

# Python 数值分析 算法实践



Python Numerical Analysis Algorithm Practice

## 第10章 矩阵特征值计算



讲授：XXX



日期：2024年4月19日

## 第10章 矩阵特征值计算

高等代数中, 求矩阵  $\mathbf{A}$  的特征值  $\lambda$  和特征向量  $\mathbf{x}$  的解法: 先求出  $\mathbf{A}$  的特征多项式

$$f(\lambda) = |\mathbf{A} - \lambda \mathbf{I}| = \begin{vmatrix} a_{11} - \lambda & a_{12} & \cdots & a_{1n} \\ a_{21} & a_{22} - \lambda & \cdots & a_{2n} \\ \cdots & \cdots & \cdots & \cdots \\ a_{n1} & a_{n2} & \cdots & a_{nn} - \lambda \end{vmatrix}$$

再求解  $f(\lambda)$  高次多项式方程, 所得根  $\lambda^*$  即为矩阵  $\mathbf{A}$  的特征值, 最后求解方程组  $(\mathbf{A} - \lambda^* \mathbf{I})\mathbf{x} = \mathbf{0}$ , 就可得出特征值  $\lambda^*$  对应的特征向量  $\mathbf{x}$ .

然而求解高次多项式的根是相当困难的, 且重根的计算精度较低, 矩阵求特征多项式系数的过程对舍入误差十分敏感. 因此, 从数值计算角度, 上述方法缺乏实用价值. 目前, 求矩阵特征值问题实际采用的是迭代法和变换法.

# 矩阵特征值计算

## 10.1 求矩阵特征值和特征向量的迭代法

10.1.1 幂法

10.1.2 反幂法

10.1.3 瑞利商加速幂法

10.1.4 原点平移反幂法

10.1.5 \* 收缩法求解矩阵全部特征值

## 10.2 求矩阵全部特征值的正交变换法

10.2.1 Schmidt正交分解QR法

10.2.2 用正交相似变换约化一般矩阵为上海森伯格矩阵

10.2.3 上海森伯格矩阵QR算法

10.2.4 位移上海森伯格矩阵QR算法

## 10.1.1 幂法

幂法也称乘幂法. 幂法用来求按模最大特征值(主特征值)与它对应的特征向量. 幂法的特点是算法简单, 易于在计算机上实现, 特别适用于高阶稀疏方阵.

设  $\mathbf{A} \in \mathbb{R}^{n \times n}$  的特征值为  $\lambda_1, \lambda_2, \dots, \lambda_n$ , 主特征值为  $\lambda_1$  且满足  $|\lambda_1| \geq |\lambda_2| \geq \dots \geq |\lambda_n|$ , 对应的特征向量为  $\mathbf{x}_1, \mathbf{x}_2, \dots, \mathbf{x}_n$ , 且线性无关. 用乘幂法求  $\mathbf{A}$  的主特征值  $\lambda_1$  和属于  $\lambda_1$  的特征向量  $\mathbf{x}_1$  的步骤为: 任取  $n$  维非零向量  $\mathbf{v}^{(0)} = (v_1^{(0)}, v_2^{(0)}, \dots, v_n^{(0)})^T$  作为初始向量, 对迭代次数  $k = 0, 1, \dots$ , 反复迭代计算, 假设第  $k$  步:

计算  $\mathbf{v}^{(k)} = \mathbf{A}\mathbf{v}^{(k-1)}$ , 并归一化, 记  $\mathbf{v}^{(k)} = (v_1^{(k)}, v_2^{(k)}, \dots, v_n^{(k)})^T$ , 令

$$\mathbf{u}^{(k-1)} = (u_1^{(k-1)}, u_2^{(k-1)}, \dots, u_n^{(k-1)})^T, \text{ 其中 } u_i^{(k-1)} = \frac{v_i^{(k)}}{v_i^{(k-1)}}, i = 1, 2, \dots, n.$$

$\lim_{k \rightarrow \infty} u_i^{(k)} = \lambda_1, i = 1, 2, \dots, n, \lim_{k \rightarrow \infty} \frac{v^{(k)}}{\lambda_1^{(k)}} = a_1 \mathbf{x}_1$ , 其中  $a_1 \in \mathbb{R}$  且  $a_1 \neq 0$ . 幂法是线性收敛的, 收敛速

度主要由  $|\lambda_2|/|\lambda_1|$  决定.

## 10.1.2 反幂法

反幂法用来求可逆矩阵的按模最小特征值(即绝对值最小的特征值)和与它对应的特征向量. 设  $\mathbf{A} \in \mathbb{R}^{n \times n}$  为非奇异矩阵,  $\lambda$  和  $\mathbf{x}$  分别为特征值和对应于的特征向量, 则

$$\mathbf{A}\mathbf{x} = \lambda\mathbf{x} \Rightarrow \mathbf{x} = \lambda\mathbf{A}^{-1}\mathbf{x} \Rightarrow \mathbf{A}^{-1}\mathbf{x} = \frac{1}{\lambda}\mathbf{x}.$$

此式表明,  $\mathbf{A}^{-1}$  的特征值是  $\mathbf{A}$  的特征值的倒数, 而相应的特征向量不变. 因此, 若对矩阵  $\mathbf{A}^{-1}$  用幂法, 即可计算出  $\mathbf{A}^{-1}$  的按模最大的特征值, 其倒数恰为  $\mathbf{A}$  的按模最小的特征值, 这就是反幂法的思想.

用反幂法求  $\mathbf{A}$  的按模最小特征值  $\lambda_n$  和  $\mathbf{A}$  的属于  $\lambda_n$  的特征向量的  $\mathbf{x}_n$  步骤为: 任取  $n$  维非零向量  $\mathbf{v}^{(0)}$  作为初始向量, 反复计算:

$\mathbf{u}^{(k)} = \mathbf{A}^{-1}\mathbf{v}^{(k-1)}$ , 若  $\mathbf{u}^{(k)}$  各分量中绝对值最大的分量为第  $j$  个分量  $u_j^{(k)}$ , 则令  $m^{(k)} = u_j^{(k)}$ , 令  $\mathbf{v}^{(k)} = \mathbf{u}^{(k)} / m^{(k)}$ ,  $k = 1, 2, \dots$ . 当  $k \rightarrow \infty$  时,  $m^{(k)} \rightarrow 1/\lambda_n$ ,  $\mathbf{v}^{(k)}$  接近于  $\mathbf{A}$  的属于  $\lambda_n$  的特征向量. 收敛速度的比值为  $|\lambda_n|/|\lambda_{n-1}|$ .

