数值线性代数与算法

第二章 正交变换和投影方法




Back
Close

向量序列的正交化和矩阵的正交 (相似) 变换在数值代数中占有独特的地位. QR 分解可用于线性方程组特别是最小二乘问题的求解. 而基于 QR 分解的 QR 方法可用于计算矩阵的特征值和特征向量. 此外, Krylov 子空间的正交化可用于构造线性方程组的迭代法. 投影方法的思想则是构造线性方程组迭代法和矩阵特征值计算的基础和出发点. 本章将主要介绍两种常用的正交变换、QR 分解、线性无关向量组的正交化以及 Krylov 子空间的性质及其正交化, 最后介绍投影方法的一般框架.

§2.1 两种常用的正交变换

§2.1.1 Householder 变换

定义 2.1 设 $\boldsymbol{u} \in \mathbb{R}^n$ 满足 $\|\boldsymbol{u}\|_2 = 1$, 称 n 阶矩阵

$$\boldsymbol{H} = \boldsymbol{I} - 2\boldsymbol{u}\boldsymbol{u}^{\mathrm{T}} \tag{2.1}$$

为 Householder 矩阵 (初等反射矩阵), $\boldsymbol{u}$ 称为 Householder 向量.

下面的定理给出了 Householder 矩阵的性质.

定理 2.1 设 $\boldsymbol{H}$ 为式 (2.1) 定义的 Householder 矩阵, 则

(1) $\det(\boldsymbol{H}) = -1$.

(2) $\boldsymbol{H}^{\mathrm{T}} = \boldsymbol{H}$, $\boldsymbol{H}^{\mathrm{T}}\boldsymbol{H} = \boldsymbol{I}$, $\boldsymbol{H}^{-1} = \boldsymbol{H}$, $\boldsymbol{H}^2 = \boldsymbol{I}$.

(3) $\boldsymbol{H}$ 仅有两个互不相同的特征值 -1 和 1, 且 -1 是单重的, 相应的特征向量为 $\boldsymbol{u}$. 而 1 是 $n-1$ 重的, 相应的特征向量为所有与 $\boldsymbol{u}$ 正交的非零向量.

(4) 设 $\boldsymbol{x}, \boldsymbol{u} \in \mathbb{R}^n$ 满足 $\boldsymbol{u}^{\mathrm{T}}\boldsymbol{x} = 0$, $\alpha \in \mathbb{R}$. 则

$$\boldsymbol{H}(\boldsymbol{x} + \alpha\boldsymbol{u}) = \boldsymbol{x} - \alpha\boldsymbol{u}.$$

证明 (1) 利用分块矩阵的乘法运算可以方便地验证 $\det(\boldsymbol{H}) = -1$. 事实上, 对

$$\begin{bmatrix} \boldsymbol{I} & \boldsymbol{0} \\ -\boldsymbol{u}^{\mathrm{T}} & 1 \end{bmatrix} \begin{bmatrix} \boldsymbol{I} & 2\boldsymbol{u} \\ \boldsymbol{0} & 1 \end{bmatrix} \begin{bmatrix} \boldsymbol{I} - 2\boldsymbol{u}\boldsymbol{u}^{\mathrm{T}} & \boldsymbol{0} \\ \boldsymbol{u}^{\mathrm{T}} & 1 \end{bmatrix}$$
$$= \begin{bmatrix} \boldsymbol{I} & \boldsymbol{0} \\ -\boldsymbol{u}^{\mathrm{T}} & 1 \end{bmatrix} \begin{bmatrix} \boldsymbol{I} & 2\boldsymbol{u} \\ \boldsymbol{u}^{\mathrm{T}} & 1 \end{bmatrix} = \begin{bmatrix} \boldsymbol{I} & 2\boldsymbol{u} \\ \boldsymbol{0} & 1 - 2\boldsymbol{u}^{\mathrm{T}}\boldsymbol{u} \end{bmatrix}$$

两端取行列式, 得 $\det(\boldsymbol{H}) = 1 - 2\boldsymbol{u}^{\mathrm{T}}\boldsymbol{u} = -1$.

(2) 利用 $\boldsymbol{H}$ 的定义, 容易验证这四个等式.

(3) 由于 $\boldsymbol{H}\boldsymbol{u} = (\boldsymbol{I} - 2\boldsymbol{u}\boldsymbol{u}^{\mathrm{T}})\boldsymbol{u} = -\boldsymbol{u}$, 故 -1 是 $\boldsymbol{H}$ 的一个特征值, 且几何重数至少为 1, $\boldsymbol{u}$ 为相应的特征向量. 另外, 对于任意与

Back

Close

$\boldsymbol{u}$ 正交的 n 维向量 $\boldsymbol{x}$, 有 $\boldsymbol{H}\boldsymbol{x} = \boldsymbol{x}$, 即 1 为 $\boldsymbol{H}$ 的一个特征值, 且几何重数至少为 $n-1$, 与 $\boldsymbol{u}$ 正交的任一非零向量作为相应的特征向量. 由于特征值的几何重数不超过代数重数, 故特征值 -1 与 1 的代数重数分别至少为 1 与 $n-1$, 其和至少为 n. 注意到矩阵的特征值代数重数之和不会超过 n, 故 -1 与 1 的代数重数分别刚好为 1 与 $n-1$.

(4) 直接计算

$$\boldsymbol{H}(\boldsymbol{x}+\alpha\boldsymbol{u}) = (\boldsymbol{I}-2\boldsymbol{u}\boldsymbol{u}^{\mathrm{T}})(\boldsymbol{x}+\alpha\boldsymbol{u}) = \boldsymbol{x}-\alpha\boldsymbol{u}.$$

证毕. □

下面的定理是 Householder 矩阵的另一个性质.

定理 2.2 设 $\boldsymbol{H}_{n-m}$ 是 $n-m$ 阶 Householder 矩阵, 则

$$\boldsymbol{H} = \begin{bmatrix} \boldsymbol{I}_m & \boldsymbol{O} \\ \boldsymbol{O} & \boldsymbol{H}_{n-m} \end{bmatrix}$$

是 n 阶 Householder 矩阵.

证明 设 $\boldsymbol{u}_{n-m}$ 是 $n-m$ 维单位列向量, 则有

$$\boldsymbol{H}_{n-m} = \boldsymbol{I}_{n-m} - 2\boldsymbol{u}_{n-m}\boldsymbol{u}_{n-m}^{\mathrm{T}},$$

$$
\begin{aligned}
\boldsymbol{H} &= \begin{bmatrix} \boldsymbol{I}_m & \boldsymbol{O} \\ \boldsymbol{O} & \boldsymbol{I}_{n-m} - 2\boldsymbol{u}_{n-m}\boldsymbol{u}_{n-m}^{\mathrm{T}} \end{bmatrix} \\
&= \begin{bmatrix} \boldsymbol{I}_m & \boldsymbol{O} \\ \boldsymbol{O} & \boldsymbol{I}_{n-m} \end{bmatrix} - 2 \begin{bmatrix} \boldsymbol{O} & \boldsymbol{O} \\ \boldsymbol{O} & \boldsymbol{u}_{n-m}\boldsymbol{u}_{n-m}^{\mathrm{T}} \end{bmatrix} \\
&= \begin{bmatrix} \boldsymbol{I}_m & \boldsymbol{O} \\ \boldsymbol{O} & \boldsymbol{I}_{n-m} \end{bmatrix} - 2 \begin{bmatrix} \boldsymbol{0} \\ \boldsymbol{u}_{n-m} \end{bmatrix} \begin{bmatrix} \boldsymbol{0} & \boldsymbol{u}_{n-m}^{\mathrm{T}} \end{bmatrix} \\
&= \boldsymbol{I}_n - 2\boldsymbol{u}_n\boldsymbol{u}_n^{\mathrm{T}},
\end{aligned}
$$

式中: $\boldsymbol{u}_n = \begin{bmatrix} \boldsymbol{0} \\ \boldsymbol{u}_{n-m} \end{bmatrix} \in \mathbb{R}^n$.

由于 $\boldsymbol{u}_n^{\mathrm{T}}\boldsymbol{u}_n = \boldsymbol{u}_{n-m}^{\mathrm{T}}\boldsymbol{u}_{n-m} = 1$, 所以 $\boldsymbol{H}$ 是 n 阶 Householder 矩阵. 证毕. □




Back
Close

定理 2.3 设 $\boldsymbol{x} = (x_1, x_2, \cdots, x_n)^{\mathrm{T}} \in \mathbb{R}^n (n > 1)$ 为任意的非零向量, 则存在 $\boldsymbol{u} \in \mathbb{R}^n$ 满足 $\|\boldsymbol{u}\|_2 = 1$, 使得式 (2.1) 定义的 Householder 矩阵满足

$$\boldsymbol{H}\boldsymbol{x} = \alpha \boldsymbol{e}_1, \tag{2.2}$$

式中: $\alpha = \pm\|\boldsymbol{x}\|_2$. 且使得式 (2.2) 成立的 $\boldsymbol{u}$ 在相差一个符号的意义下是唯一确定的.

证明 由定义 2.1, $\boldsymbol{H} = \boldsymbol{I} - 2\boldsymbol{u}\boldsymbol{u}^{\mathrm{T}}$, 于是 $\boldsymbol{H}\boldsymbol{x} = \boldsymbol{x} - 2(\boldsymbol{u}^{\mathrm{T}}\boldsymbol{x})\boldsymbol{u}$. 欲使式 (2.2) 成立, 必须满足 $2(\boldsymbol{u}^{\mathrm{T}}\boldsymbol{x})\boldsymbol{u} = \boldsymbol{x} - \alpha\boldsymbol{e}_1$. 由于 $\|\boldsymbol{u}\|_2 = 1$, 可取

$$\boldsymbol{u} = \frac{\boldsymbol{x} - \alpha\boldsymbol{e}_1}{\|\boldsymbol{x} - \alpha\boldsymbol{e}_1\|_2}.$$

又因 $\boldsymbol{H}$ 是正交矩阵, 为使式 (2.2) 成立, 必须有

$$\|\boldsymbol{x}\|_2 = \|\boldsymbol{H}\boldsymbol{x}\|_2 = \|\alpha\boldsymbol{e}_1\|_2 = |\alpha| \cdot \|\boldsymbol{e}_1\|_2 = |\alpha|,$$

即 $\alpha = \pm\|\boldsymbol{x}\|_2$. 容易验证, 如上选取的 $\boldsymbol{H}$ 满足式 (2.2). 事实上, 有

$$\begin{aligned}
\boldsymbol{H}\boldsymbol{x} &= \boldsymbol{x} - 2\frac{(\boldsymbol{x}-\alpha\boldsymbol{e}_1)(\boldsymbol{x}-\alpha\boldsymbol{e}_1)^{\mathrm{T}}}{\|\boldsymbol{x}-\alpha\boldsymbol{e}_1\|_2^2}\boldsymbol{x} \\
&= \boldsymbol{x} - \frac{2(\boldsymbol{x}-\alpha\boldsymbol{e}_1)^{\mathrm{T}}\boldsymbol{x}}{\|\boldsymbol{x}-\alpha\boldsymbol{e}_1\|_2^2}(\boldsymbol{x}-\alpha\boldsymbol{e}_1) \\
&= \boldsymbol{x} - \frac{2\|\boldsymbol{x}\|_2^2 - 2\alpha\boldsymbol{e}_1^{\mathrm{T}}\boldsymbol{x}}{\|\boldsymbol{x}\|_2^2 - 2\alpha\boldsymbol{e}_1^{\mathrm{T}}\boldsymbol{x} + \alpha^2}(\boldsymbol{x}-\alpha\boldsymbol{e}_1) \\
&= \boldsymbol{x} - \frac{2\alpha^2 - 2\alpha\boldsymbol{e}_1^{\mathrm{T}}\boldsymbol{x}}{\alpha^2 - 2\alpha\boldsymbol{e}_1^{\mathrm{T}}\boldsymbol{x} + \alpha^2}(\boldsymbol{x}-\alpha\boldsymbol{e}_1) \\
&= \boldsymbol{x} - (\boldsymbol{x}-\alpha\boldsymbol{e}_1) = \alpha\boldsymbol{e}_1.
\end{aligned}$$

此外, 由 $\boldsymbol{u}$ 的选取过程知, 这样的 $\boldsymbol{u}$ 在相差一个符号的意义下是唯一确定的. 证毕. □

注 2.1 由定理 2.3, 可按如下步骤来构造确定 $\boldsymbol{H}$ 的单位向量 $\boldsymbol{u}$:

(1) 计算 $\boldsymbol{v} = \boldsymbol{x} \pm \|\boldsymbol{x}\|_2 \boldsymbol{e}_1$.

(2) 计算 $\boldsymbol{u} = \boldsymbol{v}/\|\boldsymbol{v}\|_2$.

上述计算涉及 $\|\boldsymbol{x}\|_2$ 前的符号选取问题. 如果选取

$$\boldsymbol{v} = \boldsymbol{x} - \|\boldsymbol{x}\|_2 \boldsymbol{e}_1,$$

就会出现计算

$$v_1 = x_1 - \|\boldsymbol{x}\|_2$$

的问题, 其中 v_1, x_1 分别表示 $\boldsymbol{v}, \boldsymbol{x}$ 的第 1 个分量. 在 $x_1 > 0$ 时, 按上式计算 v_1 可能会导致有效数字的丢失. 在这种情形可改用下面的计算方式, 即

$$v_1 = x_1 - \|\boldsymbol{x}\|_2 = \frac{x_1^2 - \|\boldsymbol{x}\|_2^2}{x_1 + \|\boldsymbol{x}\|_2} = \frac{-(x_2^2 + \cdots + x_n^2)}{x_1 + \|\boldsymbol{x}\|_2},$$

并且只要在 $x_1 > 0$ 时使用这一公式计算, 就可以避免两个相近的数相减的情形.

注意到

$$\boldsymbol{H} = \boldsymbol{I} - 2\boldsymbol{u}\boldsymbol{u}^{\mathrm{T}} = \boldsymbol{I} - \frac{2}{\boldsymbol{v}^{\mathrm{T}}\boldsymbol{v}}\boldsymbol{v}\boldsymbol{v}^{\mathrm{T}} = \boldsymbol{I} - \beta\boldsymbol{v}\boldsymbol{v}^{\mathrm{T}},$$

式中: $\beta = 2/(\boldsymbol{v}^{\mathrm{T}}\boldsymbol{v})$. 这样就没有必要显式地求出向量 $\boldsymbol{u}$, 只需求出 β 和 $\boldsymbol{v}$ 即可. 在实际计算中, 往往将 $\boldsymbol{v}$ 归化为第 1 个分量为 1 的向量 (只需作变换 $\boldsymbol{v} := \boldsymbol{v}/v_1$ 即可). 这样做的优点是可以把 $\boldsymbol{v}$ 的后 $n-1$ 个分量保存在 $\boldsymbol{x}$ 的后 $n-1$ 个化为 0 的分量位置上, 而 $\boldsymbol{v}$ 的第一个分量 1 就无需保存了.

在计算时, 溢出现象也是必须要考虑的问题. 在上述计算中, 如果 $\boldsymbol{x}$ 某分量过大, 其平方运算可能会出现上溢的现象. 为了解决这个问题, 可以用 $\boldsymbol{x}/\|\boldsymbol{x}\|_\infty$ 代替 $\boldsymbol{x}$ 来构造 $\boldsymbol{v}$, 相当于在原来的 $\boldsymbol{v}$ 之前乘了一个常数 $\alpha = 1/\|\boldsymbol{x}\|_\infty$, 而 $\alpha\boldsymbol{v}$ 与 $\boldsymbol{v}$ 的单位化向量是相同的.

基于上述讨论, 可得如下基本算法.

Back

Close

算法 2.1 本算法计算实 Householder 矩阵 $\boldsymbol{H}=\boldsymbol{I}-\beta\boldsymbol{v}\boldsymbol{v}^{\mathrm{T}}$ 中满足 $v_1=1$ 的 $\boldsymbol{v}$ 和 β.

步 1, 输入 n 维实向量 $\boldsymbol{x}$. 计算 $\eta=\|\boldsymbol{x}\|_\infty$, 置 $\boldsymbol{x}:=\boldsymbol{x}/\eta$.

步 2, $\boldsymbol{v}:=[1,\boldsymbol{x}(2:n)]^{\mathrm{T}}$, 计算 $\sigma=x_2^2+\cdots+x_n^2$.

步 3, 对于 $\sigma=0$, 若 $x_1\geqslant 0$, $\beta:=0$, 否则 $\beta:=2$, 终止计算.

步 4, 对于 $\sigma>0$, 计算 $\alpha:=\sqrt{x_1^2+\sigma}$. 若 $x_1\leqslant 0$, $v_1=x_1-\alpha$, 否则 $v_1=-\sigma/(x_1+\alpha)$.

步 5, 计算 $\beta:=2v_1^2/(\sigma+v_1^2)$, $\boldsymbol{v}:=\boldsymbol{v}/v_1$.

执行算法 2.1 的运算量约为 $4n$. 进一步, 根据上述算法可编制 MATLAB 程序如下:

```
%实向量Householder变换程序-r_house.m
function [v,beta]=r_house(x)
%本函数计算Householder矩阵H=I-beta*v*v'
```

中满足v(1)=1的v和beta.

```
n=length(x);
eta=norm(x,inf); x=x/eta;
sigma=x(2:n)'*x(2:n);
v=[1; x(2:n)];
if sigma==0
     if x(1)>=0
         beta=0;
     else
         beta=2;
     end
else
    alpha=(x(1)^2+sigma)^0.5;
```




```
        if x(1)<=0
            v(1)=x(1)-alpha;
        else
            v(1)=-sigma/(x(1)+alpha);
        end
        beta=2*v(1)^2/(sigma+v(1)^2);  v=v/v(1);
    end
```

例 2.1 已知向量 $\boldsymbol{x}=(2,5,7,1)^{\mathrm{T}}$, 构造 Householder 矩阵 $\boldsymbol{H}$ 使得 $\boldsymbol{H}\boldsymbol{x}=\|\boldsymbol{x}\|_2\boldsymbol{e}_1$.

解 在 MATLAB 命令窗口依次运行如下代码:

```
>> x=[2,5,7,1]'; [v,beta]=r_house(x);
>> H=eye(length(x))-beta*v*v'; y=H*x
```

即得所需求的结果.




注 2.2 在应用 Householder 变换约化一个给定的矩阵为某一需要的形式时, 利用其特殊结构是非常重要的. 当计算 Householder 矩阵 $\boldsymbol{H} = \boldsymbol{I} - \beta \boldsymbol{v}\boldsymbol{v}^{\mathrm{T}} \in \mathbb{R}^{m\times m}$ 与一个已知矩阵 $\boldsymbol{A} \in \mathbb{R}^{m\times n}$ 的乘积时, 实际计算中 $\boldsymbol{H}$ 并不需要以显式的方式给出, 而是根据如下的公式来计算:

$$\boldsymbol{H}\boldsymbol{A} = (\boldsymbol{I} - \beta \boldsymbol{v}\boldsymbol{v}^{\mathrm{T}})\boldsymbol{A} = \boldsymbol{A} - \beta \boldsymbol{v}(\boldsymbol{A}^{\mathrm{T}}\boldsymbol{v})^{\mathrm{T}} = \boldsymbol{A} - \boldsymbol{v}\boldsymbol{w}^{\mathrm{T}},$$

式中: $\boldsymbol{w} = \beta \boldsymbol{A}^{\mathrm{T}}\boldsymbol{v}$. 换言之, 可以按下列两个步骤计算 $\boldsymbol{H}\boldsymbol{A}$:

(1) 计算 $\boldsymbol{w} = \beta \boldsymbol{A}^{\mathrm{T}}\boldsymbol{v}$.

(2) 计算 $\boldsymbol{B} = \boldsymbol{A} - \boldsymbol{v}\boldsymbol{w}^{\mathrm{T}}$.

矩阵 $\boldsymbol{B}$ 即为所求的乘积 $\boldsymbol{H}\boldsymbol{A}$. 完成这一计算任务所需要的运算量为 $4mn$.

