & * & * \\ 0 & 0 & * & * & * & * \\ 0 & 0 & + & * & * & * \\ 0 & 0 & 0 & 0 & * & * \end{bmatrix} \text{ 和 } A^{(3)} \triangleq H_3^\top A^{(2)} H_3 = \begin{bmatrix} * & * & * & * & * & * \\ * & * & * & * & * & * \\ 0 & * & * & * & * & * \\ 0 & 0 & * & * & * & * \\ 0 & 0 & + & * & * & * \\ 0 & 0 & + & + & * & * \end{bmatrix}$$

这时, **bugle** 又被向右下角方向”赶“了一个位置。

第四步 令 $H_4 = \begin{bmatrix} I_3 & 0 \\ 0 & \tilde{H}_4^\top \end{bmatrix}$, 其中 $\tilde{H}_4 \in \mathbb{R}^{3 \times 3}$ 是对应于 $A(4:6, 3)$ 的 **Householder** 变换, 使得

$$H_4^\top A^{(3)} = \begin{bmatrix} * & * & * & * & * & * \\ * & * & * & * & * & * \\ 0 & * & * & * & * & * \\ 0 & 0 & * & * & * & * \\ 0 & 0 & 0 & * & * & * \\ 0 & 0 & 0 & + & * & * \end{bmatrix} \text{ 和 } A^{(4)} \triangleq H_4^\top A^{(3)} H_4 = \begin{bmatrix} * & * & * & * & * & * \\ * & * & * & * & * & * \\ 0 & * & * & * & * & * \\ 0 & 0 & * & * & * & * \\ 0 & 0 & 0 & * & * & * \\ 0 & 0 & 0 & + & * & * \end{bmatrix}$$

第五步 只需构造一个 **Givens** 变换 $G_5 = \begin{bmatrix} I_4 & 0 \\ 0 & G(4, 5, \theta)^\top \end{bmatrix}$, 使得

$$G_5^\top A^{(4)} = \begin{bmatrix} * & * & * & * & * & * \\ * & * & * & * & * & * \\ 0 & * & * & * & * & * \\ 0 & 0 & * & * & * & * \\ 0 & 0 & 0 & * & * & * \\ 0 & 0 & 0 & 0 & * & * \end{bmatrix} \text{ 和 } A^{(5)} \triangleq G_5^\top A^{(4)} G_5 = \begin{bmatrix} * & * & * & * & * & * \\ * & * & * & * & * & * \\ 0 & * & * & * & * & * \\ 0 & 0 & * & * & * & * \\ 0 & 0 & 0 & * & * & * \\ 0 & 0 & 0 & 0 & * & * \end{bmatrix}$$

现在, **bulge** 已经被全部消除, 且

$$A^{(5)} = Q^\top A Q$$

, 其中 $Q = H_1 H_2 H_3 H_4 G_5$ 。通过直接计算可知, Q 的第一列即为 H_1 的第一列。根据隐式 Q 定理, 可以直接令 $A_3 \triangleq A^{(5)} = Q^\top A Q$ 。

位移的具体选取

在单位移 QR 迭代算法中, 若 A 的特征值都是实的, 则取 $\sigma_k = A_k(n, n)$. 推广到复共轭特征值上, 我们可以取 A_k 的右下角矩阵

$$\begin{bmatrix} A_k(n-1, n-1) & A_k(n-1, n) \\ A_k(n, n-1) & A_k(n, n) \end{bmatrix}$$

的复共轭特征值作为双位移。这样选取的位移就是 **Francis 位移**。

如果上述矩阵的两个特征值都是实的, 则选取其中模较小的特征值做单位移。

采用 **Francis 位移** 的 QR 迭代会使得 A_k 的右下角收敛到一个上三角矩阵 (两个实特征值) 或一个 2 阶的矩阵 (一对复共轭特征值), 而且通常会有二次收敛性。在实际计算中, 一个特征值一般平均只需迭代两步。