# 幂法与反幂法

例1:已知两个矩阵  $\mathbf{A}_1$  和  $\mathbf{A}_2$  如下, 用幂法和反幂法求解矩阵  $\mathbf{A}_1$  和  $\mathbf{A}_2$  的特征值和对应的特征向量, 精度要求  $\varepsilon = 10^{-16}$ , 初始向量均为  $\mathbf{v}^{(0)} = (0.5 \ 0.5 \ \cdots \ 0.5)^T$ .

$$\mathbf{A}_1 = \begin{pmatrix} 1 & 1/2 & \cdots & 1/6 \\ 1/2 & 1/3 & \cdots & 1/7 \\ \vdots & \vdots & \ddots & \vdots \\ 1/6 & 1/7 & \cdots & 1/11 \end{pmatrix}, \mathbf{A}_2 = \begin{pmatrix} 2 & 3 & 4 & 5 & 6 \\ 4 & 4 & 5 & 6 & 7 \\ 0 & 3 & 6 & 7 & 8 \\ 0 & 0 & 2 & 8 & 9 \\ 0 & 0 & 0 & 1 & 0 \end{pmatrix}.$$

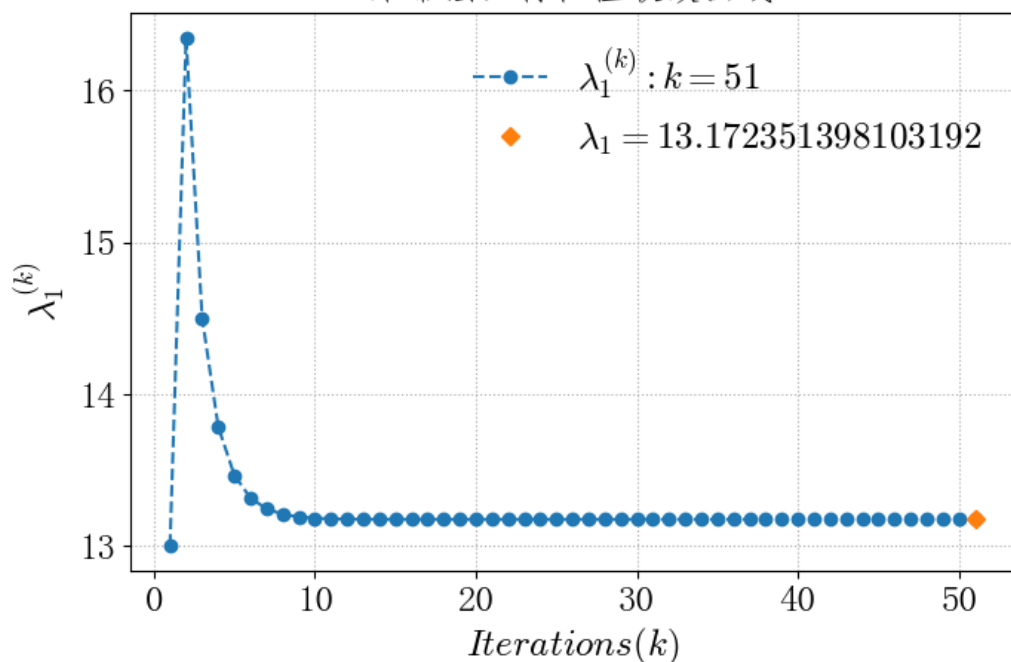
表 10-1 幂法与反幂法求解矩阵特征值和特征向量的结果 (Python 输出格式)

方法和迭代次数	特征值	特征向量 (从左到右, 从上到下构成向量, 并进行了归一化)		
幂法求 $\mathbf{A}_1$ , 迭代 20 次	1.618899858924339	1.0000000000000000	0.588628543425543	0.428327284428956
		0.339661891838709	0.282523587942149	0.242337811122849
反幂法求 $\mathbf{A}_1$ , 迭代 11 次	1.0827994844739511 e-07	-0.001995640508843	0.056928693982721	-0.384803074285784
		1.0000000000000000	-1.102879117733187	0.434248849659761
幂法求 $\mathbf{A}_2$ , 迭代 51 次	13.172351398103192	0.724952325211240	1.0000000000000000	0.792999044338331
		0.353299625949037	0.026821302838908	
反幂法求 $\mathbf{A}_2$	不收敛			