注 2.3 在复数域情形, Householder 变换是指如下形式的矩阵

$$\boldsymbol{H}(\boldsymbol{w}) = \boldsymbol{I} - 2\boldsymbol{w}\boldsymbol{w}^{\mathrm{H}}, \tag{2.3}$$

式中: $\boldsymbol{w} \in \mathbb{C}^n$ 满足 $\boldsymbol{w}^{\mathrm{H}}\boldsymbol{w} = 1$. 相应于实 Householder 矩阵是正交矩阵, 复 Householder 矩阵是一个酉矩阵. 事实上, 复 Householder 变换除具备实 Householder 变换相应的 (1) ~ (4) 条性质外, 还有如下性质:

(5) 设 $\boldsymbol{x}, \boldsymbol{y} \in \mathbb{C}^n$, $\boldsymbol{x} \neq \boldsymbol{y}$. 则存在 Householder 变换 $\boldsymbol{H}(\boldsymbol{w})$ 使得 $\boldsymbol{H}(\boldsymbol{w})\boldsymbol{x} = \boldsymbol{y}$ 的充分必要条件是

$$\boldsymbol{x}^{\mathrm{H}}\boldsymbol{x} = \boldsymbol{y}^{\mathrm{H}}\boldsymbol{y}, \quad \boldsymbol{x}^{\mathrm{H}}\boldsymbol{y} = \boldsymbol{y}^{\mathrm{H}}\boldsymbol{x}.$$

并且在上述条件成立时, 所需的向量 $\boldsymbol{w}$ 可取为

$$\boldsymbol{w} = \mathrm{e}^{\mathrm{i}\theta}\frac{\boldsymbol{x} - \boldsymbol{y}}{\|\boldsymbol{x} - \boldsymbol{y}\|_2}, \quad \forall\, \theta \in \mathbb{R}.$$




Back
Close

由注 2.3, 若取 $\boldsymbol{x} = (x_1, x_2, \cdots, x_n)^{\mathrm{T}} \neq \boldsymbol{0}$, $\boldsymbol{y} = -\overline{\tau}\|\boldsymbol{x}\|_2 \boldsymbol{e}_1$, 其中

$$\tau = \begin{cases} 1, & \text{当 } x_1 = 0 \text{ 时}, \\ \dfrac{\overline{x}_1}{|x_1|}, & \text{当 } x_1 \neq 0 \text{ 时}, \end{cases} \tag{2.4}$$

则满足 $\boldsymbol{x}^{\mathrm{H}}\boldsymbol{x} = \boldsymbol{y}^{\mathrm{H}}\boldsymbol{y}$, $\boldsymbol{x}^{\mathrm{H}}\boldsymbol{y} = \boldsymbol{y}^{\mathrm{H}}\boldsymbol{x}$. 此时成立

$$\boldsymbol{H}(\boldsymbol{w})\boldsymbol{x} = -\overline{\tau}\|\boldsymbol{x}\|_2 \boldsymbol{e}_1,$$

式中: $\boldsymbol{w}$ 可取为

$$\boldsymbol{w} = \frac{\tau\boldsymbol{x} + \gamma\boldsymbol{e}_1}{\|\tau\boldsymbol{x} + \gamma\boldsymbol{e}_1\|_2}, \quad \gamma = \|\boldsymbol{x}\|_2.$$

而且易见当 $\boldsymbol{x}$ 为实向量时, 这样得到的 $\boldsymbol{H}(\boldsymbol{w})$ 是实对称正交矩阵.

基于以上讨论, 有下面的算法.

算法 2.2 给定一个向量 $\boldsymbol{x} \in \mathbb{C}^n$, 本算法计算一个向量 $\boldsymbol{w}$ 和一个数 γ, 使得 $\boldsymbol{H}\boldsymbol{x} = \gamma\boldsymbol{e}_1$, 其中 $\boldsymbol{H} = \boldsymbol{I} - \boldsymbol{w}\boldsymbol{w}^{\mathrm{H}}$, $\|\boldsymbol{w}\|_2 = \sqrt{2}$.

function $[\boldsymbol{w}, \gamma] = \mathbf{c\_house}(\boldsymbol{x})$

$\boldsymbol{w} = \boldsymbol{x};\ \gamma = \|\boldsymbol{x}\|_2;$

if $\gamma = 0$

$w(1) = \sqrt{2};$ 结束

end

if $w(1) \neq 0$

$\tau = \overline{w(1)}/|w(1)|;$

else

$\tau = 1;$

end

$\boldsymbol{w} = (\tau/\gamma)\boldsymbol{w};\quad w(1) = w(1) + 1;$

$\boldsymbol{w} = \boldsymbol{w}/\sqrt{w(1)};\quad \gamma = -\overline{\tau}\gamma;$

注 2.4 算法 2.2 是用类似于 MATLAB 语句给出的, 后文的许多算法也将采用这种形式给出. 因此, 熟悉 MATLAB 语言是很重要的. 可以说, 用类似于 MATLAB 语言的语句来描述算法往往要比用文字和公式描述算法精到得多.

注 2.5 算法 2.1 中的 Householder 变换只是针对实向量的, 而算法 2.2 既能用于实向量又能用于复向量. 这一点是很重要的, 因为许多实际问题, 如时谐涡流场的计算, 其离散形式是一个大型稀疏且具有特殊块结构的复线性方程组. 此时使用 Krylov 子空间方法 (如 GMRES 方法) 求解时, 需要用到针对复向量的 Householder 变换.

算法 2.2 的 MATLAB 程序如下:

```
%复向量的Householder变换程序-c_house.m
function [w,gamma]=c_house(x)
```

```
%给定复向量x，本函数计算一个向量w和一个数gamma
%满足H*x=gamma*e1，其中H=I-w*w'，||w||=sqrt(2).
w=x; gamma=norm(x);
if gamma==0
    w(1)=sqrt(2); return;
end
if w(1)==0
    tau=1;
else
    tau=conj(w(1))/abs(w(1));
end
w=(tau/gamma)*w; w(1)=w(1)+1;
w=w/sqrt(w(1)); gamma=-conj(tau)*gamma;
```




Back
Close

例 2.2 已知复向量 $\boldsymbol{x} = (2+\mathrm{i}, 5-3\mathrm{i}, 7, 1+2\mathrm{i})^{\mathrm{T}}$, 构造 Householder 矩阵 $\boldsymbol{H}$ 使得 $\boldsymbol{H}\boldsymbol{x} = \|\boldsymbol{x}\|_2\boldsymbol{e}_1$.

解 在 MATLAB 命令窗口依次运行如下代码:

```
>> x=[2+i,5-3*i,7,1+2*i]'; [w,gama]=c_house(x);
>> H=eye(length(x))-w*w'; y=H*x
```

即得所需求的结果.

§2.1.2 Givens 变换

Householder 变换可以将一个向量中若干相邻分量约化为 0. 但如果将向量中的某一个分量化为 0, 则采用 Givens 变换更为有效. Givens 变换 (矩阵) 定义如下.

定义 2.2 设实数 c 和 s 满足 $c^2+s^2=1$, 称 n 阶矩阵

$$\boldsymbol{G}_{ik}(c,s)=\begin{bmatrix} 1 & & & & & & \\ & \ddots & \vdots & & \vdots & & \\ \cdots & \cdots & c & \cdots & s & \cdots & \cdots \\ & & \vdots & \ddots & \vdots & & \\ \cdots & \cdots & -s & \cdots & c & \cdots & \cdots \\ & & \vdots & & \vdots & \ddots & \\ & & \vdots & & \vdots & & 1 \end{bmatrix} \begin{matrix} \\ \\ (i) \\ \\ (k) \\ \\ \end{matrix} \qquad (i\neq k) \qquad (2.5)$$

为 Givens 变换 (矩阵), 或初等旋转矩阵.

容易证明, Givens 矩阵具有如下性质.

定理 2.4 设 $\boldsymbol{G}$ 是由式 (2.5) 定义的 Givens 矩阵. 则

(1) $\boldsymbol{G}^{\mathrm{T}}\boldsymbol{G}=\boldsymbol{I}$, 即 $\boldsymbol{G}_{ik}(c,s)$ 为正交矩阵.

(2) $\det(\boldsymbol{G})=1$.

(3) $\boldsymbol{G}_{ik}(c,s)^{-1}=\boldsymbol{G}_{ik}(c,s)^{\mathrm{T}}=\boldsymbol{G}_{ik}(c,-s)$.

(4) 对于任意的 $\boldsymbol{x}\in\mathbb{R}^n$, Givens 变换 $\boldsymbol{y}=\boldsymbol{G}_{ik}(c,s)\boldsymbol{x}$ 只改变 $\boldsymbol{x}$ 的第 i,k 个分量.

证明　结论 (1) ∼ (3) 容易验证. 只证明结论 (4). 设 $\boldsymbol{x} = (x_1, x_2, \cdots, x_n)^{\mathrm{T}}$, $\boldsymbol{y} = (y_1, y_2, \cdots, y_n)^{\mathrm{T}}$, 则由 $\boldsymbol{y} = \boldsymbol{G}_{ik}(c, s)\boldsymbol{x}$, 得

$$y_i = cx_i + sx_k, \quad y_k = -sx_i + cx_k, \quad y_j = x_j, \quad (j \neq i, k).$$

证毕. □

由定理 2.4 的结论 (4) 不难发现, 当 $x_i^2 + x_k^2 \neq 0$ 时, 选取

$$c = \frac{x_i}{\sqrt{x_i^2 + x_k^2}}, \quad s = \frac{x_k}{\sqrt{x_i^2 + x_k^2}}, \tag{2.6}$$

可使 $y_i = \sqrt{x_i^2 + x_k^2} > 0$, $y_k = 0$. 由此, 给定 $\boldsymbol{x}$ 第 i 个和第 k 个分量, 可按下列步骤计算 c 和 s 使得 $\boldsymbol{G}_{ik}(c, s)\boldsymbol{x}$ 的第 k 个分量为 0.

算法 2.3 (计算实 Givens 变换) 给定两个实数 a 和 b, 本算法计算三个数 c, s, η, 使得

$$\begin{bmatrix} c & s \\ -s & c \end{bmatrix} \begin{bmatrix} a \\ b \end{bmatrix} = \begin{bmatrix} \eta \\ 0 \end{bmatrix}.$$

function $[c, s, \eta] = \mathbf{r\_givens}(a, b)$

if $b = 0$

$c = 1;\ s = 0;\ \eta = a;$ 结束

end

if $a = 0$

$c = 0;\ s = 1;\ \eta = b;$ 结束

end

if $|b| \geqslant |a|$

$t = a/b;\ s = 1/\sqrt{1 + t^2};\ c = st;\ \eta = |b|/s;$

else

$$t = b/a;\ \ c = 1/\sqrt{1+t^2};\ \ s = ct;\ \ \eta = |a|/c;$$

end

执行算法 2.3 只需 5 次乘除法、1 次加法和 1 次开平方运算. 如果不计算 η, 则可以减少一次除法运算. 进一步, 根据上述算法可编制 MATLAB 程序如下:

```
%Given变换程序-r_givens.m
function [c,s,eta]=r_givens(a,b)
%计算c,s满足[c s;-s c]*[a;b]=[eta;0]
if b==0, c=1; s=0; eta=a; end
if a==0, c=0; s=1; eta=b; end
if abs(b)>abs(a)
    t=a/b; s=1/sqrt(1+t^2); c=s*t;eta=abs(b)/s;
```

```
else
    t=b/a; c=1/sqrt(1+t^2); s=c*t;eta=abs(a)/c;
end
```

下面定理的结论表明 Givens 变换具有与 Householder 变换相同的功能.

定理 2.5 设 $\boldsymbol{x}=(x_1,x_2,\cdots,x_n)^{\mathrm{T}}\neq\boldsymbol{0}$, 则存在有限个 Givens 矩阵的乘积 $\boldsymbol{G}$, 使得 $\boldsymbol{G}\boldsymbol{x}=\|\boldsymbol{x}\|_2\boldsymbol{e}_1$.

证明 (1) 设 $x_1\neq 0$, 依次构造

$$\boldsymbol{G}_{12}:\ c=\frac{x_1}{\sqrt{x_1^2+x_2^2}},\quad s=\frac{x_2}{\sqrt{x_1^2+x_2^2}},$$

$$\Longrightarrow\quad \boldsymbol{G}_{12}\boldsymbol{x}=\left(\sqrt{x_1^2+x_2^2},0,x_3,\cdots,x_n\right)^{\mathrm{T}}.$$

$$\boldsymbol{G}_{13}:\ c=\frac{\sqrt{x_1^2+x_2^2}}{\sqrt{x_1^2+x_2^2+x_3^2}},\quad s=\frac{x_3}{\sqrt{x_1^2+x_2^2+x_3^2}},$$

$$\Longrightarrow\quad \boldsymbol{G}_{13}(\boldsymbol{G}_{12}\boldsymbol{x})=\left(\sqrt{x_1^2+x_2^2+x_3^2},0,0,x_4,\cdots,x_n\right)^{\mathrm{T}}.$$

$$\vdots$$

$$\boldsymbol{G}_{1n}:\ c=\frac{\sqrt{x_1^2+\cdots+x_{n-1}^2}}{\sqrt{x_1^2+\cdots+x_n^2}},\quad s=\frac{x_n}{\sqrt{x_1^2+\cdots+x_n^2}},$$

$$\Longrightarrow\quad \boldsymbol{G}_{1n}(\boldsymbol{G}_{1,n-1}\cdots\boldsymbol{G}_{13}\boldsymbol{G}_{12}\boldsymbol{x})=\left(\sqrt{x_1^2+\cdots+x_n^2},0,\cdots,0\right)^{\mathrm{T}}.$$

令 $\boldsymbol{G}=\boldsymbol{G}_{1n}\boldsymbol{G}_{1,n-1}\cdots\boldsymbol{G}_{13}\boldsymbol{G}_{12}$, 则有 $\boldsymbol{G}\boldsymbol{x}=\|\boldsymbol{x}\|_2\boldsymbol{e}_1$.

(2) 设 $x_1=\cdots=x_{i-1}=0$, $x_i\neq 0$ $(1<i\leqslant n)$, 则由 $\boldsymbol{G}_{1i}$ 开始即可. 证毕. □

与 Householder 矩阵类似, 用 Givens 矩阵左乘或右乘一个已知矩阵 $\boldsymbol{A}$, 利用其特殊结构也是极为有利的. 假定 $\boldsymbol{A} \in \mathbb{R}^{m\times n}$. 如果 $\boldsymbol{G}_{ik}(c,s) \in \mathbb{R}^{m\times m}$, 则用 $\boldsymbol{G}_{ik}(c,s)\boldsymbol{A}$ 修正 $\boldsymbol{A}$ 仅影响 $\boldsymbol{A}$ 的 i, k 两行, 可以用

$$\begin{bmatrix} a_{i1} & a_{i2} & \cdots & a_{in} \\ a_{k1} & a_{k2} & \cdots & a_{kn} \end{bmatrix} := \begin{bmatrix} c & s \\ -s & c \end{bmatrix} \begin{bmatrix} a_{i1} & a_{i2} & \cdots & a_{in} \\ a_{k1} & a_{k2} & \cdots & a_{kn} \end{bmatrix}$$

实现, 这仅需 $6n$ 个 flop (浮点运算). 实现上述运算的 MATLAB 程序段如下:

```
for j=1:n
    t1=A(i,j); t2=A(k,j);
    A(i,j)=c*t1+s*t2;
    A(k,j)=-s*t1+c*t2;
end
```




Back
Close

同样, 如果 $\boldsymbol{G}_{ik}(c,s) \in \mathbb{R}^{n\times n}$, 则用 $\boldsymbol{AG}_{ik}(c,s)$ 修正 $\boldsymbol{A}$ 仅影响 $\boldsymbol{A}$ 的 i, k 两列, 可以用

$$\begin{bmatrix} a_{1i} & a_{1k} \\ a_{2i} & a_{2k} \\ \vdots & \vdots \\ a_{mi} & a_{mk} \end{bmatrix} := \begin{bmatrix} a_{1i} & a_{1k} \\ a_{2i} & a_{2k} \\ \vdots & \vdots \\ a_{mi} & a_{mk} \end{bmatrix} \begin{bmatrix} c & s \\ -s & c \end{bmatrix}$$

实现, 这只需 $6m$ 个 flop. 实现上述运算的 MATLAB 程序段如下:

```
for j=1:m
    t1=A(j,i); t2=A(j,k);
    A(j,i)=c*t1-s*t2;
    A(j,k)=s*t1+c*t2;
end
```

注 2.6 复数情形的 Givens 矩阵是指定义 2.2 中的 c 和 s 为复数且满足 $|c|^2 + |s|^2 = 1$. 此时的 Givens 矩阵 $\boldsymbol{G}_{ik}(c, s)$ 是一个酉矩阵. 设 $\boldsymbol{x} = (x_1, x_2, \cdots, x_n)^{\mathrm{T}} \in \mathbb{C}^n$, $\boldsymbol{y} = \boldsymbol{G}_{ik}(c, s)\boldsymbol{x} = (y_1, y_2, \cdots, y_n)^{\mathrm{T}}$. 容易验证, 当 $|x_i|^2 + |x_k|^2 \neq 0$ 时, 若取

$$c = \frac{|x_i|}{\sqrt{|x_i|^2 + |x_k|^2}}, \quad s = \frac{\overline{x}_k}{\sqrt{|x_i|^2 + |x_k|^2}} \frac{x_i}{|x_i|}, \tag{2.7}$$

则有

$$y_s = x_s, \ s \neq i, k, \quad y_i = \frac{x_i}{|x_i|}\sqrt{|x_i|^2 + |x_k|^2}, \quad y_k = 0, \tag{2.8}$$

即可以通过复 Givens 变换将复向量 $\boldsymbol{x}$ 的第 k 个分量化为零.

根据式 (2.7), 可设计算法如下.

算法 2.4 (计算复 Givens 变换) 给定两个复数 a 和 b, 本算法计算三个数 c, s, η, 使得

$$\begin{bmatrix} c & s \\ -\bar{s} & \bar{c} \end{bmatrix} \begin{bmatrix} a \\ b \end{bmatrix} = \begin{bmatrix} \eta \\ 0 \end{bmatrix}.$$

function $[c, s, \eta] = \mathbf{c\_givens}(a, b)$

if $b = 0$

$c = 1;\ s = 0;\ \eta = a;$ 结束

end

if $a = 0$

$c = 0;\ s = 1;\ \eta = b;$ 结束

end

$u = a/|a|;\ t = \sqrt{|a|^2 + |b|^2};$

$c = |a|/t;\ s = u\bar{b}/t;\ \eta = ut;$

算法 2.4 的 MATLAB 程序如下:

```
%复Given变换程序-c_givens.m
function [c,s,eta]=c_givens(a,b)
%计算c,s满足[c,s;-s',c']*[a;b]=[eta;0]
if b==0, c=1; s=0; eta=a; end
if a==0, c=0; s=1; eta=b; end
u=a/abs(a); t=sqrt(abs(a)^2+abs(b)^2);
c=abs(a)/t; s=u*conj(b)/t; eta=u*t;
```




§2.2 QR 分解

实现矩阵 QR 分解最常用的方法有三种, 分别是 Householder 正交化方法、Givens 正交化方法和 Gram–Schmidt 正交化方法. 本节先给出前两种方法的具体实现.

定义 2.3 设 $\boldsymbol{A} \in \mathbb{R}^{m\times n}\,(m \geqslant n)$ 为列满秩矩阵, 若有正交矩阵 $\boldsymbol{Q} \in \mathbb{R}^{m\times m}$ 与上三角矩阵 $\boldsymbol{R} \in \mathbb{R}^{m\times n}$ 使得 $\boldsymbol{A} = \boldsymbol{QR}$, 则称 $\boldsymbol{QR}$ 为 $\boldsymbol{A}$ 的 QR 分解.

§2.2.1 Householder 变换 QR 分解

本节介绍如何使用 Householder 变换求矩阵的 QR 分解.




Back
Close

定理 2.6 设 $\boldsymbol{A} \in \mathbb{R}^{m\times n}\,(m \geqslant n)$ 为列满秩矩阵, 则存在有限个 Householder 矩阵的乘积 $\boldsymbol{Q}$, 使得

$$\boldsymbol{A} = \boldsymbol{Q}\widetilde{\boldsymbol{R}} = \boldsymbol{Q}\begin{bmatrix}\boldsymbol{R}\\ \boldsymbol{O}\end{bmatrix}, \tag{2.9}$$

式中: $\boldsymbol{R} \in \mathbb{R}^{n\times n}$ 为非奇异的上三角矩阵.