收敛性判断:

判断收敛性主要是看 $A_k(n-1, n-2)$ (或 $A_k(n, n-1)$) 是否趋向于 0。

需要指出的是, QR 迭代并不是对所有的矩阵都收敛。例如:

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

对于上面的矩阵, 采用 **Francis 位移** 的 QR 迭代算法无效。另外, 也可以考虑多重位移策略, 参见[Watkins 2007]。

0.0.3 收缩 Deflation

收缩 (deflation) 技术是实用 QR 迭代中的一个非常重要概念。

隐式 QR 迭代过程中, 当矩阵 A_{k+1} 的某个下次对角线元素 $a_{i+1, i}$ 很小时, 我们可以将其设为 0。

由于 A_{k+1} 是上 **Hessenberg** 矩阵, 这时 A_{k+1} 就可以写成分块上三角形式, 其中两个对角块都是上 **Hessenberg** 矩阵。

因此我们可以将隐式 QR 迭代作用在这两个规模相对较小的矩阵上, 从而可以大大节约运算量。

0.1 特征向量的计算

设 A 的特征值都是实的, $R = Q^T A Q$ 是其 **Schur** 标准型。若 $Ax = \lambda x$, 则 $Ry = \lambda y$, 其中 $y = Q^T x$ 或 $x = Qy$ 。故只需计算 R 的特征向量 y 即可。

因为 R 的对角线元素即为 A 的特征值, 不妨设 $\lambda = R(i, i)$ 。

假定 λ 是单重特征值, 则方程 $(R - \lambda I)y = 0$ 即为

$$\begin{bmatrix} R_{11} - \lambda I R_{12} & R_{13} \\ 0 & 0 & R_{23} \\ 0 & 0 & R_{33} - \lambda I \end{bmatrix} \begin{bmatrix} y_1 \\ y_2 \\ y_3 \end{bmatrix} = 0$$

即

$$(R_{11} - \lambda I) y_1 + R_{12} y_2 + R_{13} y_3 = 0 \quad (2)$$

$$R_{23} y_3 = 0 \quad (3)$$

$$(R_{33} - \lambda I) y_3 = 0 \quad (4)$$

其中 $R_{11} \in \mathbb{R}^{(i-1) \times (i-1)}$, $R_{33} \in \mathbb{R}^{(n-i) \times (n-i)}$ 。由于 λ 是单重特征值, 故 $R_{33} - \lambda I$ 非奇异, 因此 $y_3 = 0$ 。令 $y_2 = 1$, 则可得

$$y_1 = (R_{11} - \lambda I)^{-1} R_{12}$$

因此计算特征向量 y 只需求解一个上三角线性方程组。

若 λ 是多重特征值, 则据算方法类似。但如果 A 有负特征值, 则需要利用实 Schur 标准型, 计算较复杂。

0.2 广义特征值问题

设 $A, B \in \mathbb{R}^{n \times n}$, 若存在 $\lambda \in \mathbb{C}$ 和非零向量 $x \in \mathbb{C}^n$ 使得

$$Ax = \lambda Bx$$

则称 λ 为矩阵对 (A, B) 的特征值, x 为对应的特征向量。

计算矩阵对 (A, B) 的特征值和特征向量就是[广义特征值问题](#)