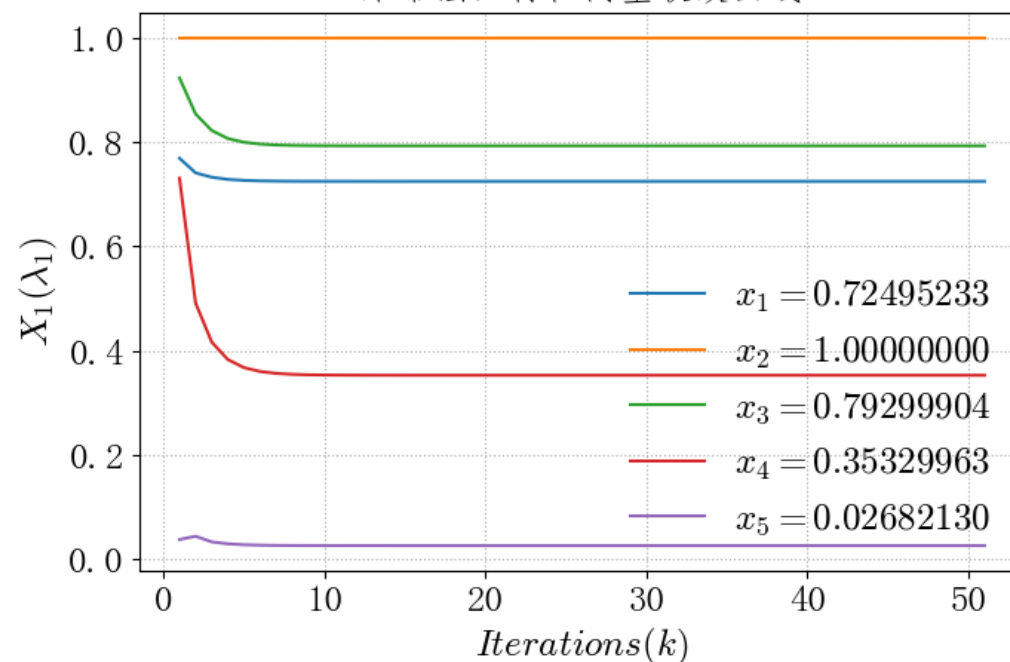
# 幂法与反幂法

幂法求  $\mathbf{A}_2$  按模最大特征值和对应特征向量的收敛曲线如图所示. 从第2次迭代后, 收敛过程较为平稳, 逐步迭代逼近精确解. 验证精度为  $\|(\mathbf{A} - \lambda^* \mathbf{I}) \mathbf{x}^*\|_2 = 5.77969 \times 10^{-15}$ . 可尝试其他初值向量.

乘幂法：特征值收敛曲线



乘幂法：特征向量收敛曲线



## 10.1.3 瑞利商加速幂法

设  $\mathbf{A} \in \mathbb{R}^{n \times n}$  为对称矩阵, 其特征值依次记为  $\lambda_1 \geq \lambda_2 \geq \cdots \geq \lambda_n$ , 则

$$(1) \text{对任意的非零向量 } \mathbf{x} \in \mathbb{R}^n, \lambda_n \leq \frac{(\mathbf{Ax}, \mathbf{x})}{(\mathbf{x}, \mathbf{x})} \leq \lambda_1;$$

$$(2) \lambda_1 = \max_{\mathbf{x} \in \mathbb{R}^n, \mathbf{x} \neq \mathbf{0}} \frac{(\mathbf{Ax}, \mathbf{x})}{(\mathbf{x}, \mathbf{x})}, \lambda_n = \min_{\mathbf{x} \in \mathbb{R}^n, \mathbf{x} \neq \mathbf{0}} \frac{(\mathbf{Ax}, \mathbf{x})}{(\mathbf{x}, \mathbf{x})}, \text{记 } R(\mathbf{x}) = \frac{(\mathbf{Ax}, \mathbf{x})}{(\mathbf{x}, \mathbf{x})}, \mathbf{x} \neq \mathbf{0}, \text{称为矩阵 } \mathbf{A}$$

的瑞利(Rayleigh)商.

瑞利商加速幂法的迭代过程: 给定迭代初值  $\mathbf{u}^{(0)} \neq \mathbf{0}$  和误差限  $\varepsilon > 0$ , 对于  $k = 0, 1, \cdots$ ,

$$\mathbf{v}^{(k+1)} = \mathbf{Au}^{(k)}, R(\mathbf{u}^{(k+1)}) = \frac{(\mathbf{Au}^{(k)}, \mathbf{u}^{(k)})}{(\mathbf{u}^{(k)}, \mathbf{u}^{(k)})}, \mathbf{u}^{(k+1)} = \frac{\mathbf{v}^{(k+1)}}{R(\mathbf{u}^{(k+1)})},$$

如果  $|R(\mathbf{u}^{(k+1)}) - R(\mathbf{u}^{(k)})| < \varepsilon$ , 则结束. 如果矩阵不对称, 也可以用瑞利商加速幂法来求其特征值, 但是加速的效果可能不是很明显.