证明 取 $\boldsymbol{A}_1 := \boldsymbol{A}$. 因为 $\boldsymbol{A} = (a_{ij}) \in \mathbb{R}^{m\times n}$ 列满秩, 故 $\boldsymbol{A}$ 的第 1 列 $\boldsymbol{a}_1 \neq \boldsymbol{0}$. 于是可以构造 Householder 矩阵 $\boldsymbol{H}_1 \in \mathbb{R}^{m\times m}$, 使得 $\boldsymbol{H}_1\boldsymbol{a}_1 = \alpha_1\boldsymbol{e}_1$, 这里 $\alpha_1 = \|\boldsymbol{a}_1\|_2$, $\boldsymbol{e}_1 \in \mathbb{R}^m$. 则有

$$\boldsymbol{A}_2 = \boldsymbol{H}_1\boldsymbol{A}_1 = \begin{bmatrix}\alpha_1 & *\\ \boldsymbol{0} & \boldsymbol{A}_{22}^{(2)}\end{bmatrix}.$$

易见 $\boldsymbol{A}_{22}^{(2)} \in \mathbb{R}^{(m-1)\times(n-1)}$ 列满秩. 对 $\boldsymbol{A}_{22}^{(2)}$ 第 1 列的 $m-1$ 维非零向量 $\boldsymbol{a}_2$ 构造 $m-1$ 阶 Householder 矩阵 $\widetilde{\boldsymbol{H}}_2$, 使得 $\widetilde{\boldsymbol{H}}_2\boldsymbol{a}_2 = \alpha_2\boldsymbol{e}_1$,




Back
Close

这里 $\alpha_2 = \|\boldsymbol{a}_2\|_2$, $\boldsymbol{e}_1 \in \mathbb{R}^{m-1}$. 令 $\boldsymbol{H}_2 = \mathrm{diag}(1, \widetilde{\boldsymbol{H}}_2)$, 则有

$$\boldsymbol{A}_3 = \boldsymbol{H}_2\boldsymbol{A}_2 = \begin{bmatrix} \alpha_1 & * & * & \cdots & * \\ 0 & \alpha_2 & * & \cdots & * \\ 0 & 0 & * & \cdots & * \\ \vdots & \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & * & \cdots & * \end{bmatrix} = \begin{bmatrix} \boldsymbol{A}_{11}^{(3)} & \boldsymbol{A}_{12}^{(3)} \\ \boldsymbol{O} & \boldsymbol{A}_{22}^{(3)} \end{bmatrix} \begin{matrix} 2 \\ m-2 \end{matrix}, \quad \begin{matrix} 2 & n-2 \end{matrix}$$

式中: $\boldsymbol{A}_{11}^{(3)}$ 为上三角矩阵.

重复上述过程, 假定已经进行了 $k-1$ 步, 得到了 Householder 变换 $\boldsymbol{H}_1, \boldsymbol{H}_2, \cdots, \boldsymbol{H}_{k-1}$, 使得

$$\boldsymbol{A}_k = \boldsymbol{H}_{k-1}\cdots\boldsymbol{H}_2\boldsymbol{H}_1\boldsymbol{A}_1 = \begin{bmatrix} \boldsymbol{A}_{11}^{(k)} & \boldsymbol{A}_{12}^{(k)} \\ \boldsymbol{O} & \boldsymbol{A}_{22}^{(k)} \end{bmatrix} \begin{matrix} k-1 \\ m-k+1 \end{matrix}, \quad \begin{matrix} k-1 & n-k+1 \end{matrix}$$

式中: $\boldsymbol{A}_{11}^{(k)}$ 为上三角矩阵.

假定 $\boldsymbol{A}_{22}^{(k)} = \left[\boldsymbol{a}_k, \boldsymbol{a}_{k+1}, \cdots, \boldsymbol{a}_n\right]$. 第 k 步是先对非零向量 $\boldsymbol{a}_k$, 构造

$$\widetilde{\boldsymbol{H}}_k = \boldsymbol{I}_{m-k+1} - \beta_k \boldsymbol{v}_k \boldsymbol{v}_k^{\mathrm{T}} \in \mathbb{R}^{(m-k+1)\times(m-k+1)},$$

使得 $\widetilde{\boldsymbol{H}}_k \boldsymbol{a}_k = \alpha_k \boldsymbol{e}_1$, 这里 $\alpha_k = \|\boldsymbol{a}_k\|_2$, $\boldsymbol{e}_1 \in \mathbb{R}^{m-k+1}$. 令 $\boldsymbol{H}_k = \mathrm{diag}(\boldsymbol{I}_{k-1}, \widetilde{\boldsymbol{H}}_k)$, 则

$$\boldsymbol{A}_{k+1} = \boldsymbol{H}_k \boldsymbol{A}_k = \begin{bmatrix} \boldsymbol{A}_{11}^{(k)} & \boldsymbol{A}_{12}^{(k)} \\ \boldsymbol{O} & \widetilde{\boldsymbol{H}}_k \boldsymbol{A}_{22}^{(k)} \end{bmatrix} = \begin{array}{c} \\ \begin{bmatrix} \boldsymbol{A}_{11}^{(k+1)} & \boldsymbol{A}_{12}^{(k+1)} \\ \boldsymbol{O} & \boldsymbol{A}_{22}^{(k+1)} \end{bmatrix} \\ \begin{array}{cc} k & n-k \end{array} \end{array} \begin{array}{l} \\ k \\ m-k \end{array},$$

式中: $\boldsymbol{A}_{11}^{(k+1)}$ 为上三角矩阵.

这样, 从 $k = 1$ 出发, 对 $\boldsymbol{A}$ 依次进行 n 次 Householder 变换, 就可将 $\boldsymbol{A}$ 约化为上三角矩阵 $\widetilde{\boldsymbol{R}}$, 即

$$\widetilde{\boldsymbol{R}} = \boldsymbol{A}_{n+1} = \boldsymbol{H}_n \boldsymbol{A}_n = \begin{bmatrix} \boldsymbol{A}_{11}^{(n+1)} \\ \boldsymbol{O} \end{bmatrix} \begin{matrix} n \\ m-n \end{matrix},$$

式中: $\boldsymbol{A}_{11}^{(n+1)}$ 为上三角矩阵.

现记 $\boldsymbol{R} = \boldsymbol{A}_{11}^{(n+1)}$, $\boldsymbol{Q}^{\mathrm{T}} = \boldsymbol{H}_n \boldsymbol{H}_{n-1} \cdots \boldsymbol{H}_1$, 则

$$\boldsymbol{A} = \boldsymbol{Q} \begin{bmatrix} \boldsymbol{R} \\ \boldsymbol{O} \end{bmatrix},$$

式中: $\boldsymbol{R} \in \mathbb{R}^{n \times n}$ 为上三角矩阵.

若令 $\boldsymbol{Q} = \big[\boldsymbol{Q}_1, \boldsymbol{Q}_2\big]$, 这里 $\boldsymbol{Q}_1 \in \mathbb{R}^{m \times n}$, 则有

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

证毕. □

注 2.7 若 $\boldsymbol{A} \in \mathbb{R}^{n\times n}$ 为非奇异矩阵, 则由定理 2.6, $\boldsymbol{A}$ 有 QR 分解

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

式中: $\boldsymbol{Q} \in \mathbb{R}^{n\times n}$ 为正交矩阵; $\boldsymbol{R} \in \mathbb{R}^{n\times n}$ 为非奇异的上三角矩阵.

下面考虑计算 $\boldsymbol{A}$ 的 QR 分解的存储问题. 一般来说, 在完成 QR 分解之后 $\boldsymbol{A}$ 就不再需要, 可用它来存放 $\boldsymbol{Q}$ 和 $\boldsymbol{R}$. 通常不必将 $\boldsymbol{Q}$ 显式地算出, 而只存放构成它的 n 个 Householder 矩阵 $\boldsymbol{H}_k (k = 1, 2, \cdots, n)$, 而对每个 $\boldsymbol{H}_k$, 只需保存 $\boldsymbol{v}_k$ 和 β_k 即可. 注意到 $\boldsymbol{v}_k$ 具有如下形式:

$$\boldsymbol{v}_k = \left(1, v_{k+1}^{(k)}, \cdots, v_n^{(k)}\right)^{\mathrm{T}} \in \mathbb{R}^{m-k+1},$$

正好可以将 $\boldsymbol{v}_k$ 的第 2 到 $m-k+1$ 个分量, 即 $\boldsymbol{v}_k(2 : m-k+1)$ 存放在 $\boldsymbol{A}$ 的第 k 列对角元以下的位置上, 而 $\boldsymbol{A}$ 的上三角部分用来存放 $\boldsymbol{R}$ 的上三角部分.

综合上述讨论, 可得如下算法.

算法 2.5 (Householder 变换 QR 分解)

for $k = 1 : n$

  if $k < m$

    $[v, \beta] = \mathbf{r\_house}(\boldsymbol{A}(k : m, k))$;

    $\boldsymbol{A}(k : m, k : n) = (\boldsymbol{I}_{m-k+1} - \beta \boldsymbol{v}\boldsymbol{v}^{\mathrm{T}})\boldsymbol{A}(k : m, k : n)$;

    $d(k) = \beta$; $\boldsymbol{A}(k + 1 : m, k) = \boldsymbol{v}(2 : m - k + 1)$;

  end

end

注 2.8 算法 2.5 是指对于给定的矩阵 $\boldsymbol{A} \in \mathbb{R}^{m\times n}\,(m \geqslant n)$, 计算 Householder 矩阵 $\boldsymbol{H}_1, \boldsymbol{H}_2, \cdots, \boldsymbol{H}_n$ 满足: 如果 $\boldsymbol{Q} = \boldsymbol{H}_1\boldsymbol{H}_2\cdots\boldsymbol{H}_n$, 则 $\boldsymbol{Q}^{\mathrm{T}}\boldsymbol{A} = \boldsymbol{R}$ 是上三角形矩阵, 且 $\boldsymbol{A}$ 的上三角部分被 $\boldsymbol{R}$ 的上三角部分所覆盖, 第 k 个 Householder 向量的第 $k+1$ 到第 m 个分量存放于 $\boldsymbol{A}(k+1 : m, k)$, $k < m$ 的位置. 容易计算出, 该算法的运算量为 $2n^2(m - n/3)$ 个 flop.

根据上述算法可编制 MATLAB 程序如下:

```
function [A,d]=house_qr(A)
%Householder变换QR分解，不显式计算和存储正交矩阵Q
[m,n]=size(A);
for k=1:n
    if k<m
        [v,beta]=r_house(A(k:m,k));
        I=eye(m-k+1);
        A(k:m,k:n)=(I-beta*v*v')*A(k:m,k:n);
        d(k)=beta;  A(k+1:m,k)=v(2:m-k+1);
    end
end
```

例 2.3 利用算法 2.5 对矩阵 $\boldsymbol{A}$ 进行 QR 分解, 其中

$$\boldsymbol{A} = \begin{bmatrix} 76 & 96 & 85 & 35 \\ 26 & 55 & 26 & 20 \\ 51 & 14 & 82 & 26 \\ 70 & 15 & 25 & 62 \\ 90 & 26 & 93 & 48 \end{bmatrix}.$$

解 由于算法 2.5 中没有显式计算和存储正交矩阵 $\boldsymbol{Q}$, 故需对程序 house_qr.m 稍作修改:

```
function [Q,R]=Qhouse_qr(A)
%Householder QR分解, 显式计算并存储正交矩阵Q
[m,n]=size(A); Q=eye(m);
for k=1:n
    if k<m
```

```
            [v,beta]=r_house(A(k:m,k));
            H=eye(m-k+1)-beta*v*v';
            A(k:m,k:n)=H*A(k:m,k:n);
            Q=Q*blkdiag(eye(k-1),H);
        end
    end
    R=triu(A);
```

然后再在命令窗口依次运行如下代码:

```
>> A=[76,96,85,35;26,55,26,20; ...
>>    51,14,82,26;70,15,25,62;90,26,93,48];
>> [Q,R]=Qhouse_qr(A)
```

即得矩阵 $\boldsymbol{A}$ 的 QR 分解, 且满足 $\|\boldsymbol{A}-\boldsymbol{QR}\|_2 = 4.3169 \times 10^{-14}$.




§2.2.2 Givens 变换 QR 分解

Givens 变换也可以实现矩阵 $\boldsymbol{A}$ 的 QR 分解. 先考虑 $\boldsymbol{A}$ 为非奇异实方阵的情形, 有下面的定理.

定理 2.7 设 $\boldsymbol{A} \in \mathbb{R}^{n\times n}$ 为非奇异矩阵, 则存在有限个 Givens 矩阵的乘积 $\boldsymbol{Q}$, 使得 $\boldsymbol{Q}^{\mathrm{T}}\boldsymbol{A}$ 为可逆上三角矩阵 $\boldsymbol{R}$, 即 $\boldsymbol{A} = \boldsymbol{Q}\boldsymbol{R}$.

证明 因为 $\boldsymbol{A} = (a_{ij}) \in \mathbb{R}^{n\times n}$ 可逆, 故 $\boldsymbol{A}$ 的第 1 列 $\boldsymbol{a}^{(0)} \neq \boldsymbol{0}$. 于是可以构造有限个 Givens 矩阵的乘积 $\boldsymbol{G}_0 \in \mathbb{R}^{n\times n}$, 使得 $\boldsymbol{G}_0\boldsymbol{a}^{(0)} = \|\boldsymbol{a}^{(0)}\|_2\boldsymbol{e}_1$, 这里 $\boldsymbol{e}_1 \in \mathbb{R}^n$. 记 $a_{11}^{(1)} = \|\boldsymbol{a}^{(0)}\|_2$, 则有

$$\boldsymbol{G}_0\boldsymbol{A} = \left[\begin{array}{c|ccc} a_{11}^{(1)} & a_{12}^{(1)} & \cdots & a_{1n}^{(1)} \\ \hline 0 & & & \\ \vdots & & \boldsymbol{A}^{(1)} & \\ 0 & & & \end{array}\right].$$

易见 $\boldsymbol{A}^{(1)}$ 可逆. 于是 $\boldsymbol{A}^{(1)}$ 的第 1 列 $\boldsymbol{a}^{(1)} \neq \boldsymbol{0}$. 故存在有限个 Givens 矩阵的乘积 $\boldsymbol{G}_1 \in \mathbb{R}^{(n-1)\times(n-1)}$, 使得 $\boldsymbol{G}_1\boldsymbol{a}^{(1)} = \|\boldsymbol{a}^{(1)}\|_2\boldsymbol{e}_1$, 这里 $\boldsymbol{e}_1 \in \mathbb{R}^{n-1}$. 记 $a_{22}^{(2)} = \|\boldsymbol{a}^{(1)}\|_2$, 则有

$$\boldsymbol{G}_1\boldsymbol{A}^{(1)} = \left[\begin{array}{c|ccc} a_{22}^{(2)} & a_{23}^{(2)} & \cdots & a_{2n}^{(2)} \\ \hline 0 & & & \\ \vdots & & \boldsymbol{A}^{(2)} & \\ 0 & & & \end{array}\right].$$

依此类推, 最后得到 $\boldsymbol{A}^{(n-2)} \in \mathbb{R}^{2\times 2}$ 可逆, $\boldsymbol{A}^{(n-2)}$ 的第 1 列 $\boldsymbol{a}^{(n-2)} \neq \boldsymbol{0}$. 那么可构造 Givens 矩阵 $\boldsymbol{G}_{n-2} \in \mathbb{R}^{2\times 2}$, 使得 $\boldsymbol{G}_{n-2}\boldsymbol{a}^{(n-2)} = \|\boldsymbol{a}^{(n-2)}\|_2\boldsymbol{e}_1$, 这里 $\boldsymbol{e}_1 \in \mathbb{R}^2$. 记 $a_{n-1,n-1}^{(n-1)} = \|\boldsymbol{a}^{(n-2)}\|_2$, 则有

$$\boldsymbol{G}_{n-2}\boldsymbol{A}^{(n-2)} = \begin{bmatrix} a_{n-1,n-1}^{(n-1)} & a_{n-1,n}^{(n-1)} \\ 0 & a_{n,n}^{(n-1)} \end{bmatrix}.$$

令

$$\boldsymbol{Q}^{\mathrm{T}} = \begin{bmatrix} \boldsymbol{I}_{n-2} & \\ & \boldsymbol{G}_{n-2} \end{bmatrix} \cdots \begin{bmatrix} \boldsymbol{I}_2 & \\ & \boldsymbol{G}_2 \end{bmatrix} \begin{bmatrix} 1 & \\ & \boldsymbol{G}_1 \end{bmatrix} \boldsymbol{G}_0,$$

则 $\boldsymbol{Q}$ 仍为有限个 Givens 矩阵之积, 且有

$$\boldsymbol{Q}^{\mathrm{T}}\boldsymbol{A} = \begin{bmatrix} a_{11}^{(1)} & a_{12}^{(1)} & \cdots & a_{1,n-1}^{(1)} & a_{1n}^{(1)} \\ & a_{22}^{(2)} & \cdots & a_{2,n-1}^{(2)} & a_{2n}^{(2)} \\ & & \ddots & \vdots & \vdots \\ & & & a_{n-1,n-1}^{(n-1)} & a_{n-1,n}^{(n-1)} \\ & & & & a_{nn}^{(n-1)} \end{bmatrix} := \boldsymbol{R},$$

即 $\boldsymbol{A} = \boldsymbol{Q}\boldsymbol{R}$. 证毕. □

定理 2.7 的结论可以推广到 $\boldsymbol{A}$ 是长方阵的情形.

定理 2.8 设 $\boldsymbol{A} \in \mathbb{R}^{m\times n}\,(m \geqslant n)$ 是列满秩的, 则有 m 阶正

交矩阵 $\boldsymbol{Q}$ 及 n 阶上三角矩阵 $\boldsymbol{R}$, 使得

$$\boldsymbol{A} = \boldsymbol{Q}\begin{bmatrix}\boldsymbol{R}\\ \boldsymbol{O}\end{bmatrix} = \boldsymbol{Q}_1\boldsymbol{R},$$

式中: $\boldsymbol{Q}_1$ 为由 $\boldsymbol{Q}$ 的前 n 列构成的矩阵.