当 B 非奇异时, 广义特征值问题就等价于标准特征值问题

$$B^{-1}Ax = \lambda x \text{ 或 } AB^{-1}y = \lambda y$$

其中 $y = Bx$ 。

容易看出, λ 是 (A, B) 的一个特征值当且仅当

$$\det(A - \lambda B) = 0 \quad (5)$$

若5对所有 $\lambda \in \mathbb{C}$ 都成立, 则称矩阵对 (A, B) 是[奇异矩阵对](#), 否则称为[正则矩阵对](#)。

当 B 非奇异时, 特征方程5是一个 n 次多项式, 因此恰好有 n 个特好找呢个字。当 B 奇异时, 特征方程5的次数低于 n , 因此方程的解的个数小于 n 。但是, 注意带 $\lambda \neq 0$ 是 (A, B) 的 t 特征值当且仅当 $\mu = \frac{1}{\lambda}$ 是 (B, A) 的特征值。因此, 当 B 奇异时, $\mu = 0$ 是 (B, A) 的特征值, 于是我们自然的把 $\lambda = \frac{1}{\mu} = \infty$ 当作是 (A, B) 的特征值。所以广义特征值不是分布在 \mathbb{C} 上, 而是分布在 $\mathbb{C} \cup \{\infty\}$ 上。

容易验证, 若 U, V 非奇异, 则矩阵对 (U^*AV, U^*BV) 的特征值与 (A, B) 是一样的。因此我们称这种变换为[矩阵对的等价变换](#)。如果 U, V 是酉矩阵, 则称为[酉等价变换](#)。

0.2.1 广义 Schur 分解

广义 Schur 分解是矩阵对在酉等价变换下的最简形式。

定理 (广义 Schur 分解) 设 $A, B \in \mathbb{C}^{n \times n}$, 则存在酉矩阵 $Q, Z \in \mathbb{C}^{n \times n}$, 使得

$$Q^*AZ = R_A, \quad Q^*BZ = R_B \quad (6)$$

其中 $R_A, R_B \in \mathbb{C}^{n \times n}$ 都是上三角矩阵。此时矩阵对 (A, B) 的特征值为 R_A 和 R_B 的对角线元素的比值, 即

$$\lambda_i = \frac{R_A(i, i)}{R_B(i, i)}, \quad i = 1, 2, \dots, n$$

当 $R_B(i, i) = 0$ 时, 对应的特征值 $\lambda_i = \infty$ 。

证明参见[Xu-Qian 2011]。

与实 Schur 分解类似, 当 A, B 都是实矩阵时, 我们有相应的广义实 Schur 分解。

定理 (广义 Schur 分解) 设 $A, B \in \mathbb{R}^{n \times n}$, 则存在酉矩阵 $Q, Z \in \mathbb{R}^{n \times n}$, 使得

$$Q^TAZ = T_A, \quad Q^TBZ = T_B \quad (7)$$

其中 $T_A, T_B \in \mathbb{R}^{n \times n}$ 都是拟上三角矩阵。

证明参见[Xu-Qian 2011]。

0.2.2 QZ 迭代

QZ 迭代是用于计算 (A, B) 的广义 Schur 分解的算法, 是 QR 算法的自然推广, 实质上可以看作是将 QR 算法作用到矩阵 AB^{-1} 上。

详细算法可参见[Kressner 2005, Xu-Qian 2011]。

0.3 应用: 多项式求根

考虑 n 次多项式

$$q_n(x) = x^n + c_{n-1}x^{n-1} + \dots + c_1x + c_0, \quad c_i \in \mathbb{R}$$

- 由代数学基本定理可知, $p_n(x)$ 在复数域中有且仅有 n 的零点
- $n \geq 5$ 时, 不存在求根公式
- 非线性迭代方法求解
- MATLAB 中的 `roots` 命令: 通过特征值计算方法求出所有零点

友矩阵

$$A = \begin{bmatrix} 0 & & -c_0 \\ 1 & 0 & & -c_1 \\ \ddots & \ddots & \vdots & \\ & & 1 & -c_{n-1} \end{bmatrix}$$

多项式 $q_n(x)$ 的零点 $\iff A$ 的特征值

- 无需上 **Hessenberg** 化
- A 非常稀疏, 但经过一步 **QR** 迭代后, 上三角部分的零元素会消失, 总运算量仍是 $O(n^3)$
- **快速 QR 方法**: 利用 A 的特殊结构, 运算量 $O(n^2)$

将 A 写成一个酉矩阵与秩一矩阵之差, 具体实现参见相关文献