## 10.1.3 瑞利商加速幂法

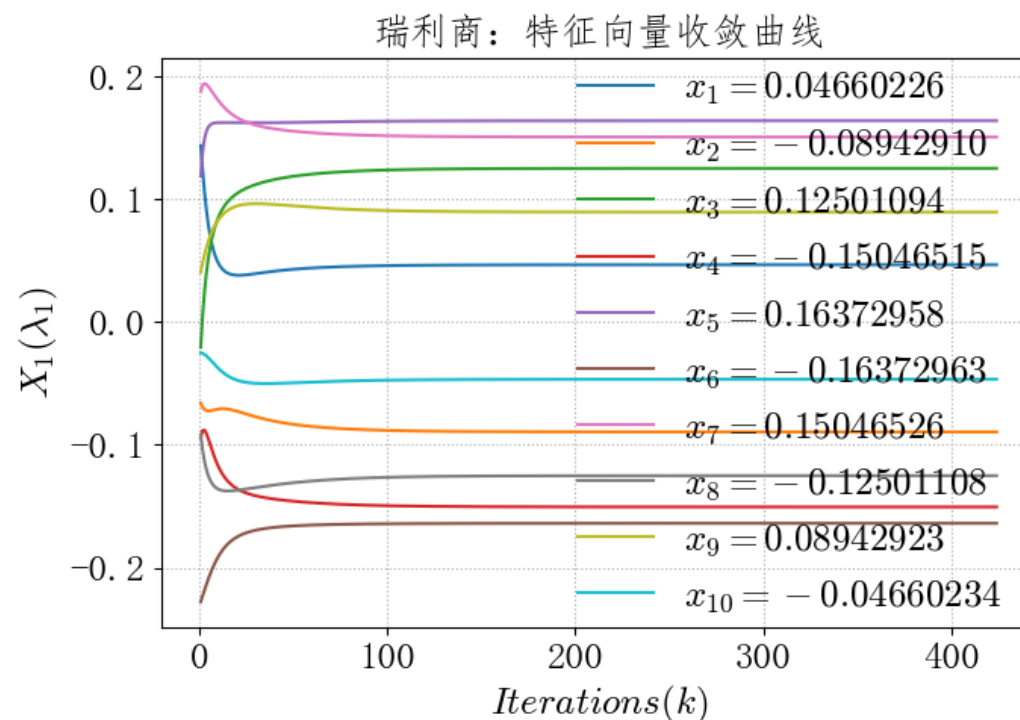
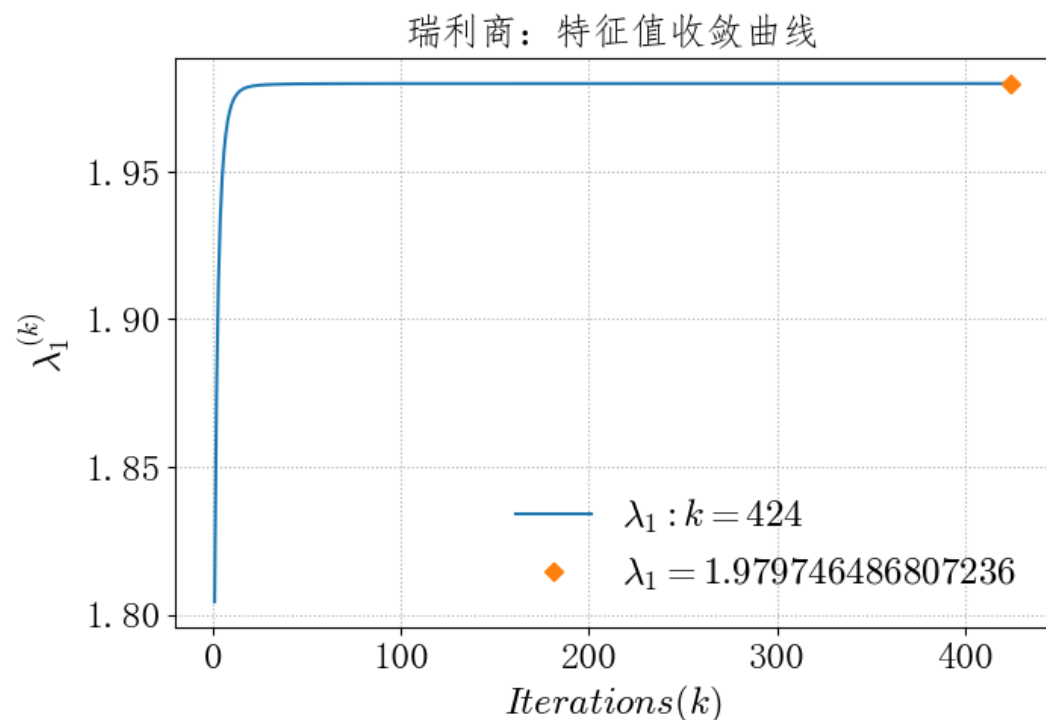
**例2:**已知三对角矩阵  $\mathbf{A}_{10 \times 10}$ , 该矩阵出现在求解热方程的向后有限差分方法中. 用瑞利商加速幂法和幂法求解特征值和特征向量, 精度  $\varepsilon = 10^{-16}$ , 初始向量  $\mathbf{u}^{(0)} = 0.1(u_1^{(0)}, u_2^{(0)}, \dots, u_{10}^{(0)})^T$ , 其中  $u_k^{(0)} \sim N(0, 1)$ ,  $k = 1, 2, \dots, 10$ , 随机种子为1. 分别取  $\alpha = 0.25$ 、 $\alpha = 0.5$  和  $\alpha = 0.75$ .

$$\mathbf{A}_{10 \times 10} = \begin{pmatrix} 1+2\alpha & -\alpha & 0 & \cdots & 0 \\ -\alpha & 1+2\alpha & -\alpha & \ddots & \vdots \\ 0 & \ddots & \ddots & \ddots & 0 \\ \vdots & \ddots & \ddots & \ddots & -\alpha \\ 0 & \cdots & 0 & -\alpha & 1+2\alpha \end{pmatrix}$$

当  $\alpha = 0.25$  时, 瑞利商加速幂法, 满足精度要求下需迭代424次, 而幂法需迭代874次, 保留小数点后15位, 则瑞利商加速幂法所求特征值为  $\lambda^* = 1.979746486807236$ . 特征值与特征向量收敛过程曲线如图所示.

## 10.1.3 瑞利商加速幂法

当 $\alpha = 0.25$ 时, 瑞利商加速幂法, 满足精度要求下需迭代424次, 而幂法需迭代874次, 保留小数点后15位, 则瑞利商加速幂法所求特征值为 $\lambda^* = 1.979746486807236$ . 特征值与特征向量收敛过程曲线如图所示.



### 10.1.3 瑞利商加速幂法

如果所求矩阵为对称矩阵，瑞利商加速幂法可以提高收敛速度，但同时瑞利商的计算会引入舍入误差，如表所示，所求特征向量与库函数`np.linalg.eig`法的求解存在一定的误差，瑞利商加速幂法验证精度 $\|(\mathbf{A} - \lambda^* \mathbf{I})\mathbf{x}^*\|_2 = 6.252 \times 10^{-8}$ 。如果算法终止的条件设置为 $\|\mathbf{u}^{(k+1)} - \mathbf{u}^{(k)}\|_2 < \varepsilon$ ，则迭代1000次的验证精度提高为 $\|(\mathbf{A} - \lambda^* \mathbf{I})\mathbf{x}^*\|_2 = 1.640 \times 10^{-15}$ 。

表 10-2 瑞利商加速幂法特征向量  $(\alpha = 0.25, \mathbf{x}^* = (x_1^*, x_2^*, \dots, x_{10}^*)^T)$

方法	$x_1^*$	$x_2^*$	$x_3^*$	$x_4^*$	$x_5^*$
瑞利商加速幂法	0.28462940	-0.54619987	0.76352058	-0.91898549	0.99999975
np.linalg.eig	-0.28462968	0.54620035	-0.76352112	0.91898595	-1.00000000
方法	$x_6^*$	$x_7^*$	$x_8^*$	$x_9^*$	$x_{10}^*$
瑞利商加速幂法	-1.00000000	0.91898617	-0.76352147	0.54620069	-0.28462988
np.linalg.eig	1.00000000	-0.91898595	0.76352112	-0.54620035	0.28462968