证明 已知 $\boldsymbol{A}$ 是列满秩的, 即它的 n 个列向量线性无关, 使用与证明定理 2.7 相同的方法, 可找到有限个 m 阶 Givens 矩阵的乘积 $\boldsymbol{G}$, 使得 $\boldsymbol{GA} = \begin{bmatrix}\boldsymbol{R}\\ \boldsymbol{O}\end{bmatrix}$, 再令 $\boldsymbol{Q} = \boldsymbol{G}^{\mathrm{T}}$ 即得所求. 证毕. □

下面考虑 Givens 变换 QR 分解的实现过程. 可以用一个 5×3 阶矩阵的例子来表明其一般思想.

$$\begin{bmatrix} \times & \times & \times \\ \times & \times & \times \\ \times & \times & \times \\ \times & \times & \times \\ \times & \times & \times \end{bmatrix} \xrightarrow{(4,5)} \begin{bmatrix} \times & \times & \times \\ \times & \times & \times \\ \times & \times & \times \\ \times & \times & \times \\ 0 & \times & \times \end{bmatrix} \xrightarrow{(3,4)} \begin{bmatrix} \times & \times & \times \\ \times & \times & \times \\ \times & \times & \times \\ 0 & \times & \times \\ 0 & \times & \times \end{bmatrix} \xrightarrow{(2,3)}$$

$$\begin{bmatrix} \times & \times & \times \\ \times & \times & \times \\ 0 & \times & \times \\ 0 & \times & \times \\ 0 & \times & \times \end{bmatrix} \xrightarrow{(1,2)} \begin{bmatrix} \times & \times & \times \\ 0 & \times & \times \\ 0 & \times & \times \\ 0 & \times & \times \\ 0 & \times & \times \end{bmatrix} \xrightarrow{(4,5)} \begin{bmatrix} \times & \times & \times \\ 0 & \times & \times \\ 0 & \times & \times \\ 0 & \times & \times \\ 0 & 0 & \times \end{bmatrix} \xrightarrow{(3,4)}$$

$$\begin{bmatrix} \times & \times & \times \\ 0 & \times & \times \\ 0 & \times & \times \\ 0 & 0 & \times \\ 0 & 0 & \times \end{bmatrix} \xrightarrow{(2,3)} \begin{bmatrix} \times & \times & \times \\ 0 & \times & \times \\ 0 & 0 & \times \\ 0 & 0 & \times \\ 0 & 0 & \times \end{bmatrix} \xrightarrow{(4,5)} \begin{bmatrix} \times & \times & \times \\ 0 & \times & \times \\ 0 & 0 & \times \\ 0 & 0 & \times \\ 0 & 0 & 0 \end{bmatrix} \xrightarrow{(3,4)}$$

$$\begin{bmatrix} \times & \times & \times \\ 0 & \times & \times \\ 0 & 0 & \times \\ 0 & 0 & 0 \\ 0 & 0 & 0 \end{bmatrix} := \begin{bmatrix} \boldsymbol{R}_1 \\ \boldsymbol{O} \end{bmatrix} := \boldsymbol{R},$$

此处已标记定义所对应 Givens 变换的 2 维向量. 显然, 若 $\boldsymbol{G}_i$ 表示在约化过程中的第 i 次 Givens 变换, 则 $\boldsymbol{Q}^{\mathrm{T}}\boldsymbol{A} = \boldsymbol{R}$ 是上三角形矩阵, 其中 $\boldsymbol{Q} = \boldsymbol{G}_1\boldsymbol{G}_2\cdots\boldsymbol{G}_t$, 且 t 是 Givens 变换的总次数. 对于一




般的 $\boldsymbol{A} \in \mathbb{R}^{m\times n}\,(m \geqslant n)$, 有如下算法.

算法 2.6 (Givens 变换 QR 分解)

$[m, n] = \mathbf{size}(\boldsymbol{A})$;

for $k = 1 : n$

  for $i = m : -1 : k+1$

    $[c, s] = \mathbf{r\_givens}(\boldsymbol{A}(i-1, k), \boldsymbol{A}(i, k))$;

    $$\boldsymbol{A}(i-1 : i, k : n) = \begin{bmatrix} c & s \\ -s & c \end{bmatrix} \boldsymbol{A}(i-1 : i, k : n);$$

  end

end

注 2.9 算法 2.6 是指对于给定的矩阵 $\boldsymbol{A} \in \mathbb{R}^{m\times n}\,(m \geqslant n)$, 计算 Givens 矩阵 $\boldsymbol{G}_1, \boldsymbol{G}_2, \cdots, \boldsymbol{G}_t\,(t = \frac{1}{2}n(2m-n-1))$ 满足: 如果

$\boldsymbol{Q} = \boldsymbol{G}_1\boldsymbol{G}_2\cdots\boldsymbol{G}_t$, 则 $\boldsymbol{Q}^{\mathrm{T}}\boldsymbol{A} = \boldsymbol{R}$ 是上三角形矩阵, 且 $\boldsymbol{A}$ 被 $\boldsymbol{R}$ 所覆盖. 容易计算出, 该算法的运算量为 $3n^2(m - n/3)$ 个 flop.

根据上述算法可编制 MATLAB 程序如下:

```
function [A]=givens_qr(A)
%Givens变换QR分解
[m,n]=size(A);
for k=1:n
    for i=m:-1:k+1
        [c,s]=r_givens(A(i-1,k), A(i,k));
        A(i-1:i, k:n)=[c, s; -s, c]*A(i-1:i, k:n);
    end
end
```

例 2.4 利用算法 2.6 对例 2.3 中的矩阵 $\boldsymbol{A}$ 进行 QR 分解.




解 由于算法 2.6 中没有显式计算和存储正交矩阵 $\boldsymbol{Q}$, 故不能直接调用程序 givens_qr.m, 需对其稍作修改:

```
function [Q,R]=Qgivens_qr(A)
%Givens变换QR分解, 显式计算并存储正交矩阵Q
[m,n]=size(A); Q=eye(m);
for k=1:n
    for i=m:-1:k+1
        [c,s]=r_givens(A(i-1,k),A(i,k));
        A(i-1:i,k:n)=[c,s;-s,c]*A(i-1:i,k:n);
        Q(:,i-1:i)=Q(:,i-1:i)*[c,s;-s,c]';
    end
end
R=triu(A);
```

然后再在命令窗口依次运行如下代码:

```
>> A=[76,96,85,35;26,55,26,20; ...
>>    51,14,82,26;70,15,25,62;90,26,93,48];
>> [Q,R]=Qgivens_qr(A)
```

即得矩阵 $\boldsymbol{A}$ 的 QR 分解, 且满足 $\|\boldsymbol{A}-\boldsymbol{QR}\|_2 = 7.6458 \times 10^{-14}$.

作为 Givens 变换 QR 分解的一个应用, 下面讨论上 Hessenberg 矩阵的 QR 分解问题. 从前述讨论可知, Givens 变换 QR 分解的运算量约为 Householder 变换的 1.5 倍. 尽管如此, 用 Givens 变换对上 Hessenberg 矩阵进行 QR 分解却具有独特的优势. 上 Hessenberg 矩阵是一类具有重要应用的矩阵类, 例如第 4 章即将介绍的广义极小残量法 (GMRES) 需要求解一个上 Hessenberg 矩阵的最小二乘问题. 下面介绍上 Hessenberg 矩阵的 Givens 变换 QR 分解的具体实现.

定义 2.4 如果 n 阶矩阵的下次对角线以下的元素都为零, 则称该矩阵为上 Hessenberg 矩阵. 上 Hessenberg 矩阵的转置称为下 Hessenberg 矩阵.

根据上 Hessenberg 矩阵的特殊结构 (矩阵的每一列对角线以下的元素中只有第 1 个元素可能非零), 因而只需对每一列作一次 Givens 变换即可, 具体来说有下面的算法.

算法 2.7 (上 Hessenberg 矩阵 QR 分解)

for $k = 1 : n - 1$

$$[c_k, s_k] = \mathbf{r\_givens}(\boldsymbol{A}(k,k), \boldsymbol{A}(k+1,k));$$

$$\boldsymbol{A}(k:k+1, k:n) = \begin{bmatrix} c_k & s_k \\ -s_k & c_k \end{bmatrix} \boldsymbol{A}(k:k+1, k:n);$$

end

注 2.10 算法 2.7 是指对于给定的上 Hessenberg 矩阵 $\boldsymbol{A} \in \mathbb{R}^{n\times n}$, 计算 Givens 矩阵 $\boldsymbol{G}_1, \boldsymbol{G}_2, \cdots, \boldsymbol{G}_{n-1}$ 满足: 如果

$$\boldsymbol{Q} = \boldsymbol{G}_1\boldsymbol{G}_2\cdots\boldsymbol{G}_{n-1},$$

则 $\boldsymbol{Q}^{\mathrm{T}}\boldsymbol{A} = \boldsymbol{R}$ 是上三角形矩阵, 且 $\boldsymbol{A}$ 被 $\boldsymbol{R}$ 所覆盖, 其中 $\boldsymbol{G}_k$ 形如 $\boldsymbol{G}_{k,k+1}(c_k, s_k)$. 容易计算出, 该算法的运算量只有 $3n^2$ 个 flop.

根据上述算法可编制 MATLAB 程序如下:

```
function [A]=hessenberg_qr(A)
%上Hessenberg矩阵的QR分解
[n]=size(A,2);
for k=1:n-1
    [c,s]=r_givens(A(k,k),A(k+1,k));
    A(k:k+1,k:n)=[c,s;-s,c]*A(k:k+1,k:n);
end
```

作为 Givens 变换 (或 Householder 变换) 的另一个应用, 下面介绍 n 阶实矩阵正交相似于上 Hessenberg 矩阵的计算问题.

定理 2.9 设 $\boldsymbol{A} \in \mathbb{R}^{n\times n}$, 则存在有限个 Givens 矩阵 (或 Householder 矩阵) 的乘积 $\boldsymbol{Q}$, 使得 $\boldsymbol{QAQ}^{\mathrm{T}}$ 为上 Hessenberg 矩阵.

证明 仅讨论使用 Givens 矩阵的情形.

第 1 步, 设 $\boldsymbol{A} = (a_{ij}) \in \mathbb{R}^{n\times n}$, 记 $\boldsymbol{\alpha}^{(0)} = (a_{21}, \cdots, a_{n1})^{\mathrm{T}} \in \mathbb{R}^{n-1}$, 当 $\boldsymbol{\alpha}^{(0)} = \boldsymbol{0}$ 时转入第 2 步; 当 $\boldsymbol{\alpha}^{(0)} \neq \boldsymbol{0}$ 时, 构造有限个 Givens 矩阵的乘积 $\boldsymbol{G}_0$, 使得

$$\boldsymbol{G}_0\boldsymbol{\alpha}^{(0)} = \|\boldsymbol{\alpha}^{(0)}\|_2\boldsymbol{e}_1, \quad (\boldsymbol{e}_1 \in \mathbb{R}^{n-1}).$$

记 $a_{21}^{(1)} = \|\boldsymbol{\alpha}^{(0)}\|_2$, 则有

$$\begin{bmatrix} 1 & \\ & \boldsymbol{G}_0 \end{bmatrix} \boldsymbol{A} \begin{bmatrix} 1 & \\ & \boldsymbol{G}_0 \end{bmatrix}^{\mathrm{T}} = \left[\begin{array}{c|ccc} a_{11} & a_{12}^{(1)} & a_{13}^{(1)} & \cdots & a_{1n}^{(1)} \\ \hline a_{21}^{(1)} & & & & \\ 0 & & & & \\ \vdots & & \boldsymbol{A}^{(1)} & & \\ 0 & & & & \end{array}\right].$$

第 2 步, 设 $\boldsymbol{A}^{(1)} \in \mathbb{R}^{(n-1)\times(n-1)}$, 记 $\boldsymbol{\alpha}^{(1)} = (a_{32}^{(1)}, \cdots, a_{n2}^{(1)})^{\mathrm{T}} \in \mathbb{R}^{n-2}$, 当 $\boldsymbol{\alpha}^{(1)} = \boldsymbol{0}$ 时转入第 3 步; 当 $\boldsymbol{\alpha}^{(1)} \neq \boldsymbol{0}$ 时, 构造有限个 Givens 矩阵的乘积 $\boldsymbol{G}_1$, 使得

$$\boldsymbol{G}_1\boldsymbol{\alpha}^{(1)} = \|\boldsymbol{\alpha}^{(1)}\|_2\boldsymbol{e}_1, \quad (\boldsymbol{e}_1 \in \mathbb{R}^{n-2}).$$




Back
Close

记 $a_{32}^{(2)} = \|\boldsymbol{\alpha}^{(1)}\|_2$, 则有

$$\begin{bmatrix} 1 & \\ & \boldsymbol{G}_1 \end{bmatrix} \boldsymbol{A}^{(1)} \begin{bmatrix} 1 & \\ & \boldsymbol{G}_1 \end{bmatrix}^{\mathrm{T}} = \left[\begin{array}{c|ccc} a_{22}^{(1)} & a_{23}^{(2)} & a_{24}^{(2)} & \cdots & a_{2n}^{(2)} \\ \hline a_{32}^{(2)} & & & & \\ 0 & & & & \\ \vdots & & \multicolumn{3}{c}{\boldsymbol{A}^{(2)}} \\ 0 & & & & \end{array}\right].$$

继续上述过程, 直到第 $n-2$ 步:

$$\boldsymbol{A}^{(n-3)} = \begin{bmatrix} a_{n-2,n-2}^{(n-3)} & a_{n-2,n-1}^{(n-3)} & a_{n-2,n}^{(n-3)} \\ a_{n-1,n-2}^{(n-3)} & a_{n-1,n-1}^{(n-3)} & a_{n-1,n}^{(n-3)} \\ a_{n,n-2}^{(n-3)} & a_{n,n-1}^{(n-3)} & a_{n,n}^{(n-3)} \end{bmatrix} \in \mathbb{R}^{3\times 3},$$

记 $\boldsymbol{\alpha}^{(n-3)} = \begin{bmatrix} a_{n-1,n-2}^{(n-3)} \\ a_{n,n-2}^{(n-3)} \end{bmatrix}$. 当 $\boldsymbol{\alpha}^{(n-3)} = \mathbf{0}$ 时结束; 当 $\boldsymbol{\alpha}^{(n-3)} \neq \mathbf{0}$ 时, 构造 Givens 矩阵 $\boldsymbol{G}_{n-3}$, 使得

$$\boldsymbol{G}_{n-3}\boldsymbol{\alpha}^{(n-3)} = \|\boldsymbol{\alpha}^{(n-3)}\|_2 \boldsymbol{e}_1, \quad (\boldsymbol{e}_1 \in \mathbb{R}^2).$$

记 $a_{n-1,n-2}^{(n-2)} = \|\boldsymbol{\alpha}^{(n-3)}\|_2$, 则有

$$\begin{bmatrix} 1 & \\ & \boldsymbol{G}_{n-3} \end{bmatrix} \boldsymbol{A}^{(n-3)} \begin{bmatrix} 1 & \\ & \boldsymbol{G}_{n-3} \end{bmatrix}^{\mathrm{T}} = \begin{bmatrix} a_{n-2,n-2}^{(n-3)} & a_{n-2,n-1}^{(n-3)} & a_{n-2,n}^{(n-3)} \\ a_{n-1,n-2}^{(n-2)} & a_{n-1,n-1}^{(n-2)} & a_{n-1,n}^{(n-2)} \\ 0 & a_{n,n-1}^{(n-2)} & a_{n,n}^{(n-2)} \end{bmatrix}.$$

最后, 构造正交矩阵

$$\boldsymbol{Q} = \begin{bmatrix} \boldsymbol{I}_{n-2} & \\ & \boldsymbol{G}_{n-3} \end{bmatrix} \cdots \begin{bmatrix} \boldsymbol{I}_2 & \\ & \boldsymbol{G}_1 \end{bmatrix} \begin{bmatrix} 1 & \\ & \boldsymbol{G}_0 \end{bmatrix},$$

可使 $\boldsymbol{QAQ}^{\mathrm{T}}$ 为上 Hessenberg 矩阵. 证毕. □

推论 2.1 设 $\boldsymbol{A}$ 是 n 阶实对称矩阵, 则存在有限个 Givens 矩阵 (或 Householder 矩阵) 的乘积 $\boldsymbol{Q}$, 使得 $\boldsymbol{QAQ}^{\mathrm{T}}$ 为实对称三对角矩阵.

证明 对 $\boldsymbol{A}$ 应用定理 2.9, 则存在正交矩阵 $\boldsymbol{Q}$, 使得 $\boldsymbol{QAQ}^{\mathrm{T}}$ 为上 Hessenberg 矩阵. 由于 $\boldsymbol{A}^{\mathrm{T}} = \boldsymbol{A}$, 又有 $\boldsymbol{QAQ}^{\mathrm{T}} = (\boldsymbol{QAQ}^{\mathrm{T}})^{\mathrm{T}}$ 为下 Hessenberg 矩阵, 故 $\boldsymbol{QAQ}^{\mathrm{T}}$ 是三对角矩阵, 从而 $\boldsymbol{QAQ}^{\mathrm{T}}$ 是实对称的三对角矩阵. 证毕. □

§2.3 线性无关向量组的正交化

本节讨论线性无关向量组的正交化. 对以这些向量为列的矩阵施以 Householder 变换或 Givens 变换作正交化, 就可以得到该线性无关向量组的正交化. 首先介绍经典的 Gram–Schmidt 正交化过程.

§2.3.1 Gram–Schmidt 正交化

考虑 m 个线性无关的 n 维向量组 $\{\boldsymbol{x}_i\}_{i=1}^m (m \leqslant n)$. 记列满秩 $n \times m$ 阶矩阵 $\boldsymbol{X} = [\boldsymbol{x}_1, \boldsymbol{x}_2, \cdots, \boldsymbol{x}_m]$. 对向量组 $\{\boldsymbol{x}_i\}_{i=1}^m$ 用 Gram–Schmidt 正交化产生相互正交的单位向量组 $\{\boldsymbol{q}_i\}_{i=1}^m$, 使得

$$\mathrm{span}\{\boldsymbol{x}_1, \boldsymbol{x}_2, \cdots, \boldsymbol{x}_k\} = \mathrm{span}\{\boldsymbol{q}_1, \boldsymbol{q}_2, \cdots, \boldsymbol{q}_k\}, \ k = 1, 2, \cdots, m. \tag{2.10}$$

按如下方式构造 $\{\boldsymbol{q}_i\}_{i=1}^m$. 令 $\boldsymbol{q}_1 = \boldsymbol{x}_1/\|\boldsymbol{x}_1\|_2$. 设已经构造好相互正交的单位向量组 $\{\boldsymbol{q}_i\}_{i=1}^{k-1}$ 使得式 (2.10) 对 $k-1$ 成立. 利用正交性, 得

$$\widetilde{\boldsymbol{q}}_k = \boldsymbol{x}_k - \sum_{i=1}^{k-1} r_{ik}\boldsymbol{q}_i,\ r_{is} = (\boldsymbol{x}_k, \boldsymbol{q}_i),\ i = 1, 2, \cdots, k-1;\ \boldsymbol{q}_k = \frac{\widetilde{\boldsymbol{q}}_k}{\|\widetilde{\boldsymbol{q}}_k\|_2}. \tag{2.11}$$

从而 $\boldsymbol{q}_k$ 与 $\boldsymbol{q}_1, \boldsymbol{q}_2, \cdots, \boldsymbol{q}_{k-1}$ 都正交且为单位向量. 在 $\boldsymbol{q}_1, \boldsymbol{q}_2, \cdots, \boldsymbol{q}_{k-1}$ 都已经单位正交化的前提下, 式 (2.11) 表明 $\boldsymbol{x}_k$ 去掉它在 $\boldsymbol{q}_i$ 上的分量 (投影) $r_{ik}\boldsymbol{q}_i\,(i = 1, 2, \cdots, k-1)$ 后剩下的部分与 $\boldsymbol{q}_1, \boldsymbol{q}_2, \cdots, \boldsymbol{q}_{k-1}$ 都正交. 据此, 可以写出 Gram–Schmidt 正交化的算法如下:

算法 2.8 (标准 Gram–Schmidt 正交化) 给定 $n \times m\,(m \leqslant n)$ 阶列满秩矩阵 $\boldsymbol{X} = [\boldsymbol{x}_1, \boldsymbol{x}_2, \cdots, \boldsymbol{x}_m]$, 本算法产生 $n \times m$ 阶矩阵 $\boldsymbol{Q} = [\boldsymbol{q}_1, \boldsymbol{q}_2, \cdots, \boldsymbol{q}_m]$ (其列是单位正交的) 和 $m \times m$ 阶非奇异

上三角矩阵 $\boldsymbol{R} = (r_{ik})$.

$r_{11} = \|\boldsymbol{x}_1\|_2;\ \boldsymbol{q}_1 = \boldsymbol{x}_1/r_{11};$

for $k = 2 : m$

for $(i = 1 : k-1),\ r_{ik} = (\boldsymbol{x}_k, \boldsymbol{q}_i);$ **end**

$\widetilde{\boldsymbol{q}} = \boldsymbol{x}_k - \sum_{i=1}^{k-1} r_{ik}\boldsymbol{q}_i;$

$r_{kk} = \|\widetilde{\boldsymbol{q}}\|_2;\ \boldsymbol{q}_k = \widetilde{\boldsymbol{q}}/r_{kk};$

end

由算法 2.8 可以得到关系式

$$\boldsymbol{x}_k = \sum_{i=1}^{k} r_{ik}\boldsymbol{q}_i, \quad k = 1, 2, \cdots, m. \tag{2.12}$$

写成矩阵形式即为

$$\boldsymbol{X} = \boldsymbol{Q}\boldsymbol{R}, \tag{2.13}$$

式中: $\boldsymbol{Q}=[\boldsymbol{q}_1,\boldsymbol{q}_2,\cdots,\boldsymbol{q}_m]\in\mathbb{R}^{n\times m}$, 其列是相互正交的单位向量; $\boldsymbol{R}$ 为 $m\times m$ 阶上三角矩阵, $\boldsymbol{x}_1,\boldsymbol{x}_2,\cdots,\boldsymbol{x}_m$ 的线性无关性保证了所有的 $r_{ii}(i=1,2,\cdots,m)$ 都不为零, 因而 $\boldsymbol{R}$ 是非奇异的.