## 10.1.3 瑞利商加速幂法

表 10-3 不同  $\alpha$  值下瑞利商加速幂法求解特征值结果

$\alpha$ 值	方法	特征值	迭代次数	$\ (\mathbf{A} - \lambda^* \mathbf{I}) \mathbf{x}^*\ _2$
$\alpha = 0.5$	瑞利商加速幂法	2.959492973614489	329	7.145155e-08
	幂法	2.959492973614509	662	9.479005e-14
	np.linalg.eig	2.959492973614497		
$\alpha = 0.75$	瑞利商加速幂法	3.939239460421726	286	1.360484e-07
	幂法	3.939239460421754	604	6.169457e-14
	np.linalg.eig	3.939239460421740		

# 矩阵特征值计算

## 10.1 求矩阵特征值和特征向量的迭代法

10.1.1 幂法

10.1.2 反幂法

10.1.3 瑞利商加速幂法

10.1.4 原点平移反幂法

10.1.5 \* 收缩法求解矩阵全部特征值

## 10.2 求矩阵全部特征值的正交变换法

10.2.1 Schmidt正交分解QR法

10.2.2 用正交相似变换约化一般矩阵为上海森伯格矩阵

10.2.3 上海森伯格矩阵QR算法

10.2.4 位移上海森伯格矩阵QR算法

## 10.2.1 Schmidt正交分解QR法

QR算法是求矩阵特征值的最有效和应用最广泛的一种方法, 算法的基本依据:

设  $\mathbf{A}$  是  $n$  阶矩阵, 其  $n$  个特征值为  $\lambda_1, \lambda_2, \dots, \lambda_n$ , 那么存在一个酉矩阵  $\mathbf{U}$ , 使得  $\mathbf{U}^H \mathbf{A} \mathbf{U}$  是以  $\lambda_1, \lambda_2, \dots, \lambda_n$  为对角元的上三角矩阵; 设  $\mathbf{A}$  是  $n$  阶实矩阵, 那么存在一个酉矩阵  $\mathbf{Q}$ , 使得  $\mathbf{Q}^H \mathbf{A} \mathbf{Q}$  为一个准上三角矩阵, 它的每一个对角元是  $\mathbf{A}$  的一个特征值, 对角元上的二阶块矩阵的两个特征值是  $\mathbf{A}$  的一对共轭复特征值.

QR正交化求解矩阵全部特征值的方法: 给定精度  $\varepsilon > 0$  和最大迭代次数  $N$ , 其中  $\mathbf{A}^{(1)} = \mathbf{A}$ , 计算:

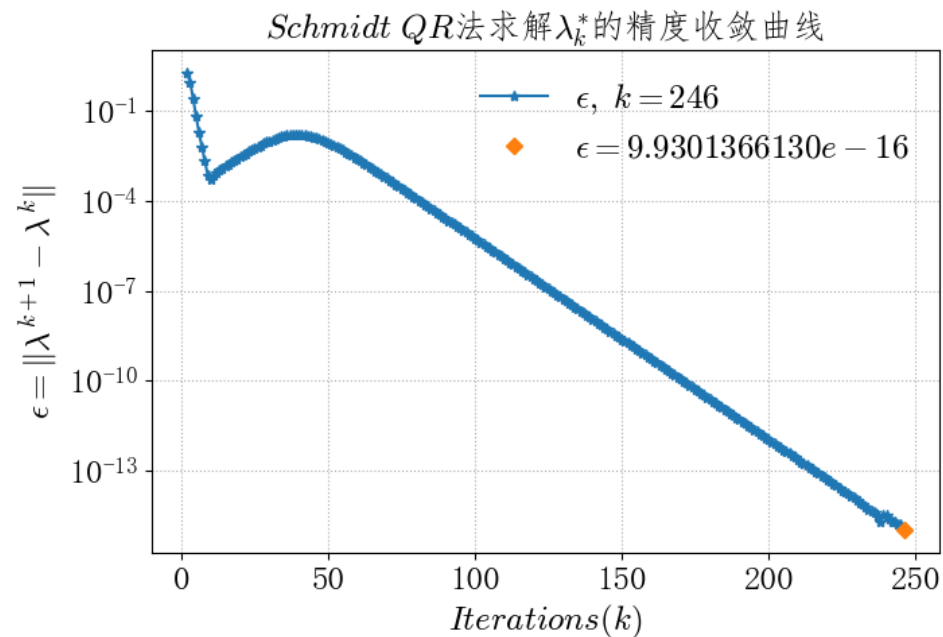
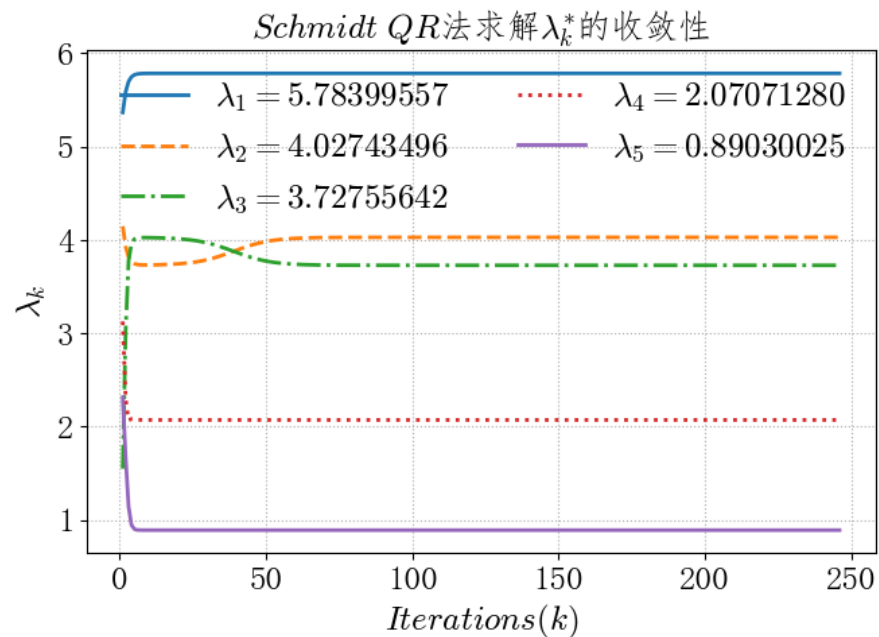
$$\begin{cases} \mathbf{A}^{(k)} = \mathbf{Q}^{(k)} \mathbf{R}^{(k)}, \\ \mathbf{A}^{(k+1)} = \mathbf{R}^{(k)} \mathbf{Q}^{(k)}. \end{cases}$$

QR算法的收敛性质: 如果  $\mathbf{A}$  的特征值满足  $|\lambda_1| \geq |\lambda_2| \geq \dots \geq |\lambda_n|$ , 则QR算法产生的矩阵序列  $\{\mathbf{A}^k\}_{k=1}^N$  基本收敛到上三角矩阵(特别地, 当  $\mathbf{A}$  为对称矩阵时, 收敛到对角阵), 对角元素收敛到  $\mathbf{A}$  的特征值.

## 10.2.1 Schmidt正交分解QR法

例5:采用基本的QR正交分解法求解如下矩阵 $\mathbf{A}$ 的全部特征值, 精度要求 $\varepsilon = 10^{-15}$ .

$$\mathbf{A} = \begin{pmatrix} 5 & -1 & 0 & 0 & 0 \\ -1 & 4.5 & 0.2 & 0 & 0 \\ 0 & 0.2 & 1 & -0.4 & 0 \\ 0 & 0 & -0.4 & 3 & 1 \\ 0 & 0 & 0 & 1 & 3 \end{pmatrix}$$



## 10.2.1 Schmidt正交分解QR法

例5:采用基本的QR正交分解法求解如下矩阵 $\mathbf{A}$ 的全部特征值, 精度要求 $\varepsilon = 10^{-15}$ .

$$\mathbf{A} = \begin{pmatrix} 5 & -1 & 0 & 0 & 0 \\ -1 & 4.5 & 0.2 & 0 & 0 \\ 0 & 0.2 & 1 & -0.4 & 0 \\ 0 & 0 & -0.4 & 3 & 1 \\ 0 & 0 & 0 & 1 & 3 \end{pmatrix}$$

表 10-6 QR 正交分解法求解矩阵全部特征值结果

求解方法	$\lambda_1^*$	$\lambda_2^*$	$\lambda_3^*$
QR 正交分解法	5.78399556651164648002	4.02743495825100072949	3.72755642442787449298
np.linalg.eig(A)	5.78399556651164470367	4.02743495825099895313	3.72755642442787360480
求解方法	$\lambda_4^*$	$\lambda_5^*$	
QR 正交分解法	2.07071280409289215640	0.89030024671658780644	
np.linalg.eig(A)	2.07071280409289126823	0.89030024671658658519	



## 10.2.2 用正交相似变换约化一般矩阵为上海森伯格矩阵

设  $\mathbf{A} = (a_{ij}) \in \mathbb{R}^{n \times n}$ , 可选择初等反射矩阵  $\mathbf{U}_1, \mathbf{U}_2, \dots, \mathbf{U}_{n-2}$ , 使  $\mathbf{A}$  经正交相似变换约化为一个上海森伯格矩阵.

假设经过  $k-1$  次约化, 即有  $\mathbf{A}_k = \mathbf{U}_{k-1} \mathbf{A}_{k-1} \mathbf{U}_{k-1}$  或  $\mathbf{A}_k = \mathbf{U}_{k-1} \cdots \mathbf{U}_1 \mathbf{A} \mathbf{U}_1 \cdots \mathbf{U}_{k-1}$ , 且

$$\mathbf{A} = \left( \begin{array}{ccccc|ccccc} a_{11}^{(1)} & a_{12}^{(2)} & \cdots & a_{1,k-1}^{(k-1)} & a_{1k}^{(k)} & a_{1,k+1}^{(k)} & \cdots & a_{1n}^{(k)} & \\ -\sigma_1 & a_{22}^{(2)} & \cdots & a_{2,k-1}^{(k-1)} & a_{2k}^{(k)} & a_{2,k+1}^{(k)} & \cdots & a_{2n}^{(k)} & \\ & & \ddots & \vdots & \vdots & \vdots & & \vdots & \\ & & & -\sigma_{k-1} & a_{kk}^{(k)} & a_{k,k+1}^{(k)} & \cdots & a_{kn}^{(k)} & \\ \hline & & & & \mathbf{0} & a_{k+1,k}^{(k)} & a_{k+1,k+1}^{(k)} & \cdots & a_{k+1,n}^{(k)} \\ & & & & & \vdots & \vdots & \ddots & \vdots \\ & & & & & a_{nk}^{(k)} & a_{n,k+1}^{(k)} & \cdots & a_{nn}^{(k)} \end{array} \right) = \begin{pmatrix} k & n-k \\ \mathbf{A}_{11}^{(k)} & \mathbf{A}_{12}^{(k)} \\ \mathbf{0} & \mathbf{c}_k & \mathbf{A}_{22}^{(k)} \end{pmatrix}_{n-k},$$

其中  $\mathbf{c}_k = (a_{k+1,k}^{(k)}, \dots, a_{nk}^{(k)})^T \in \mathbb{R}^{n-k}$ ,  $\mathbf{A}_{11}^{(k)}$  为  $k$  阶上海森伯格矩阵,  $\mathbf{A}_{22}^{(k)} \in \mathbb{R}^{(n-k) \times (n-k)}$ .

## 10.2.2 用正交相似变换约化一般矩阵为上海森伯格矩阵

设  $\mathbf{c}_k \neq 0$ , 于是可选择初等反射矩阵  $\mathbf{R}_k$  使  $\mathbf{R}_k \mathbf{c}_k = -\sigma_k \mathbf{e}_1$ , 其中  $\mathbf{R}_k$  的计算公式为:

$$\sigma_k = \operatorname{sgn}(a_{k+1,k}^{(k)}) \left( \sum_{i=k+1}^n (a_{ik}^{(k)})^2 \right)^{\frac{1}{2}}, \mathbf{u}_k = \mathbf{c}_k + \sigma_k \mathbf{e}_1, \beta_k = \sigma_k (a_{k+1,k}^{(k)} + \sigma_k), \mathbf{R}_k = \mathbf{I} - \beta_k^{-1} \mathbf{u}_k \mathbf{u}_k^T.$$