算法 2.8 表明, 通过对矩阵 $\boldsymbol{X}$ 列向量的 Gram–Schmidt 正交化, 实现了矩阵 $\boldsymbol{X}$ 的 QR 分解.

标准 Gram–Schmidt 正交化过程的 MATLAB 程序如下:

```
%标准Gram-Schmidt正交化程序-G_Schmidt1.m
function [Q,R]=G_Schmidt1(X)
[m]=size(X,2); %矩阵的列
R=zeros(m); R(1,1)=norm(X(:,1));
Q(:,1)=X(:,1)/R(1,1); %单位化
for k=2:m
    for (i=1:k-1), R(i,k)=Q(:,i)'*X(:,k); end
```

```
        qt=X(:,k);
        for (i=1:k-1), qt=qt-R(i,k)*Q(:,i); end
        R(k,k)=norm(qt); Q(:,k)=qt/R(k,k); %单位化
    end
```

从算法 2.8 可以看出, 当 $\boldsymbol{x}_k$ 与前面的 $\boldsymbol{x}_i$ 接近线性相关时, r_{kk} 将接近于 0, 用它作分母会导致很大的舍入误差, $\boldsymbol{q}_i$ 之间很快失去了正交性. 一种改进就是修正的 Gram–Schmidt 正交化. 它把算法 2.8 的第 3 行和第 4 行修改为下面的循环:

$$\widetilde{\boldsymbol{q}} = \boldsymbol{x}_k, \quad r_{ik} = (\widetilde{\boldsymbol{q}}, \boldsymbol{q}_i), \quad \widetilde{\boldsymbol{q}} := \widetilde{\boldsymbol{q}} - r_{ik}\boldsymbol{q}_i, \quad i = 1, 2, \cdots, k-1. \tag{2.14}$$

在不考虑舍入误差时, 上述计算结果 $\widetilde{\boldsymbol{q}}$ 与算法 2.8 中第 4 行的 $\widetilde{\boldsymbol{q}}$ 是相同的. 实际上, 修正的 Gram–Schmidt 正交化利用了最新的 $\widetilde{\boldsymbol{q}}$, 有助于减少舍入误差影响.

为了保证正交性, 更可靠的方法是进行重正交化. 设已用式 (2.14) 计算了 $\widetilde{\boldsymbol{q}}$. 若 $\|\widetilde{\boldsymbol{q}}\|_2$ 很小 (与 $\|\boldsymbol{x}_k\|_2$ 相比), 用它作分母计算 $\boldsymbol{q}_k$ 会有很大的舍入误差. 此时, 对已计算的 $\widetilde{\boldsymbol{q}}$ 再正交化, 得

$$\widetilde{r}_{ik} = (\widetilde{\boldsymbol{q}}, \boldsymbol{q}_i), \quad \widetilde{\boldsymbol{q}} := \widetilde{\boldsymbol{q}} - \widetilde{r}_{ik}\boldsymbol{q}_i, \quad i = 1, 2, \cdots, k-1. \tag{2.15}$$

由于 $\widetilde{\boldsymbol{q}}$ 已与 $\boldsymbol{q}_i$ 接近正交, 故 $\widetilde{r}_{ik}$ 接近 0. 把它累加到 r_{ik} 上, 得

$$r_{ik} := r_{ik} + \widetilde{r}_{ik}, \quad i = 1, 2, \cdots, k-1. \tag{2.16}$$

可以写出如下的修正 Gram-Schmidt 正交化算法.

算法 2.9 (修正 Gram–Schmidt 正交化) 给定 $n \times m\,(m \leqslant n)$ 阶列满秩矩阵 $\boldsymbol{X} = [\boldsymbol{x}_1, \boldsymbol{x}_2, \cdots, \boldsymbol{x}_m]$, 本算法产生 $n \times m$ 阶矩阵 $\boldsymbol{Q} = [\boldsymbol{q}_1, \boldsymbol{q}_2, \cdots, \boldsymbol{q}_m]$ (其列是单位正交的) 和 $m \times m$ 阶非奇异上三角矩阵 $\boldsymbol{R} = (r_{ik})$.

$r_{11} = \|\boldsymbol{x}_1\|_2;\ \ \boldsymbol{q}_1 = \boldsymbol{x}_1/r_{11};$

for $\ k = 2 : m$

$\widetilde{\boldsymbol{q}} = \boldsymbol{x}_k;$

for $(i = 1 : k-1)$

$r_{ik} = (\widetilde{\boldsymbol{q}}, \boldsymbol{q}_i);\ \ \widetilde{\boldsymbol{q}} = \widetilde{\boldsymbol{q}} - r_{ik}\boldsymbol{q}_i;$

end

for $(i = 1 : k-1)$ %重正交化

$\widetilde{r}_{ik} = (\widetilde{\boldsymbol{q}}, \boldsymbol{q}_i);\ \ \widetilde{\boldsymbol{q}} = \widetilde{\boldsymbol{q}} - \widetilde{r}_{ik}\boldsymbol{q}_i;$

$r_{ik} = r_{ik} + \widetilde{r}_{ik};$

end

$r_{kk} = \|\widetilde{\boldsymbol{q}}\|_2;\ \ \boldsymbol{q}_k = \widetilde{\boldsymbol{q}}/r_{kk};$

end

再给出修正 Gram–Schmidt 正交化过程的 MATLAB 程序如下:

```
%修正Gram-Schmidt正交化程序-G_Schmidt2.m
function [Q,R]=G_Schmidt2(X)
[n,m]=size(X); %矩阵的列
R=zeros(m); Q=zeros(n,m);
R(1,1)=norm(X(:,1));
Q(:,1)=X(:,1)/R(1,1); %单位化
for k=2:m
    qt=X(:,k);
    for i=1:k-1
        R(i,k)=qt'*Q(:,i);
        qt=qt-R(i,k)*Q(:,i); %对剩余向量进行修正
    end
```

```
    for i=1:k-1  %重正交化
        rt=qt'*Q(:,i);
        qt=qt-rt*Q(:,i);
        R(i,k)=R(i,k)+rt;
    end
    R(k,k)=norm(qt);
    Q(:,k)=qt/R(k,k); %对本次得到的向量单位化
end
```

例 2.5 用 Gram–Schmidt 方法对矩阵 $\boldsymbol{A}$ 的列向量进行正交化, 其中

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

解 分别用标准 Gram–Schmidt 方法和修正 Gram–Schmidt 方法对矩阵 $\boldsymbol{A}$ 列向量进行正交化. 在命令窗口输入:

```
>> A=[9,1,4,5;6,2,1,3;4,1,9,9;5,2,9,4;4,2,5,1];
>> [Q1,R1]=G_Schmidt1(A);err1=norm(A-Q1*R1)
>> [Q2,R2]=G_Schmidt2(A);err2=norm(A-Q2*R2)
```

运行后即得结果, 且分别满足 $\|\boldsymbol{A} - \boldsymbol{Q}_1\boldsymbol{R}_1\|_2 = 1.0175 \times 10^{-15}$ 和 $\|\boldsymbol{A} - \boldsymbol{Q}_2\boldsymbol{R}_2\|_2 = 2.2204 \times 10^{-16}$.




§2.3.2 Householder 正交化

2.2 节中介绍了 Householder 变换可以对矩阵 $\boldsymbol{A} \in \mathbb{R}^{m\times n}\,(m \geqslant n)$ 进行 QR 分解. 这一过程也可以对任意 m 个线性无关的 n 维向量组进行. 与 Gram–Schmidt 正交化相比, 它具有更好的数值稳定性. 给定列满秩矩阵 $\boldsymbol{X} = [\boldsymbol{x}_1, \boldsymbol{x}_2, \cdots, \boldsymbol{x}_m]$, 这里假设 $m \leqslant n$. 令 $\boldsymbol{X}_1 = \boldsymbol{X}$, 构造 n 阶 Householder 矩阵 $\boldsymbol{H}_1$ 使得 $\boldsymbol{X}_2 = \boldsymbol{H}_1\boldsymbol{X}_1$ 的第 1 列的后 $n-1$ 个分量全部为 0. 一般地, 假设构造了 $k-1$ 个 Householder 矩阵 $\{\boldsymbol{H}_i\}_{i=1}^{k-1}$, 使得

$$\boldsymbol{X}_k = \boldsymbol{H}_{k-1}\boldsymbol{H}_{k-2}\cdots\boldsymbol{H}_1\boldsymbol{X}_1 = \begin{bmatrix} * & * & * & * & \cdots & * \\ & \ddots & * & * & \cdots & * \\ & & * & * & \cdots & * \\ & & & x_{kk} & \cdots & x_{km} \\ & & & \vdots & \ddots & \vdots \\ & & & x_{nk} & \cdots & x_{nm} \end{bmatrix}. \tag{2.17}$$

由于 $\boldsymbol{x}_1, \cdots, \boldsymbol{x}_m$ 线性无关, 故 $\boldsymbol{X}_k$ 的前 k 列线性无关. 故 $\widetilde{\boldsymbol{x}}_k = (x_{kk}, x_{k+1,k}, \cdots, x_{nk})^{\mathrm{T}}$ 不是零向量. 构造 $n-k+1$ 阶 Householder 矩阵 $\widetilde{\boldsymbol{H}}_k = \boldsymbol{I}_{n-k+1} - \beta_k \boldsymbol{v}_k \boldsymbol{v}_k^{\mathrm{T}}$, 使得 $\widetilde{\boldsymbol{H}}_k \widetilde{\boldsymbol{x}}_k$ 的后 $n-k$ 个分量为 0. 把 n 阶 Householder 矩阵 $\boldsymbol{H}_k = \mathrm{diag}(\boldsymbol{I}_{k-1}, \widetilde{\boldsymbol{H}}_k)$ 作用到 $\boldsymbol{X}_k$ 后得到

$$\boldsymbol{X}_{k+1} = \boldsymbol{H}_k \boldsymbol{X}_k = \boldsymbol{H}_k \boldsymbol{H}_{k-1} \cdots \boldsymbol{H}_1 \boldsymbol{X}_1 = \begin{bmatrix} * & * & * & * & * & \cdots & * \\ & \ddots & * & * & * & \cdots & * \\ & & * & * & * & \cdots & * \\ & & & \otimes & \otimes & \cdots & \otimes \\ & & & 0 & \otimes & \cdots & \otimes \\ & & & \vdots & \vdots & \ddots & \vdots \\ & & & 0 & \otimes & \cdots & \otimes \end{bmatrix}. \quad (2.18)$$

$\boldsymbol{X}_{k+1}$ 与 $\boldsymbol{X}_k$ 相比, 改变的仅仅是 $\boldsymbol{X}_k$ 右下角的 $n-k+1$ 阶子矩阵, 并产生如式 (2.18) 右端矩阵中位置的零元素. 上述过程进行 m 次之后, 就产生了 $n \times m$ 阶矩阵

$$\boldsymbol{X}_{m+1} = \boldsymbol{H}_m \boldsymbol{H}_{m-1} \cdots \boldsymbol{H}_1 \boldsymbol{X}_1 = \begin{bmatrix} \boldsymbol{R} \\ \boldsymbol{O} \end{bmatrix},$$

式中: $\boldsymbol{R}$ 为 m 阶上三角矩阵. 令

$$\boldsymbol{H} = \boldsymbol{H}_m \boldsymbol{H}_{m-1} \cdots \boldsymbol{H}_1, \ \boldsymbol{Q} = \boldsymbol{H}^{\mathrm{T}} \begin{bmatrix} \boldsymbol{I}_m \\ \boldsymbol{O} \end{bmatrix} = \left[\boldsymbol{q}_1, \boldsymbol{q}_2, \cdots, \boldsymbol{q}_m\right] \in \mathbb{R}^{n \times m},$$

则

$$\boldsymbol{X} = \boldsymbol{H}^{\mathrm{T}} \begin{bmatrix} \boldsymbol{R} \\ \boldsymbol{O} \end{bmatrix} = \boldsymbol{H}^{\mathrm{T}} \begin{bmatrix} \boldsymbol{I}_m \\ \boldsymbol{O} \end{bmatrix} \boldsymbol{R} = \boldsymbol{Q}\boldsymbol{R},$$

这就对一般矩阵实现了 QR 分解.

下面考虑这一分解算法的程序实现. 注意到

$$\boldsymbol{H}_k \boldsymbol{e}_i^{(n)} = \boldsymbol{e}_i^{(n)}, \quad k > i,$$

Back

Close

式中: $\boldsymbol{e}_i^{(n)}$ 为 n 维坐标向量, 故 $\boldsymbol{Q}=[\boldsymbol{q}_1,\boldsymbol{q}_2,\cdots,\boldsymbol{q}_m]$ 满足

$$\boldsymbol{q}_i=\boldsymbol{Q}\boldsymbol{e}_i^{(n)}=\boldsymbol{H}_1\boldsymbol{H}_2\cdots\boldsymbol{H}_m\boldsymbol{e}_i^{(n)}=\boldsymbol{H}_1\boldsymbol{H}_2\cdots\boldsymbol{H}_i\boldsymbol{e}_i^{(n)}.$$

为了描述整个算法, 再引入 $\boldsymbol{X}_{m+1}$ 的第 k 列为

$$\begin{aligned}\boldsymbol{r}_k&=\boldsymbol{H}_m\boldsymbol{H}_{m-1}\cdots\boldsymbol{H}_1\boldsymbol{X}_1\boldsymbol{e}_k=\boldsymbol{H}_m\boldsymbol{H}_{m-1}\cdots\boldsymbol{H}_1\boldsymbol{x}_k\\&=\boldsymbol{H}_k\boldsymbol{H}_{k-1}\cdots\boldsymbol{H}_1\boldsymbol{x}_k,\ 1\leqslant k\leqslant m,\end{aligned}\tag{2.19}$$

它就是 $\boldsymbol{X}_{k+1}$ 的第 k 列. 显然, 在计算出 $\boldsymbol{r}_k$ 之后, 在以后的各次正交化过程中, 它不再改变. 下面的算法给出了这一正交化过程的详细步骤.

算法 2.10 (Householder 正交化) 给定 $n\times m$ 阶列满秩矩阵 $\boldsymbol{X}=[\boldsymbol{x}_1,\boldsymbol{x}_2,\cdots,\boldsymbol{x}_m]$, 本算法产生 $n\times m$ 阶矩阵 $\boldsymbol{X}_{m+1}=[\boldsymbol{r}_1,\boldsymbol{r}_2,\cdots,\boldsymbol{r}_m]$ 和 $\boldsymbol{Q}=[\boldsymbol{q}_1,\boldsymbol{q}_2,\cdots,\boldsymbol{q}_m]$ (其列是单位正交的, 如果

不需要 $\boldsymbol{Q}$ 则不需计算). $\boldsymbol{X}_{m+1}$ 的前 m 行构成 m 阶上三角矩阵 $\boldsymbol{R}$.

① $\boldsymbol{r}_1 = \boldsymbol{x}_1$;

for $k = 1 : m$

② **if** $(k > 1)$, 计算 $\boldsymbol{r}_k = \boldsymbol{H}_{k-1}\boldsymbol{H}_{k-2}\cdots\boldsymbol{H}_1\boldsymbol{x}_k$; **end**

③ 根据 $\boldsymbol{r}_k$ 的后 $n-k+1$ 个分量构成的向量 $\widetilde{\boldsymbol{x}}$, 计算 $\widetilde{\boldsymbol{H}}_k$ 满足

$$\widetilde{\boldsymbol{H}}_k\widetilde{\boldsymbol{x}} = \eta_k\boldsymbol{e}_1^{(n-k+1)};\ \text{令}\ \boldsymbol{H}_k = \mathrm{diag}(\boldsymbol{I}_{k-1}, \widetilde{\boldsymbol{H}}_k);$$

④ 计算 $\boldsymbol{r}_k = \boldsymbol{H}_k\boldsymbol{r}_k$;

⑤ 计算 $\boldsymbol{q}_k = \boldsymbol{H}_1\boldsymbol{H}_2\cdots\boldsymbol{H}_k\boldsymbol{e}_k$;

end

注 2.11 算法 2.10 的第 ② 步可采取如下方式实现:

$$
\begin{aligned}
&\boldsymbol{z} = \boldsymbol{x}_k;\\
&\textbf{for } (i = 1 : k-1),\ \boldsymbol{z} = \boldsymbol{H}_i\boldsymbol{z};\ \textbf{end}\\
&\boldsymbol{r}_k = \boldsymbol{z};
\end{aligned}
\tag{2.20}
$$

类似地, 第 ⑤ 步可采取如下方式实现:

$$
\begin{aligned}
&\boldsymbol{z} = \boldsymbol{e}_k;\\
&\textbf{for } (i = k : -1 : 1),\ \boldsymbol{z} = \boldsymbol{H}_i\boldsymbol{z};\ \textbf{end}\\
&\boldsymbol{q}_k = \boldsymbol{z};
\end{aligned}
$$

现在来看一下算法 2.10 的程序执行. 当 $k = 1$ 时, 根据 $\boldsymbol{r}_1 = \boldsymbol{x}_1$ 的后 n 个分量构成 $\widetilde{\boldsymbol{x}}_1$ (就是 n 维列向量 $\boldsymbol{x}_1$), 构造 $\widetilde{\boldsymbol{H}}_1 = \boldsymbol{H}_1$, 计算 $\boldsymbol{X}_{m+1}$ 的第 1 列 $\boldsymbol{r}_1 := \boldsymbol{H}_1\boldsymbol{x}_1$, $k = 1$ 迭代结束. 当 $k = 2$ 时, 计




算 $\boldsymbol{r}_2 := \boldsymbol{H}_1\boldsymbol{x}_2$, 根据 $\boldsymbol{r}_2$ 的后 $n-1$ 个分量构成 $\widetilde{\boldsymbol{x}}_2$, 构造 $\widetilde{\boldsymbol{H}}_2$ 和 $\boldsymbol{H}_2 = \mathrm{diag}(\boldsymbol{I}_1, \widetilde{\boldsymbol{x}}_2)$. 计算 $\boldsymbol{X}_{m+1}$ 的第 2 列 $\boldsymbol{r}_2 := \boldsymbol{H}_2\boldsymbol{r}_2 = \boldsymbol{H}_2\boldsymbol{H}_1\boldsymbol{x}_1$. 至此, 确实 $\boldsymbol{r}_2$ 已经计算了. 注意算法 2.10 的第 ② 步计算的 $\boldsymbol{r}_k$ 是 $\boldsymbol{X}_k$ 的第 k 列, 第 ④ 步计算的 $\boldsymbol{r}_k$ 才是 $\boldsymbol{X}_{k+1}$ 的第 k 列, 亦即 $\boldsymbol{X}_{m+1}$ 的第 k 列.

此外, 该算法要保存 $\widetilde{\boldsymbol{H}}_k = \boldsymbol{I}_{n-k+1} - \beta_k\boldsymbol{v}_k\boldsymbol{v}_k^{\mathrm{T}}$ 中的数 β_k 和 $n-k+1$ 维列向量 $\boldsymbol{v}_k$. 在式 (2.20) $\boldsymbol{H}_i\boldsymbol{z}$ 的计算中, 由于 $\boldsymbol{H}_i = \mathrm{diag}(\boldsymbol{I}_{i-1}, \widetilde{\boldsymbol{H}}_i)$, 所以 $\boldsymbol{H}_i\boldsymbol{z}$ 的前 $i-1$ 个分量与 $\boldsymbol{z}$ 的前 $s-1$ 个分量相同, 而 $\boldsymbol{H}_i\boldsymbol{z}$ 的后 $n-i+1$ 个分量构成的列向量就是 $\widetilde{\boldsymbol{H}}_k$ 与 $\boldsymbol{z}$ 后 $n-i+1$ 个分量构成的列向量的乘积.