令  $\mathbf{U}_k = \begin{pmatrix} \mathbf{I} \\ \mathbf{R}_k \end{pmatrix}$ , 则  $\mathbf{A}_{k+1} = \mathbf{U}_k \mathbf{A}_k \mathbf{U}_k = \begin{pmatrix} \mathbf{A}_{11}^{(k)} & \mathbf{A}_{12}^{(k)} \mathbf{R}_k \\ \mathbf{0} & \mathbf{R}_k \mathbf{A}_{22}^{(k)} \mathbf{R}_k \end{pmatrix} = \begin{pmatrix} \mathbf{A}_{11}^{(k+1)} & \mathbf{A}_{12}^{(k+1)} \\ \mathbf{0} & \mathbf{A}_{22}^{(k+1)} \end{pmatrix}$ , 其中  $\mathbf{A}_{11}^{(k+1)}$  为  $k+1$  阶上海森伯格矩阵, 第  $k$  次约化只需计算  $\mathbf{A}_{12}^{(k)} \mathbf{R}_k$  和  $\mathbf{R}_k \mathbf{A}_{22}^{(k)} \mathbf{R}_k$  (当  $\mathbf{A}$  为对称矩阵时, 只需计算  $\mathbf{R}_k \mathbf{A}_{22}^{(k)} \mathbf{R}_k$ ).

重复计算, 则有

$$\mathbf{U}_{n-2} \cdots \mathbf{U}_2 \mathbf{U}_1 \mathbf{A} \mathbf{U}_1 \mathbf{U}_2 \cdots \mathbf{U}_{n-2} = \begin{pmatrix} a_{11} & * & * & \cdots & * & * \\ -\sigma_1 & a_{22}^{(2)} & * & \cdots & * & * \\ & -\sigma_2 & a_{33}^{(3)} & \cdots & * & * \\ & & \ddots & \ddots & \vdots & \vdots \\ & & & -\sigma_{n-2} & a_{n-1,n-1}^{(n-2)} & * \\ & & & & -\sigma_{n-1} & a_{nn}^{(n-1)} \end{pmatrix} = \mathbf{A}_{n-1}.$$

## 10.2.3 上海森伯格矩阵QR算法

当  $\mathbf{A}$  为一般矩阵时, QR基本算法的计算量很大, 在实际使用时, 通常采用的做法是先将矩阵  $\mathbf{A}$  正交相似变换为上海森伯格矩阵, 然后再实施QR算法.

QR分解算法除Schmidt正交变换外, 还有豪斯霍尔德Householder变换法和吉文斯Givens变换法, 其原理和算法参考6.5 QR分解法. Householder变换法: 数值稳定, 适用于稠密矩阵, 时间复杂度  $2n^3/3$ ; Givens变换: 数值稳定, 适用于稀疏矩阵, 时间复杂度  $4n^3/3$ . Schmidt正交变换, 适合小矩阵计算, 在有限精度的病态矩阵中会导致大量误差.

给定精度  $\varepsilon > 0$  和最大迭代次数  $N$ ,  $\mathbf{A}^{(1)} = \text{hessenberg}(\mathbf{A})$ ,  $k = 1, 2, \dots, N$ , 计算:

$$\begin{cases} \mathbf{A}^{(k)} = \mathbf{Q}^{(k)} \mathbf{R}^{(k)}, \\ \mathbf{A}^{(k+1)} = \mathbf{R}^{(k)} \mathbf{Q}^{(k)}, \end{cases}$$

每一步迭代计算的  $\mathbf{Q}^{(k)}$  和  $\mathbf{R}^{(k)}$  都是上海森伯格矩阵.

## 10.2.3 上海森伯格矩阵QR算法

**例6:**用正交相似变换约化一般矩阵为上海森伯格矩阵,然后采用不同的QR正交分解法求解如下矩阵的全部特征值,精度要求 $\varepsilon = 10^{-15}$ ,最大迭代次数1000.

$$\mathbf{A} = \begin{pmatrix} 2 & 3 & 4 & 5 & 6 \\ 4 & 4 & 5 & 6 & 7 \\ 0 & 3 & 6 & 7 & 8 \\ 0 & 0 & 2 & 8 & 9 \\ 0 & 0 & 0 & 1 & 0 \end{pmatrix}$$

正交相似变换约化后的上海森伯格矩阵为

$$\mathbf{H} = \begin{pmatrix} 2 & -3 & -4 & -5 & 6 \\ -4 & 4 & 5 & 6 & -7 \\ 0 & 3 & 6 & 7 & -8 \\ 0 & 0 & 2 & 8 & -9 \\ 0 & 0 & 0 & -1 & 0 \end{pmatrix}$$