算法 2.10 的 MATLAB 程序如下:

```
%Householder正交化程序-House_orth.m
function [Q,R]=House_orth(X)
[n,m]=size(X); %矩阵的行和列
R(:,1)=X(:,1); E=eye(n); Y=cell(m,1);
for k=1:m
    if k>1
        z=X(:,k);
        for i=1:k-1
            v=Y{i};  %取出来
            z=blkdiag(eye(i-1),eye(n-i+1)-b(i)*v*v')*z;
        end
        R(:,k)=z;
```

```
    end
    [v,beta]=r_house(R(k:n,k));
    H=blkdiag(eye(k-1),eye(n-k+1)-beta*v*v');
    R(:,k)=H*R(:,k);
    b(k)=beta;  Y{k}=v;  %存起来
    z=E(:,k);
    for i=k:-1:1
     v=Y{i};  %取出来
     z=blkdiag(eye(i-1),eye(n-i+1)-b(i)*v*v')*z;
    end
    Q(:,k)=z;
end
R=R(1:m,:);
```

例 2.6 用 Householder 正交化方法对例 2.5 中矩阵 $\boldsymbol{A}$ 的列向量进行正交化.

解 利用程序 House_orth.m, 在命令窗口输入:

```
A=[9,1,4,5;6,2,1,3;4,1,9,9;5,2,9,4;4,2,5,1];
[Q,R]=House_orth(A); err=norm(A-Q*R)
```

运行后即得结果, 且满足 $\|\boldsymbol{A}-\boldsymbol{QR}\|_2 = 4.3565 \times 10^{-15}$.




§2.4 Krylov 子空间及其正交化

本节阐述与 Krylov 子空间有关的一些基本概念和重要性质, 并且给出计算其正交基的几个实用的算法.

§2.4.1 Krylov 子空间

首先来介绍一下 Krylov 子空间的有关概念和性质. 考虑这样一个问题: 假如并不明确地知道矩阵 $\boldsymbol{A}$, 而只知道对任意给定向量 $\boldsymbol{x}$ 可以产生向量 $\boldsymbol{y} = \boldsymbol{A}\boldsymbol{x}$ 的机制, 那么可以从中得到 $\boldsymbol{A}$ 的一些什么信息呢?

当然, 在这种情况下, 若选定一个向量 $\boldsymbol{v}$, 便可得到一个序列

$$\begin{aligned} &\boldsymbol{v}_0 = \boldsymbol{v},\ \boldsymbol{v}_1 = \boldsymbol{A}\boldsymbol{v}_0 = \boldsymbol{A}\boldsymbol{v},\ \boldsymbol{v}_2 = \boldsymbol{A}\boldsymbol{v}_1 = \boldsymbol{A}^2\boldsymbol{v}, \\ &\cdots,\ \boldsymbol{v}_{k+1} = \boldsymbol{A}\boldsymbol{v}_k = \boldsymbol{A}^{k+1}\boldsymbol{v},\ \cdots. \end{aligned} \tag{2.21}$$

假定 $\boldsymbol{A}$ 的特征多项式为

$$p_A(\lambda) = \det(\lambda \boldsymbol{I} - \boldsymbol{A}) = \lambda^n + \alpha_{n-1}\lambda^{n-1} + \cdots + \alpha_1\lambda + \alpha_0,$$

则由 Cayley–Hamilton 定理可知

$$\boldsymbol{v}_n + \alpha_{n-1}\boldsymbol{v}_{n-1} + \cdots + \alpha_1\boldsymbol{v}_1 + \alpha_0\boldsymbol{v}_0 = \boldsymbol{0},$$

从而有

$$\boldsymbol{V}\boldsymbol{p} = -\boldsymbol{v}_n, \tag{2.22}$$

式中:

$$\boldsymbol{V} = [\boldsymbol{v}_0, \boldsymbol{v}_1, \cdots, \boldsymbol{v}_{n-1}], \quad \boldsymbol{p} = (\alpha_0, \alpha_1, \cdots, \alpha_{n-1})^{\mathrm{T}}.$$

若 $\boldsymbol{V}$ 非奇异, 则由方程组 (2.22) 可唯一地确定特征多项式的系数构成的向量 $\boldsymbol{p}$, 从而可求得特征多项式, 进而得到 $\boldsymbol{A}$ 的所有特征值. 这就是说, 在所给定的条件下, 能得到 $\boldsymbol{A}$ 所有的特征值的信息.

上述求 $\boldsymbol{A}$ 的特征多项式的方法, 是在 1931 年首先由 Krylov 提出的. 因此, 后人就将序列式 (2.21) 称为 Krylov 序列, 而将子空间

$$\mathcal{K}_k(\boldsymbol{A}, \boldsymbol{v}) = \text{span}\{\boldsymbol{v}, \boldsymbol{A}\boldsymbol{v}, \cdots, \boldsymbol{A}^{k-1}\boldsymbol{v}\} \tag{2.23}$$

称为 Krylov 子空间, 将矩阵

$$\boldsymbol{K}_k(\boldsymbol{A}, \boldsymbol{v}) = [\boldsymbol{v}, \boldsymbol{A}\boldsymbol{v}, \cdots, \boldsymbol{A}^{k-1}\boldsymbol{v}] \tag{2.24}$$

称为 Krylov 矩阵.

由 Krylov 子空间的定义, 容易验证其有如下的性质.

定理 2.10 假定 $\boldsymbol{A} \in \mathbb{R}^{n\times n}$ 和 $\boldsymbol{v} \in \mathbb{R}^n$ 已经给定, 其中 $\boldsymbol{v} \neq \boldsymbol{0}$, 则 Krylov 子空间有如下性质:

(1) Krylov 子空间序列满足

$$\mathcal{K}_k(\boldsymbol{A}, \boldsymbol{v}) \subset \mathcal{K}_{k+1}(\boldsymbol{A}, \boldsymbol{v}), \quad \boldsymbol{A}\mathcal{K}_k(\boldsymbol{A}, \boldsymbol{v}) \subset \mathcal{K}_{k+1}(\boldsymbol{A}, \boldsymbol{v}).$$




Back
Close

(2) 对任意的非零实数 α, 有

$$\mathcal{K}_k(\boldsymbol{A}, \boldsymbol{v}) = \mathcal{K}_k(\alpha\boldsymbol{A}, \boldsymbol{v}) = \mathcal{K}_k(\boldsymbol{A}, \alpha\boldsymbol{v}) \quad \text{(伸缩不变性)}.$$

(3) 对任意的实数 μ, 有

$$\mathcal{K}_k(\boldsymbol{A}, \boldsymbol{v}) = \mathcal{K}_k(\boldsymbol{A} - \mu\boldsymbol{I}, \boldsymbol{v}) \quad \text{(位移不变性)}.$$

(4) 对任意的非奇异矩阵 $\boldsymbol{W} \in \mathbb{R}^{n\times n}$, 有

$$\mathcal{K}_k(\boldsymbol{W}^{-1}\boldsymbol{A}\boldsymbol{W}, \boldsymbol{W}^{-1}\boldsymbol{v}) = \boldsymbol{W}^{-1}\mathcal{K}_k(\boldsymbol{A}, \boldsymbol{v}).$$

(5) 若记次数不超过 $k-1$ 的实系数多项式的全体为 $\mathcal{P}_{k-1}$, 则 $\mathcal{K}_k(\boldsymbol{A}, \boldsymbol{v})$ 有如下表示

$$\mathcal{K}_k(\boldsymbol{A}, \boldsymbol{v}) = \left\{p(\boldsymbol{A})\boldsymbol{v} : p \in \mathcal{P}_{k-1}\right\}.$$

注 2.12 定理 2.10 的第 2 条是说, 对矩阵 $\boldsymbol{A}$ 和向量 $\boldsymbol{v}$ 乘以任意非零实数, 并不改变其生成的 Krylov 子空间, 称 Krylov 子空




间的这一性质为伸缩不变性. 第 3 条是说, 对矩阵进行位移, 并不改变其生成的 Krylov 子空间, 称这一性质为位移不变性. 第 4 条性质给出了在矩阵的相似变换下相应的 Krylov 子空间之间的关系, 这对处理与 Krylov 子空间有关的理论时十分有用. 第 5 条给出了 Krylov 子空间一个重要性质, 即其每个元都可看作是某个矩阵多项式作用到初始向量 $\boldsymbol{v}$ 上而得到的.

假设 $(\lambda, \boldsymbol{v})$ 是 $\boldsymbol{A}$ 的特征对, 则有 $\boldsymbol{A}^i\boldsymbol{v} = \lambda^i\boldsymbol{v}$, 从而有

$$\mathcal{K}_k(\boldsymbol{A}, \boldsymbol{v}) = \mathcal{K}_1(\boldsymbol{A}, \boldsymbol{v}), \quad k = 1, 2, \cdots .$$

换句话说, 此时在 Krylov 序列中, 从第 1 项之后就没有任何新的向量产生, 即它终止于第一项. 一般来讲, Krylov 序列在第 ℓ 项终止, 是指 ℓ 为满足 $\mathcal{K}_{\ell+1}(\boldsymbol{A}, \boldsymbol{v}) = \mathcal{K}_\ell(\boldsymbol{A}, \boldsymbol{v})$ 的最小整数.




Back
Close

定理 2.11 若 Krylov 序列在第 ℓ 项终止, 则 $\mathcal{K}_\ell(\boldsymbol{A},\boldsymbol{v})$ 是 $\boldsymbol{A}$ 的一个 ℓ 维不变子空间. 反过来, 若 $\boldsymbol{v}$ 属于 $\boldsymbol{A}$ 的一个 m 维不变子空间, 则必存在一个 $\ell \leqslant m$, 使得对应的 Krylov 序列在第 ℓ 项终止.

证明 若 Krylov 序列在第 ℓ 项终止, 则对任意 $\boldsymbol{x} \in \mathcal{K}_\ell(\boldsymbol{A},\boldsymbol{v})$, 有

$$\boldsymbol{A}\boldsymbol{x} \in \mathcal{K}_{\ell+1}(\boldsymbol{A},\boldsymbol{v}) = \mathcal{K}_\ell(\boldsymbol{A},\boldsymbol{v}),$$

即 $\mathcal{K}_\ell(\boldsymbol{A},\boldsymbol{v})$ 是 $\boldsymbol{A}$ 的不变子空间. 此外, 由于 ℓ 是使得 $\mathcal{K}_{\ell+1}(\boldsymbol{A},\boldsymbol{v}) = \mathcal{K}_\ell(\boldsymbol{A},\boldsymbol{v})$ 的最小整数, 故必有 $\boldsymbol{v}, \boldsymbol{A}\boldsymbol{v}, \cdots, \boldsymbol{A}^{\ell-1}\boldsymbol{v}$ 这 ℓ 个向量是线性无关的, 从而就有 $\dim \mathcal{K}_\ell(\boldsymbol{A},\boldsymbol{v}) = \ell$.

反过来, 若 $\boldsymbol{v} \in \mathcal{X}_m$, 这里 $\mathcal{X}_m$ 是 $\boldsymbol{A}$ 的 m 维不变子空间, 则对任意的整数 k, 必有 $\boldsymbol{A}^k\boldsymbol{v} \in \mathcal{X}_m$. 故必存在一个 $\ell \leqslant m$, 使得 $\boldsymbol{v}, \boldsymbol{A}\boldsymbol{v}, \cdots, \boldsymbol{A}^{\ell-1}\boldsymbol{v}$ 线性无关, 但 $\boldsymbol{v}, \boldsymbol{A}\boldsymbol{v}, \cdots, \boldsymbol{A}^{\ell-1}\boldsymbol{v}, \boldsymbol{A}^{\ell}\boldsymbol{v}$ 线性相关,

从而有 $\mathcal{K}_{\ell+1}(\boldsymbol{A},\boldsymbol{v})=\mathcal{K}_{\ell}(\boldsymbol{A},\boldsymbol{v})$. 当然这样的 ℓ 必须是使得上述等式成立的最小整数. 因此, 对应的 Krylov 序列必在第 ℓ 项终止. 证毕. □

由定理 2.11 可知, 当 Krylov 序列在第 ℓ 项终止时, 则 $\mathcal{K}_{\ell}(\boldsymbol{A},\boldsymbol{v})$ 中就包含了 $\boldsymbol{A}$ 的一些特征向量. 这对完成诸多的计算任务是十分有利的, 但也有不利的一面. 例如, 如果 ℓ 很小, 那么 $\mathcal{K}_{\ell}(\boldsymbol{A},\boldsymbol{v})$ 中可能并不含有所需要的信息, 这就会给计算带来困难, 必须重新开始.

§2.4.2 Arnoldi 正交分解

在实际应用 Krylov 子空间时, 需要它的一些基向量来描述它. 当然, 定义它的向量组应该是它的一组天然的基. 然而这组基在实际计算时是没有什么用处的, 因为随着 k 的增加, Krylov 矩阵会变得越来越病态. 因此, 为了实际计算的需要, 必须寻找其他对扰动

并不敏感的基向量.

设 $\boldsymbol{K}_{k+1}(\boldsymbol{A},\boldsymbol{v})=[\boldsymbol{v},\boldsymbol{A}\boldsymbol{v},\cdots,\boldsymbol{A}^k\boldsymbol{v}]$ 是列满秩的, 并假定它的 QR 分解为

$$\boldsymbol{K}_{k+1}(\boldsymbol{A},\boldsymbol{v})=\boldsymbol{V}_{k+1}\boldsymbol{R}_{k+1}, \tag{2.25}$$

式中: $\boldsymbol{V}_{k+1}=[\boldsymbol{v}_1,\boldsymbol{v}_2,\cdots,\boldsymbol{v}_{k+1}]\in\mathbb{R}^{n\times(k+1)}$ 满足 $\boldsymbol{V}_{k+1}^{\mathrm{T}}\boldsymbol{V}_{k+1}=\boldsymbol{I}_{k+1}$, $\boldsymbol{R}_{k+1}$ 为非奇异的上三角矩阵.

通常称 $\boldsymbol{V}_{k+1}$ 为 $\boldsymbol{K}_{k+1}(\boldsymbol{A},\boldsymbol{v})$ 的 QR 分解的 Q 因子, 简称为 $\boldsymbol{K}_{k+1}(\boldsymbol{A},\boldsymbol{v})$ 的 Q 因子. 将 $\boldsymbol{V}_{k+1}$ 和 $\boldsymbol{R}_{k+1}$ 作如下分块:

$$\boldsymbol{V}_{k+1}=[\boldsymbol{V}_k,\boldsymbol{v}_{k+1}],\qquad \boldsymbol{R}_{k+1}=\begin{bmatrix}\boldsymbol{R}_k & \boldsymbol{r}_{k+1}\\ \boldsymbol{0} & \rho_{k+1,k+1}\end{bmatrix}.$$

再注意到 $\boldsymbol{K}_{k+1}(\boldsymbol{A},\boldsymbol{v})=[\boldsymbol{K}_k(\boldsymbol{A},\boldsymbol{v}),\boldsymbol{A}^k\boldsymbol{v}]$, 由式 (2.25), 得

$$\boldsymbol{K}_k(\boldsymbol{A},\boldsymbol{v})=\boldsymbol{V}_k\boldsymbol{R}_k, \tag{2.26}$$

即 $\boldsymbol{V}_k$ 就是 $\boldsymbol{K}_k(\boldsymbol{A},\boldsymbol{v})$ 的 QR 分解的 Q 因子. 从而有

$$\mathcal{K}_k(\boldsymbol{A},\boldsymbol{v}) = \mathcal{R}(\boldsymbol{V}_k) = \mathrm{span}\{\boldsymbol{v}_1, \boldsymbol{v}_2, \cdots, \boldsymbol{v}_k\},$$

即 $\boldsymbol{V}_k$ 的列向量就构成了 Krylov 子空间 $\mathcal{K}_k(\boldsymbol{A},\boldsymbol{v})$ 的一组标准正交基.

这样自然想到利用 $\boldsymbol{K}_k(\boldsymbol{A},\boldsymbol{v})$ 的 QR 分解来产生 $\mathcal{K}_k(\boldsymbol{A},\boldsymbol{v})$ 的一组正交基. 然而这一方法是不可取的, 因为 $\boldsymbol{K}_k(\boldsymbol{A},\boldsymbol{v})$ 是十分病态的, 在计算机上形成 $\boldsymbol{K}_k(\boldsymbol{A},\boldsymbol{v})$ 就会引起较大的误差. 因此, 需要寻求其他方法来计算 $\boldsymbol{V}_k$.

事实上, 由式 (2.25) 和式 (2.26) 可以导出如下结果.

定理 2.12 给定矩阵 $\boldsymbol{A} \in \mathbb{R}^{n\times n}$ 和向量 $\boldsymbol{v} \in \mathbb{R}^n$, 并假定 $\boldsymbol{K}_{k+1}(\boldsymbol{A},\boldsymbol{v})$ 是列满秩的, 而且其 QR 分解的 Q 因子为 $\boldsymbol{V}_{k+1}$, 则必




Back
Close

有一个 $(k+1)\times k$ 阶不可约上 Hessenberg 矩阵 $\widetilde{\boldsymbol{H}}_k$, 使得

$$\boldsymbol{A}\boldsymbol{V}_k = \boldsymbol{V}_{k+1}\widetilde{\boldsymbol{H}}_k, \tag{2.27}$$

式中: $\boldsymbol{V}_k$ 为由 $\boldsymbol{V}_{k+1}$ 的前 k 列构成的矩阵. 反过来, 若有

$$\boldsymbol{V}_{k+1} = [\boldsymbol{V}_k, \boldsymbol{v}_{k+1}] \in \mathbb{R}^{n\times(k+1)}, \quad \boldsymbol{V}_{k+1}^{\mathrm{T}}\boldsymbol{V}_{k+1} = \boldsymbol{I}_{k+1},$$

以及不可约的上 Hessenberg 矩阵 $\widetilde{\boldsymbol{H}}_k$ 使得式 (2.27) 成立, 则 $\boldsymbol{V}_{k+1}$ 必是 $\boldsymbol{K}_{k+1}(\boldsymbol{A}, \boldsymbol{v}_1)$ 的 QR 分解的 Q 因子, 这里 $\boldsymbol{v}_1 = \boldsymbol{V}_{k+1}\boldsymbol{e}_1$ 是 $\boldsymbol{V}_{k+1}$ 的第 1 列.

证明 设 $\boldsymbol{V}_{k+1} = [\boldsymbol{V}_k, \boldsymbol{v}_{k+1}]$ 是 $\boldsymbol{K}_{k+1}(\boldsymbol{A}, \boldsymbol{v})$ 的 QR 分解的 Q 因子, 则由式 (2.26) 得

$$[\boldsymbol{A}\boldsymbol{v}, \cdots, \boldsymbol{A}^k\boldsymbol{v}] = \boldsymbol{A}\boldsymbol{K}_k(\boldsymbol{A}, \boldsymbol{v}) = \boldsymbol{A}\boldsymbol{V}_k\boldsymbol{R}_k. \tag{2.28}$$

再由式 (2.25) 得

$$[\boldsymbol{v}, \boldsymbol{A}\boldsymbol{K}_k(\boldsymbol{A}, \boldsymbol{v})] = \boldsymbol{K}_{k+1}(\boldsymbol{A}, \boldsymbol{v}) = \boldsymbol{V}_{k+1}\boldsymbol{R}_{k+1}.$$

比较上式两边的后 k 列, 得

$$\boldsymbol{A}\boldsymbol{K}_k(\boldsymbol{A}, \boldsymbol{v}) = \boldsymbol{V}_{k+1}\widehat{\boldsymbol{H}}_k, \tag{2.29}$$

式中: $\widehat{\boldsymbol{H}}_k$ 为由 $\boldsymbol{R}_{k+1}$ 的后 k 列构成的 $(k+1) \times k$ 阶上 Hessenberg 矩阵.

注意: $\boldsymbol{R}_{k+1}$ 非奇异蕴含着 $\widehat{\boldsymbol{H}}_k$ 是不可约的. 将式 (2.28) 与式 (2.29) 相结合, 得

$$\boldsymbol{A}\boldsymbol{V}_k\boldsymbol{R}_k = \boldsymbol{V}_{k+1}\widehat{\boldsymbol{H}}_k \Longrightarrow \boldsymbol{A}\boldsymbol{V}_k = \boldsymbol{V}_{k+1}\widetilde{\boldsymbol{H}}_k,$$

式中: $\widetilde{\boldsymbol{H}}_k = \widehat{\boldsymbol{H}}_k\boldsymbol{R}_k^{-1}$.