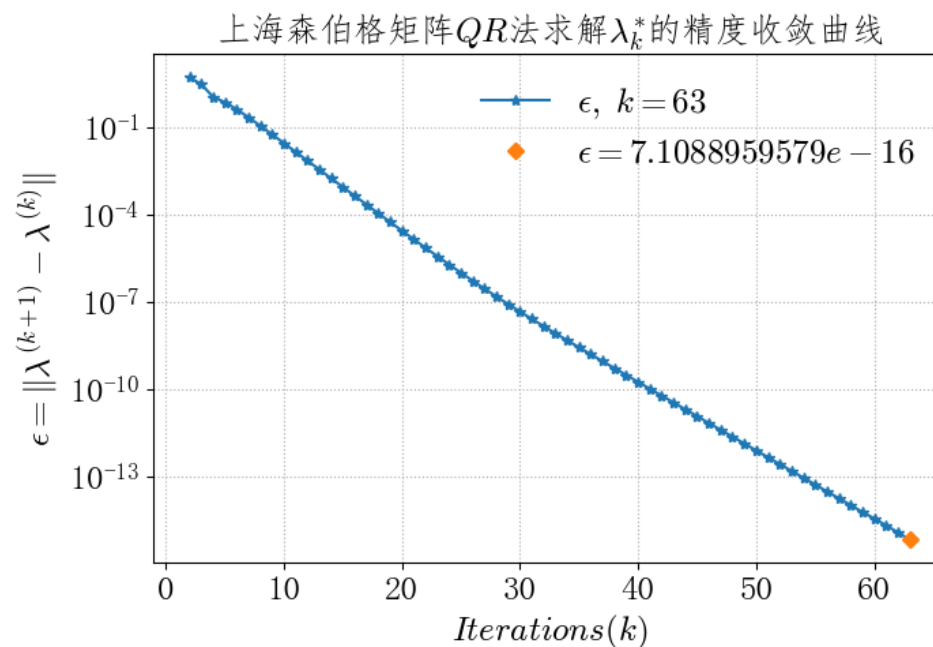
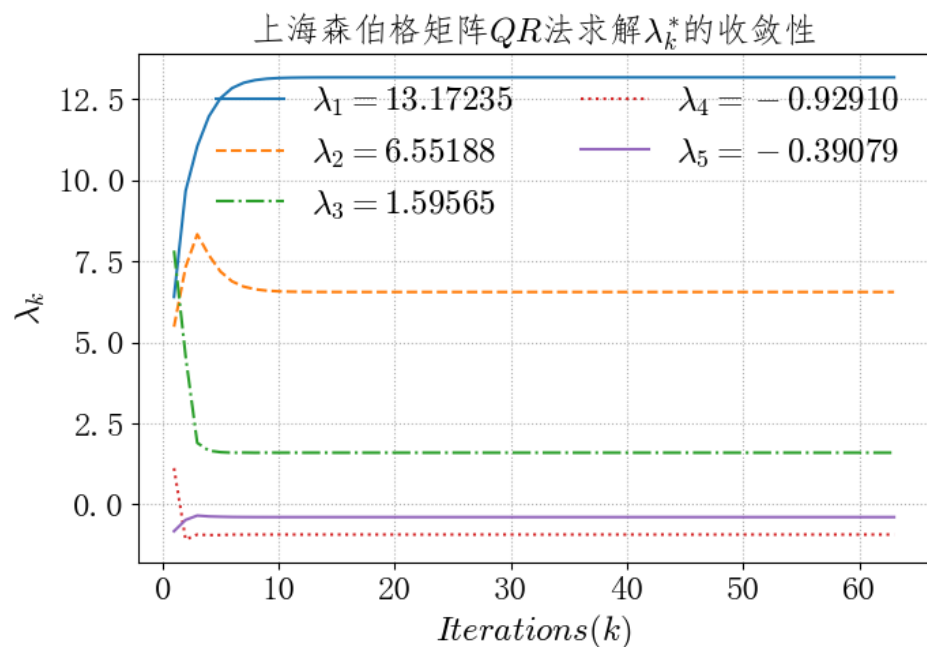


表 10-7 上海森伯格 QR 方法求解矩阵全部特征值结果与 np.linalg.eig 对比

QR 变换方法	迭代	矩阵全部特征值		
Schmidt, 针对 $\mathbf{H}$ 求解	457	13.172351398103398	6.551878351915636	1.595654573149940
		-0.929096277752297	-0.390788045416487	
Householder, 针对 $\mathbf{H}$ 求解	1000	13.172351157660193	6.551878592358674	1.595654574865366
		-0.929096280823645	-0.390788044060518	
Givens, 针对 $\mathbf{H}$ 求解	63	13.172351398103194	6.551878351915669	1.595654573149938
		-0.929096277752299	-0.390788045416489	
np.linalg.eig(A), 针对 $\mathbf{A}$ 求解		13.172351398103185	6.551878351915660	1.595654573149939
		-0.929096277752298	-0.390788045416488	

## 10.2.4 位移上海森伯格矩阵QR算法

位移QR算法加快QR算法的收敛速度，它类似于原点平移反幂法. 其算法的迭代过程如下：

给定精度  $\varepsilon > 0$  和最大迭代次数  $N$ ,  $\mathbf{A}^{(1)} = \text{hessenberg}(\mathbf{A})$ ,  $k = 1, 2, \dots, N$ , 选择  $\mu^{(k)}$ , 计算：

$$\begin{cases} \mathbf{A}^{(k)} - \mu^{(k)} \mathbf{I} = \mathbf{Q}^{(k)} \mathbf{R}^{(k)}, \\ \mathbf{A}^{(k+1)} = \mathbf{R}^{(k)} \mathbf{Q}^{(k)} + \mu^{(k)} \mathbf{I}. \end{cases}$$

一般  $\mu^{(k)}$  选择的方法有以下两种：

(1) 选  $\mu^{(k)} = a_{nn}^{(k)}$ , 即瑞利商位移；

(2) 迭代过程中，当子矩阵  $\begin{pmatrix} a_{n-1,n-1}^{(k)} & a_{n-1,n}^{(k)} \\ a_{n,n-1}^{(k)} & a_{n,n}^{(k)} \end{pmatrix}$  的两个特征值为实数时，选最接近  $a_{nn}^{(k)}$  的那个特征

值作为  $\mu^{(k)}$ , 即威尔金斯位移.

## 10.2.4 位移上海森伯格矩阵QR算法

针对例6示例矩阵  $\mathbf{A}_2$ : 用正交相似变换约化一般矩阵为上海森伯格矩阵, 然后采用位移QR方法求解矩阵的全部特征值, 精度要求  $\varepsilon = 10^{-15}$ , 最大迭代次数1000. 结果如表所示, 除位移SchmidtQR法收敛速度较慢之外, Householder和Givens方法均加快了收敛速度.

表 10-8 威尔金斯位移上海森伯格矩阵 QR 方法求解矩阵全部特征值结果

QR 变换方法	迭代	矩阵全部特征值		
Schmidt	1000	13.159532755266255	6.557612578805546	1.5955235439306743
		-0.3907559441976094	-0.9290962774025306	
Householder	35	13.172350869521098	6.551878880498133	1.5956545425533455
		-0.39078801482028047	-0.9290962777522952	
Givens	60	13.172351398103192	6.551878351915666	1.5956545731499354
		-0.39078804541648837	-0.9290962777522973	

## 10.2.4 位移上海森伯格矩阵QR算法

图为Wilkins位移上海森伯格QR分解(Householder)法求解矩阵的特征值的收敛曲线, 收敛过程较为平稳.