由于 $\widehat{\boldsymbol{H}}_k$ 是不可约的上 Hessenberg 矩阵, 而 $\boldsymbol{R}_k$ 是非奇异的上三角矩阵, 因此 $\widetilde{\boldsymbol{H}}_k$ 必为不可约的上 Hessenberg 矩阵.

反过来, 若有式 (2.27) 成立, 比较式 (2.27) 两边的各列, 得

$$\boldsymbol{A}\boldsymbol{v}_j = \sum_{i=1}^{j+1} h_{ij}\boldsymbol{v}_i, \quad j = 1, 2, \cdots, k.$$

于是有

$$\begin{aligned}
\boldsymbol{v}_2 &= \frac{1}{h_{21}}(\boldsymbol{A}\boldsymbol{v}_1 - h_{11}\boldsymbol{v}_1) = \rho_{12}\boldsymbol{v}_1 + \rho_{22}\boldsymbol{A}\boldsymbol{v}_1, \quad \rho_{22} = \frac{1}{h_{21}} \neq 0; \\
\boldsymbol{v}_3 &= \frac{1}{h_{32}}(\boldsymbol{A}\boldsymbol{v}_2 - h_{12}\boldsymbol{v}_1 - h_{22}\boldsymbol{v}_2) \\
&= \frac{1}{h_{32}}(\boldsymbol{A} - h_{22}\boldsymbol{I})(\rho_{12}\boldsymbol{v}_1 + \rho_{22}\boldsymbol{A}\boldsymbol{v}_1) - \frac{h_{12}}{h_{32}}\boldsymbol{v}_1 \\
&= \rho_{13}\boldsymbol{v}_1 + \rho_{23}\boldsymbol{A}\boldsymbol{v}_1 + \rho_{33}\boldsymbol{A}^2\boldsymbol{v}_1, \quad \rho_{33} = \frac{\rho_{22}}{h_{32}} \neq 0.
\end{aligned}$$

如此下去, 得

$$\boldsymbol{v}_j = \rho_{1j}\boldsymbol{v}_1 + \rho_{2j}\boldsymbol{A}\boldsymbol{v}_1 + \cdots + \rho_{jj}\boldsymbol{A}^{j-1}\boldsymbol{v}_1, \ \rho_{jj} = \frac{\rho_{j-1,j-1}}{h_{j,j-1}} \neq 0, \ 2 \leqslant j \leqslant k+1. \tag{2.30}$$

令

$$\boldsymbol{R}^{-1}=\begin{bmatrix} 1 & \rho_{12} & \rho_{13} & \cdots & \rho_{1,k+1} \\ & \rho_{22} & \rho_{23} & \cdots & \rho_{2,k+1} \\ & & \rho_{33} & \cdots & \rho_{3,k+1} \\ & & & \ddots & \vdots \\ & & & & \rho_{k+1,k+1} \end{bmatrix},$$

则将式 (2.30) 写成矩阵形式, 并且再补充第 1 列 $\boldsymbol{v}_1$, 得

$$[\boldsymbol{v}_1, \boldsymbol{v}_2, \cdots, \boldsymbol{v}_{k+1}] = [\boldsymbol{v}_1, \boldsymbol{A}\boldsymbol{v}_1, \cdots, \boldsymbol{A}^k\boldsymbol{v}_1]\boldsymbol{R}^{-1},$$

即

$$\boldsymbol{K}_{k+1}(\boldsymbol{A}, \boldsymbol{v}_1) = \boldsymbol{V}_{k+1}\boldsymbol{R},$$

这表明 $\boldsymbol{V}_{k+1}$ 是 $\boldsymbol{K}_{k+1}(\boldsymbol{A}, \boldsymbol{v}_1)$ 的 QR 分解的 Q 因子. 证毕. □

基于这一结果, 称形如式 (2.27) 的公式为一个长度为 k 的 Arnoldi 分解; 若其中的 $\widetilde{\boldsymbol{H}}_k$ 是不可约的, 则称这一分解是不可约的, 否则称为可约的.

由定理 2.12 可知, 只需求得分解式 (2.27), 即 $\boldsymbol{K}_k(\boldsymbol{A},\boldsymbol{v})$ 的 QR 分解的 Q 因子, 就能得到 Krylov 子空间 $\mathcal{K}_k(\boldsymbol{A},\boldsymbol{v})$ 的一组标准正交基. 下面计算分解式 (2.27).

将 $\widetilde{\boldsymbol{H}}_k$ 分块为

$$\widetilde{\boldsymbol{H}}_k = \begin{bmatrix} \boldsymbol{H}_k \\ \beta_k \boldsymbol{e}_k^{\mathrm{T}} \end{bmatrix},$$

则 Arnoldi 分解式 (2.27) 可改写为

$$\boldsymbol{A}\boldsymbol{V}_k = \boldsymbol{V}_k\boldsymbol{H}_k + \beta_k \boldsymbol{v}_{k+1}\boldsymbol{e}_k^{\mathrm{T}}. \tag{2.31}$$

这种形式有时使用起来更方便. 在式 (2.31) 两边左乘 $\boldsymbol{V}_k^{\mathrm{T}}$, 并且注意到 $\boldsymbol{V}_k$ 与 $\boldsymbol{v}_{k+1}$ 的正交性, 即有 $\boldsymbol{H}_k = \boldsymbol{V}_k^{\mathrm{T}}\boldsymbol{A}\boldsymbol{V}_k$. 通常称矩阵 $\boldsymbol{H}_k$ 为 $\boldsymbol{A}$ 关于 $\boldsymbol{V}_k$ 的 Rayleigh 商. 事实上, 式 (2.31) 提供了一种由 $\boldsymbol{V}_k$

来计算 $\boldsymbol{v}_{k+1}$ 的方法. 比较式 (2.31) 两边矩阵的最后一列, 得

$$\boldsymbol{A}\boldsymbol{v}_k = \boldsymbol{V}_k\boldsymbol{h}_k + \beta_k\boldsymbol{v}_{k+1}, \tag{2.32}$$

式中: $\boldsymbol{h}_k$ 为 $\boldsymbol{H}_k$ 的最后一列.

由于 $\boldsymbol{V}_{k+1}$ 是列正交的, 故在式 (2.32) 两边左乘 $\boldsymbol{V}_k^{\mathrm{T}}$, 得

$$\boldsymbol{h}_k = \boldsymbol{V}_k^{\mathrm{T}}\boldsymbol{A}\boldsymbol{v}_k. \tag{2.33}$$

又由式 (2.32), 得

$$\beta_k\boldsymbol{v}_{k+1} = \boldsymbol{A}\boldsymbol{v}_k - \boldsymbol{V}_k\boldsymbol{h}_k. \tag{2.34}$$

于是, 便有

$$\beta_k = \|\boldsymbol{A}\boldsymbol{v}_k - \boldsymbol{V}_k\boldsymbol{h}_k\|_2, \quad \boldsymbol{v}_{k+1} = (\boldsymbol{A}\boldsymbol{v}_k - \boldsymbol{V}_k\boldsymbol{h}_k)/\beta_k. \tag{2.35}$$

这本质上是一个求向量 $\boldsymbol{A}\boldsymbol{v}_k$ 在 $\mathcal{R}(\boldsymbol{V}_k)^{\perp}$ 上的正交投影的 Gram–Schmidt 的正交化过程. 从对 Gram–Schmidt 正交化过程的研究可




知, 为了保证其数值稳定性, 应该使用修正的 Gram–Schmidt 正交化过程. 记

$$\boldsymbol{h}_k = (h_{1k}, h_{2k}, \cdots, h_{kk})^{\mathrm{T}}, \quad h_{k+1,k} = \beta_k,$$
$$\boldsymbol{w} = \boldsymbol{A}\boldsymbol{v}_k, \quad \boldsymbol{r} = \boldsymbol{A}\boldsymbol{v}_k - \boldsymbol{V}_k\boldsymbol{h}_k.$$

利用这些记号, 将式 (2.33) 和式 (2.34) 换一种表达方式即为

$$h_{ik} = \boldsymbol{v}_i^{\mathrm{T}}\boldsymbol{w}, \quad i = 1, 2, \cdots, k, \quad \boldsymbol{r} = \boldsymbol{w} - \sum_{i=1}^{k} h_{ik}\boldsymbol{v}_i.$$

上式是说, 经典的 Gram–Schmidt 正交化过程, 是将 $\boldsymbol{w}$ 在每个 $\boldsymbol{v}_i$ 上的投影 h_{ik} 都算好之后, 再从 $\boldsymbol{w}$ 中一并减去 $\boldsymbol{w}$ 在 $\mathcal{R}(\boldsymbol{V}_k)^{\perp}$ 上的正交投影. 而修正的 Gram–Schmidt 正交化过程是, 当算好 $\boldsymbol{w}$ 在 $\boldsymbol{v}_1$ 上的正交投影 $h_{1k} = \boldsymbol{v}_1^{\mathrm{T}}\boldsymbol{w}$ 后, 马上从 $\boldsymbol{w}$ 中减去 $h_{1k}\boldsymbol{v}_1$, 即计算 $\boldsymbol{w}_1 = \boldsymbol{w} - h_{1k}\boldsymbol{v}_1$. 然后, 再用 $h_{2k} = \boldsymbol{v}_2^{\mathrm{T}}\boldsymbol{w}_1$ 确定 h_{2k}, 而不是

用 $h_{2k} = \boldsymbol{v}_2^{\mathrm{T}}\boldsymbol{w}$ 确定. 当然, 从理论上来讲, 二者是等价的, 即 $h_{2k} = \boldsymbol{v}_2^{\mathrm{T}}\boldsymbol{w} = \boldsymbol{v}_2^{\mathrm{T}}\boldsymbol{w}_1$. 然后再从 $\boldsymbol{w}_1$ 中减去 $h_{2k}\boldsymbol{v}_2$, 即计算 $\boldsymbol{w}_2 = \boldsymbol{w}_1 - h_{2k}\boldsymbol{v}_2$, 依此类推. 数值实验证明, 修正的 Gram–Schmidt 正交化过程要比经典的 Gram–Schmidt 过程数值性态好得多.

综合上面的讨论, 可得如下算法.

算法 2.11 (Arnoldi 正交分解) 给定矩阵 $\boldsymbol{A} \in \mathbb{R}^{n\times n}$, 向量 $\boldsymbol{v}_1 \in \mathbb{R}^n\,(\|\boldsymbol{v}_1\|_2 = 1)$ 和正整数 k, 本算法计算一个长度为 k 的 Arnoldi 分解: $\boldsymbol{A}\boldsymbol{V}_k = \boldsymbol{V}_k\boldsymbol{H}_k + h_{k+1,k}\boldsymbol{v}_{k+1}\boldsymbol{e}_k^{\mathrm{T}}$.

for $j = 1 : k$

$\boldsymbol{w} = \boldsymbol{A}\boldsymbol{v}_j$;

for $i = 1 : j$

$h_{ij} = \boldsymbol{v}_i^{\mathrm{T}}\boldsymbol{w}$; $\boldsymbol{w} = \boldsymbol{w} - h_{ij}\boldsymbol{v}_i$;

end

Back

Close

$h_{j+1,j} = \|\boldsymbol{w}\|_2$;

if $h_{j+1,j} = 0$

stop

else

$\boldsymbol{v}_{j+1} = \boldsymbol{w}/h_{j+1,j}$;

end

end

注 2.13 算法 2.11 只在计算 $\boldsymbol{w}$ 时用到 $\boldsymbol{A}$ 与一个向量作乘积, 这使得该算法可充分利用 $\boldsymbol{A}$ 的稀疏性和其所具有的特殊结构. 此外, 若 Arnoldi 过程中途中断, 即计算过程中出现了 $h_{j+1,j} = 0$, 则这表明 $\mathcal{K}_j(\boldsymbol{A}, \boldsymbol{v}_1)$ 已经是 $\boldsymbol{A}$ 的不变子空间. 在很多情况下, 这是有利的.

这里需要指出的是, 数值实验表明, 算法 2.11 在实际使用时, $\boldsymbol{v}_j$ 之间的正交性很快就会损失掉. 解决这一问题的一种方法就是在计算过程中使用重正交化技术 (即在算法中再重复执行一次 Gram–Schmidt 正交化过程). 这就得到如下的算法.

算法 2.12 (重正交化 Arnoldi 分解) 给定矩阵 $\boldsymbol{A} \in \mathbb{R}^{n\times n}$, 向量 $\boldsymbol{v}_1 \in \mathbb{R}^n$ 满足 $\|\boldsymbol{v}_1\|_2 = 1$ 以及正整数 k, 本算法计算一个长度为 k 的 Arnoldi 分解: $\boldsymbol{A}\boldsymbol{V}_k = \boldsymbol{V}_k\boldsymbol{H}_k + h_{k+1,k}\boldsymbol{v}_{k+1}\boldsymbol{e}_k^{\mathrm{T}}$, 并且在计算过程中使用重正交化技术.

for $j = 1 : k$

$\boldsymbol{w} = \boldsymbol{A}\boldsymbol{v}_j$;

for $i = 1 : j$

$h_{ij} = \boldsymbol{v}_i^{\mathrm{T}}\boldsymbol{w}$; $\boldsymbol{w} = \boldsymbol{w} - h_{ij}\boldsymbol{v}_i$;

end

for $i = 1 : j$ （重正交化）

$s = \boldsymbol{v}_i^{\mathrm{T}} \boldsymbol{w}; \quad h_{ij} = h_{ij} + s; \quad \boldsymbol{w} = \boldsymbol{w} - s\boldsymbol{v}_i;$

end

$h_{j+1,j} = \|\boldsymbol{w}\|_2;$

if $h_{j+1,j} = 0$

stop

else

$\boldsymbol{v}_{j+1} = \boldsymbol{w}/h_{j+1,j};$

end

end

§2.4.3 Lanczos 正交分解

当 $\boldsymbol{A} \in \mathbb{R}^{n\times n}$ 是对称矩阵时, 在 Arnoldi 分解中, 其关于 $\boldsymbol{V}_k$ 的 Rayleigh 商 $\boldsymbol{H}_k = \boldsymbol{V}_k^{\mathrm{T}}\boldsymbol{A}\boldsymbol{V}_k$ 就是一个对称三对角矩阵. 这样对应的 Arnoldi 分解就变成

$$\boldsymbol{A}\boldsymbol{V}_k = \boldsymbol{V}_k\boldsymbol{T}_k + \beta_k\boldsymbol{v}_{k+1}\boldsymbol{e}_k^{\mathrm{T}}, \tag{2.36}$$

式中:

$$\boldsymbol{T}_k = \begin{bmatrix} \alpha_1 & \beta_1 & & \\ \beta_1 & \alpha_2 & \ddots & \\ & \ddots & \ddots & \beta_{k-1} \\ & & \beta_{k-1} & \alpha_k \end{bmatrix}.$$

此时, 称式 (2.36) 为一个长度为 k 的 Lanczos 分解.

比较式 (2.36) 两边的各列, 得




Back
Close

$$\boldsymbol{A}\boldsymbol{v}_1 = \alpha_1\boldsymbol{v}_1 + \beta_1\boldsymbol{v}_2,$$

$$\boldsymbol{A}\boldsymbol{v}_j = \beta_{j-1}\boldsymbol{v}_{j-1} + \alpha_j\boldsymbol{v}_j + \beta_j\boldsymbol{v}_{j+1}, \quad j = 2, \cdots, k.$$

于是有

$$\alpha_1 = \boldsymbol{v}_1^{\mathrm{T}}\boldsymbol{A}\boldsymbol{v}_1,$$

$$\beta_1 = \|\boldsymbol{A}\boldsymbol{v}_1 - \alpha_1\boldsymbol{v}_1\|_2, \quad \boldsymbol{v}_2 = (\boldsymbol{A}\boldsymbol{v}_1 - \alpha_1\boldsymbol{v}_1)/\beta_1,$$

$$\alpha_j = \boldsymbol{v}_j^{\mathrm{T}}\boldsymbol{A}\boldsymbol{v}_j, \quad \boldsymbol{r}_j = \boldsymbol{A}\boldsymbol{v}_j - \alpha_j\boldsymbol{v}_j - \beta_{j-1}\boldsymbol{v}_{j-1},$$

$$\beta_j = \|\boldsymbol{r}_j\|_2, \quad \boldsymbol{v}_{j+1} = \boldsymbol{r}_j/\beta_j, \quad j = 2, \cdots, k.$$

这样, 就得到了如下的算法.

算法 2.13 (Lanczos 方法) 给定矩阵 $\boldsymbol{A} \in \mathbb{R}^{n\times n}$ 满足 $\boldsymbol{A}^{\mathrm{T}} = \boldsymbol{A}$, 向量 $\boldsymbol{v}_1 \in \mathbb{R}^n$ 满足 $\|\boldsymbol{v}_1\|_2 = 1$ 以及正整数 k, 本算法计算一个长度为 k 的 Lanczos 分解: $\boldsymbol{A}\boldsymbol{V}_k = \boldsymbol{V}_k\boldsymbol{T}_k + \beta_k\boldsymbol{v}_{k+1}\boldsymbol{e}_k^{\mathrm{T}}$.

$\beta_0 = 0;\ \boldsymbol{v}_0 = \boldsymbol{0};$

for $j = 1 : k$

$\boldsymbol{w} = \boldsymbol{A}\boldsymbol{v}_j; \quad \alpha_j = \boldsymbol{v}_j^{\mathrm{T}}\boldsymbol{w};$

$\boldsymbol{w} = \boldsymbol{w} - \alpha_j\boldsymbol{v}_j - \beta_{j-1}\boldsymbol{v}_{j-1};$

$\beta_j = \|\boldsymbol{w}\|_2;$

if $\beta_j = 0$

stop

else

$\boldsymbol{v}_{j+1} = \boldsymbol{w}/\beta_j;$

end

end

当然在实际应用时, 该算法所产生的 $\boldsymbol{v}_j$ 也将很快的失去它们之间的正交性. 弥补这一损失的方法仍然是重正交化方法. 其实重正交化就是在算法中的 $\boldsymbol{w} = \boldsymbol{w} - \alpha_j\boldsymbol{v}_j - \beta_{j-1}\boldsymbol{v}_{j-1}$ 之后, 再加入一

句

$$\boldsymbol{w} = \boldsymbol{w} - \sum_{i=1}^{j-1} (\boldsymbol{v}_i^{\mathrm{T}} \boldsymbol{w}) \boldsymbol{v}_i.$$

具体算法如下.

算法 2.14 (重正交化的 Lanczos 方法) 给定矩阵 $\boldsymbol{A} \in \mathbb{R}^{n \times n}$ 满足 $\boldsymbol{A}^{\mathrm{T}} = \boldsymbol{A}$, 向量 $\boldsymbol{v}_1 \in \mathbb{R}^n$ 满足 $\|\boldsymbol{v}_1\|_2 = 1$ 以及正整数 k, 本算法计算一个长度为 k 的 Lanczos 分解: $\boldsymbol{A}\boldsymbol{V}_k = \boldsymbol{V}_k \boldsymbol{T}_k + \beta_k \boldsymbol{v}_{k+1} \boldsymbol{e}_k^{\mathrm{T}}$.

$\beta_0 = 0$; $\boldsymbol{v}_0 = \boldsymbol{0}$;

for $j = 1 : k$

$\boldsymbol{w} = \boldsymbol{A}\boldsymbol{v}_j$; $\alpha_j = \boldsymbol{v}_j^{\mathrm{T}} \boldsymbol{w}$;

$\boldsymbol{w} = \boldsymbol{w} - \alpha_j \boldsymbol{v}_j - \beta_{j-1} \boldsymbol{v}_{j-1}$;

$\boldsymbol{w} = \boldsymbol{w} - \sum_{i=1}^{j-1} (\boldsymbol{v}_i^{\mathrm{T}} \boldsymbol{w}) \boldsymbol{v}_i$;

$\beta_j = \|\boldsymbol{w}\|_2;$

if $\beta_j = 0$

stop

else

$\boldsymbol{v}_{j+1} = \boldsymbol{w}/\beta_j;$

end

end

当然, 在具体计算时, 也可以用修正的 Gram–Schmidt 正交化方法来实现上式的计算. 有时为了使算法所产生的 $\boldsymbol{v}_j$ 之间有更好的正交性, 也可以连续进行两次重正交化, 即所谓的完全重正交化.

下面再给出一个具体的例子, 观察一下重正交化的 Lanczos 过程的效果.

例 2.7 考虑矩阵

$$\boldsymbol{A} = \begin{bmatrix} 1 & 1 & 1 & \cdots & 1 \\ 1 & 2 & 2 & \cdots & 2 \\ 1 & 2 & 3 & \cdots & 3 \\ \vdots & \vdots & \vdots & \ddots & \vdots \\ 1 & 2 & 3 & \cdots & n \end{bmatrix}. \tag{2.37}$$

随机地选取一个初始向量 $\boldsymbol{v}_1$, 并且取 $n = 1000$, $k = 50$. 若用算法 2.13, 则计算得到的矩阵 $\boldsymbol{V}_{50}$ 满足

$$\|\boldsymbol{V}_{50}^{\mathrm{T}}\boldsymbol{V}_{50} - \boldsymbol{I}_{50}\|_{\mathrm{F}} = 9.5938.$$

而若用重正交化的 Lanczos 算法, 则计算得到的矩阵 $\boldsymbol{V}_{50}$ 满足

$$\|\boldsymbol{V}_{50}^{\mathrm{T}}\boldsymbol{V}_{50} - \boldsymbol{I}_{50}\|_{\mathrm{F}} = 2.0073 \times 10^{-13}.$$

由此观察到, 重正交化的 Lanczos 过程的效果是明显的.

§2.5 投影方法

大多数已有的求解线性方程组迭代方法都按某种方式使用一种投影过程. 投影过程表达了从一个子空间推断出线性方程组近似解的一种规范方式. 本节在一般的框架下描述这种规范方式, 并给出一些理论. 进一步, 对一维情形的投影方法给出了较为详细的描述, 这给第 4 章中更为复杂的子空间投影方法提供了一个预览.

§2.5.1 投影算子及其性质

投影算子 $\boldsymbol{P}$ 是从 $\mathbb{C}^n$ 到其自身的一个幂等线性映射, 即满足

$$\boldsymbol{P}^2 = \boldsymbol{P}.$$

由此定义可得下述一些简单的性质. 首先, 如果 $\boldsymbol{P}$ 是一个投影, 则

$\boldsymbol{I}-\boldsymbol{P}$ 也是投影, 且下述关系成立,

$$\mathcal{N}(\boldsymbol{P})=\mathcal{R}(\boldsymbol{I}-\boldsymbol{P}).$$

另外, 两个子空间 $\mathcal{N}(\boldsymbol{P})$ 和 $\mathcal{R}(\boldsymbol{P})$ 的交只有元素 $\mathbf{0}$, 即 $\mathcal{N}(\boldsymbol{P})\cap\mathcal{R}(\boldsymbol{P})=\{\mathbf{0}\}$. 事实上, 如果向量 $\boldsymbol{x}$ 属于 $\mathcal{R}(\boldsymbol{P})$, 则由幂等性有 $\boldsymbol{P}\boldsymbol{x}=\boldsymbol{x}$. 如果它也属于 $\mathcal{N}(\boldsymbol{P})$, 则 $\boldsymbol{P}\boldsymbol{x}=\mathbf{0}$, 因此 $\boldsymbol{x}=\boldsymbol{P}\boldsymbol{x}=\mathbf{0}$. 此外, $\mathbb{C}^n$ 的每个元素可被写成 $\boldsymbol{x}=\boldsymbol{P}\boldsymbol{x}+(\boldsymbol{I}-\boldsymbol{P})\boldsymbol{x}$, 因此, 空间 $\mathbb{C}^n$ 可被分解成直和

$$\mathbb{C}^n=\mathcal{N}(\boldsymbol{P})\oplus\mathcal{R}(\boldsymbol{P}).$$

反之, 每个形成 $\mathbb{C}^n$ 的直和的子空间对 $\mathcal{K}$ 和 $\mathcal{S}$ 唯一定义一个投影算子, 使得 $\mathcal{R}(\boldsymbol{P})=\mathcal{K}$ 和 $\mathcal{N}(\boldsymbol{P})=\mathcal{S}$. 相应的投影算子 $\boldsymbol{P}$ 映 $\mathbb{C}^n$ 的一个元素 $\boldsymbol{x}$ 为分量 $\boldsymbol{x}_1$, 其中 $\boldsymbol{x}_1$ 是相应直和唯一分解 $\boldsymbol{x}=\boldsymbol{x}_1+\boldsymbol{x}_2$ 中的 $\mathcal{K}$ 分量.

事实上, 这种对应是唯一的, 即任意投影算子 $\boldsymbol{P}$ 可由给定的两个子空间完全确定: $\boldsymbol{P}$ 的值域 $\mathcal{K}$ 和它的核 $\mathcal{S}$ (亦即 $\boldsymbol{I}-\boldsymbol{P}$ 的值域). 若对任意的 $\boldsymbol{x}$, 向量 $\boldsymbol{P}\boldsymbol{x}$ 满足

$$\boldsymbol{P}\boldsymbol{x}\in\mathcal{K},\quad \boldsymbol{x}-\boldsymbol{P}\boldsymbol{x}\in\mathcal{S},$$

则称线性映射 $\boldsymbol{P}$ 沿 (或平行于) 子空间 $\mathcal{S}$ 将 $\boldsymbol{x}$ 投影到 $\mathcal{K}$ 上. 如果 $\boldsymbol{P}$ 的秩为 m, 则 $\boldsymbol{I}-\boldsymbol{P}$ 的值域是 $n-m$ 维的. 因此, 通过其 m 维的正交补 $\mathcal{L}=\mathcal{S}^{\perp}$ 来定义 $\mathcal{S}$ 是自然的. 对任意的 $\boldsymbol{x}$, 上述定义 $\boldsymbol{u}=\boldsymbol{P}\boldsymbol{x}$ 的条件变为

$$\boldsymbol{u}\in\mathcal{K}, \tag{2.38}$$

$$\boldsymbol{x}-\boldsymbol{u}\perp\mathcal{L}. \tag{2.39}$$

这定义了一个正交于子空间 $\mathcal{L}$ 的到 $\mathcal{K}$ 上的投影算子 $\boldsymbol{P}$. 式 (2.38) 建立了 m 个自由度, 式 (2.39) 给出了从这些自由度来定义 $\boldsymbol{P}\boldsymbol{x}$ 的




m 个约束.

现在的问题是: 任给两个都是 m 维的子空间 $\mathcal{K}$ 和 $\mathcal{L}$, 是否总能通过式 (2.38) 和式 (2.39) 定义一个正交于 $\mathcal{L}$ 的到 $\mathcal{K}$ 上的投影算子 $\boldsymbol{P}$? 下面的引理回答这个问题.

引理 2.1 *给定两个 m 维子空间 $\mathcal{K}$ 和 $\mathcal{L}$, 下面两个条件是等价的:*

(1) $\mathcal{K}$ 中没有非零向量正交于 $\mathcal{L}$.

(2) 对任意的 $\boldsymbol{x} \in \mathbb{C}^n$, 存在唯一向量 $\boldsymbol{u}$ 满足式 (2.38) 和 (2.39).

引理 2.1 说明给定两个满足条件 $\mathcal{K} \cap \mathcal{L}^{\perp} = \{\boldsymbol{0}\}$ 的子空间 $\mathcal{K}$ 和 $\mathcal{L}$, 存在一个正交于 $\mathcal{L}$ 的到 $\mathcal{K}$ 上的投影算子 $\boldsymbol{P}$, 它用式 (2.38) 和 (2.39) 定义任意向量 $\boldsymbol{x}$ 的投影向量 $\boldsymbol{u}$. 此投影算子满足

$$\mathcal{R}(\boldsymbol{P}) = \mathcal{K}, \quad \mathcal{N}(\boldsymbol{P}) = \mathcal{L}^{\perp}.$$




Back
Close

特别地, 条件 $\boldsymbol{Px}=\boldsymbol{0}$ 转化为 $\boldsymbol{x}\in\mathcal{N}(\boldsymbol{P})$, 而这意味着 $\boldsymbol{x}\in\mathcal{L}^{\perp}$. 反之亦真. 这可得下面的性质:

$$\boldsymbol{Px}=\boldsymbol{0}\iff\boldsymbol{x}\in\mathcal{L}^{\perp}.$$

为了得到一般投影算子的矩阵表示, 需要两组基: 子空间 $\mathcal{K}=\mathcal{R}(\boldsymbol{P})$ 的一组基 $\boldsymbol{V}=[\boldsymbol{v}_1,\cdots,\boldsymbol{v}_m]$ 和子空间 $\mathcal{L}$ 的一组基 $\boldsymbol{W}=[\boldsymbol{w}_1,\cdots,\boldsymbol{w}_m]$. 当

$$(\boldsymbol{v}_i,\boldsymbol{w}_j)=\delta_{ij}=\begin{cases}1,&\text{若 }i=j\\0,&\text{若 }i\neq j\end{cases},$$

时, 称这两组基是双正交的, 写成矩阵形式为 $\boldsymbol{W}^{\mathrm{H}}\boldsymbol{V}=\boldsymbol{I}$. 由于 $\boldsymbol{Px}$ 属于 $\mathcal{K}$, 故可表示为 $\boldsymbol{Px}=\boldsymbol{Vy}$. 约束 $\boldsymbol{x}-\boldsymbol{Px}\perp\mathcal{L}$ 等价于条件

$$(\boldsymbol{x}-\boldsymbol{Vy},\boldsymbol{w}_i)=0,\quad i=1,\cdots,m.$$

写成矩阵形式, 即

$$\boldsymbol{W}^{\mathrm{H}}(\boldsymbol{x}-\boldsymbol{V}\boldsymbol{y})=\boldsymbol{0}. \tag{2.40}$$

如果两组基是双正交的, 则得 $\boldsymbol{y}=\boldsymbol{W}^{\mathrm{H}}\boldsymbol{x}$. 因此, 此时 $\boldsymbol{P}\boldsymbol{x}=\boldsymbol{V}\boldsymbol{W}^{\mathrm{H}}\boldsymbol{x}$, 这导致 $\boldsymbol{P}$ 的矩阵表示

$$\boldsymbol{P}=\boldsymbol{V}\boldsymbol{W}^{\mathrm{H}}. \tag{2.41}$$

对两组基 $\boldsymbol{V}$ 和 $\boldsymbol{W}$ 不是双正交的情形, 由式 (2.40) 易见

$$\boldsymbol{P}=\boldsymbol{V}(\boldsymbol{W}^{\mathrm{H}}\boldsymbol{V})^{-1}\boldsymbol{W}^{\mathrm{H}}.$$

如果假设 $\mathcal{K}$ 中没有向量正交于 $\mathcal{L}$, 则可以证明 $m\times m$ 阶矩阵 $\boldsymbol{W}^{\mathrm{H}}\boldsymbol{V}$ 是非奇异的.

当子空间 $\mathcal{L}=\mathcal{K}$ 时, 称投影算子 $\boldsymbol{P}$ 为到 $\mathcal{K}$ 上的正交投影算子. 非正交投影算子称为斜投影算子. 对于正交投影算子, 有

$$\mathcal{N}(\boldsymbol{P})=\mathcal{R}(\boldsymbol{P})^{\perp}.$$

因此, 可由下面对任意向量 $\boldsymbol{x}$ 都要满足的要求来定义一个正交投影算子:

$$\boldsymbol{Px} \in \mathcal{K} \quad \text{且} \quad (\boldsymbol{I}-\boldsymbol{P})\boldsymbol{x} \perp \mathcal{K},$$

或等价地, 有

$$\boldsymbol{Px} \in \mathcal{K} \quad \text{且} \quad \big((\boldsymbol{I}-\boldsymbol{P})\boldsymbol{x}, \boldsymbol{y}\big) = 0, \quad \forall\, \boldsymbol{y} \in \mathcal{K}.$$

考虑映射 $\boldsymbol{P}$ 的伴随映射 $\boldsymbol{P}^{\mathrm{H}}$, 它定义为

$$(\boldsymbol{P}^{\mathrm{H}}\boldsymbol{x}, \boldsymbol{y}) = (\boldsymbol{x}, \boldsymbol{Py}), \quad \forall\, \boldsymbol{x}, \boldsymbol{y}. \tag{2.42}$$

首先注意到 $\boldsymbol{P}^{\mathrm{H}}$ 也是一个投影算子, 因为对所有的 $\boldsymbol{x}$ 和 $\boldsymbol{y}$,

$$((\boldsymbol{P}^{\mathrm{H}})^2\boldsymbol{x}, \boldsymbol{y}) = (\boldsymbol{P}^{\mathrm{H}}\boldsymbol{x}, \boldsymbol{Py}) = (\boldsymbol{x}, \boldsymbol{P}^2\boldsymbol{y}) = (\boldsymbol{x}, \boldsymbol{Py}) = (\boldsymbol{P}^{\mathrm{H}}\boldsymbol{x}, \boldsymbol{y}).$$

由式 (2.42), 得

$$\mathcal{N}(\boldsymbol{P}^{\mathrm{H}}) = \mathcal{R}(\boldsymbol{P})^{\perp}, \tag{2.43}$$




$$\mathcal{N}(\boldsymbol{P}) = \mathcal{R}(\boldsymbol{P}^{\mathrm{H}})^{\perp}. \tag{2.44}$$

上述关系导致下面的性质.

性质 2.1 投影算子是正交的当且仅当它是 Hermite 的.

任给一个矩阵 $\boldsymbol{V} \in \mathbb{C}^{n\times m}$, 其列满足 $\boldsymbol{v}_i^{\mathrm{H}}\boldsymbol{v}_i = 1\,(i = 1, 2, \cdots, m)$. 则其列形成 $\mathcal{K} = \mathcal{R}(\boldsymbol{P})$ 的一组标准正交基, 可用矩阵 $\boldsymbol{P} = \boldsymbol{V}\boldsymbol{V}^{\mathrm{H}}$ 表示 $\boldsymbol{P}$. 这是投影算子矩阵表示 (2.41) 的一种特殊情况. 除了是幂等的, 与此矩阵相应的线性映射满足上述给出的特征, 即

$$\boldsymbol{V}\boldsymbol{V}^{\mathrm{H}}\boldsymbol{x} \in \mathcal{K} \quad 且 \quad (\boldsymbol{I} - \boldsymbol{V}\boldsymbol{V}^{\mathrm{H}})\boldsymbol{x} \in \mathcal{K}^{\perp}.$$

重要的是注意到正交投影算子 $\boldsymbol{P}$ 的这种表示不是唯一的. 事实上, 任意标准正交基 $\boldsymbol{V}$ 将给出 $\boldsymbol{P}$ 的一种不同的形式的表示. 因此, 对 $\mathcal{K}$ 的任意两组正交基 $\boldsymbol{V}_1$ 和 $\boldsymbol{V}_2$, 必须有 $\boldsymbol{V}_1\boldsymbol{V}_1^{\mathrm{H}} = \boldsymbol{V}_2\boldsymbol{V}_2^{\mathrm{H}}$, 这一等式可独立地证明.

下面来看看正交投影的性质. 当 $\boldsymbol{P}$ 是一个正交投影算子时, 分解 $\boldsymbol{x} = \boldsymbol{P}\boldsymbol{x} + (\boldsymbol{I} - \boldsymbol{P})\boldsymbol{x}$ 中的两个向量是正交的, 得到下述关系:

$$\|\boldsymbol{x}\|_2^2 = \|\boldsymbol{P}\boldsymbol{x}\|_2^2 + \|(\boldsymbol{I} - \boldsymbol{P})\boldsymbol{x}\|_2^2.$$

因此, 对任意 $\boldsymbol{x}$, 有

$$\|\boldsymbol{P}\boldsymbol{x}\|_2 \leqslant \|\boldsymbol{x}\|_2.$$

故对所有 $\boldsymbol{x} \in \mathbb{C}^n$, $\|\boldsymbol{P}\boldsymbol{x}\|_2/\|\boldsymbol{x}\|_2$ 的最大值不超过 1, 另外, 对 $\mathcal{R}(\boldsymbol{P})$ 中任意元素达到 1. 所以对任意正交投影算子 $\boldsymbol{P}$, 有

$$\|\boldsymbol{P}\|_2 = 1.$$

正交投影算子只有两个特征值: 0 和 1. $\boldsymbol{P}$ 的值域中的任意向量都是相应于特征值 1 的特征向量, 核中的任意向量都是相应于特征值 0 的特征向量.

下面建立关于正交投影算子的一个重要的最优性性质.

Back

Close

定理 2.13 设 $\boldsymbol{P}$ 是到子空间 $\mathcal{K}$ 上的正交投影算子, 则对任意给定的向量 $\boldsymbol{x} \in \mathbb{C}^n$, 下式成立:

$$\min_{\boldsymbol{y}\in\mathcal{K}} \|\boldsymbol{x}-\boldsymbol{y}\|_2 = \|\boldsymbol{x}-\boldsymbol{P}\boldsymbol{x}\|_2. \tag{2.45}$$

证明 设 $\boldsymbol{y}$ 为 $\mathcal{K}$ 中任意向量, 考虑它到 $\boldsymbol{x}$ 的距离的平方. 由于 $\boldsymbol{x}-\boldsymbol{P}\boldsymbol{x}$ 正交于 $\mathcal{K}$, 而 $\boldsymbol{P}\boldsymbol{x}-\boldsymbol{y}$ 属于 $\mathcal{K}$, 则

$$\|\boldsymbol{x}-\boldsymbol{y}\|_2^2 = \|\boldsymbol{x}-\boldsymbol{P}\boldsymbol{x}+(\boldsymbol{P}\boldsymbol{x}-\boldsymbol{y})\|_2^2 = \|\boldsymbol{x}-\boldsymbol{P}\boldsymbol{x}\|_2^2 + \|\boldsymbol{P}\boldsymbol{x}-\boldsymbol{y}\|_2^2.$$

因此, 对任意 $\boldsymbol{y} \in \mathcal{K}$, $\|\boldsymbol{x}-\boldsymbol{y}\|_2^2 \geqslant \|\boldsymbol{x}-\boldsymbol{P}\boldsymbol{x}\|_2^2$. 注意到对 $\boldsymbol{y} = \boldsymbol{P}\boldsymbol{x}$ 达到极小, 这就证明了式 (2.45). 证毕. □

通过对子空间 $\mathcal{K}$ 上的正交投影算子 $\boldsymbol{P}$ 定义 $\boldsymbol{y}^* \equiv \boldsymbol{P}\boldsymbol{x}$, 可将上述结果重新表示成一个充分必要条件的形式, 这个条件可以确定一个向量 $\boldsymbol{x}$ 在最小二乘意义下的最佳逼近.

推论 2.2 设给定一个子空间 $\mathcal{K}$ 和一个向量 $\boldsymbol{x} \in \mathbb{C}^n$, 则

$$\min_{\boldsymbol{y} \in \mathcal{K}} \|\boldsymbol{x} - \boldsymbol{y}\|_2 = \|\boldsymbol{x} - \boldsymbol{y}^*\|_2,$$

当且仅当满足下述条件:

$$\begin{cases} \boldsymbol{y}^* \in \mathcal{K}, \\ \boldsymbol{x} - \boldsymbol{y}^* \perp \mathcal{K}. \end{cases}$$

§2.5.2 投影方法的基本框架

考虑线性方程组 $\boldsymbol{A}\boldsymbol{x} = \boldsymbol{b}$, 其中 $\boldsymbol{A} \in \mathbb{R}^{n \times n}$. 这里, 符号 $\boldsymbol{A}$ 既用来表示矩阵, 也用来表示它所代表的 $\mathbb{R}^n$ 中的线性映射. 投影方法的思想是从 $\mathbb{R}^n$ 的一个子空间推断出问题的一个近似解. 如果 $\mathcal{K}$ 是这种候选近似子空间, 或称“搜索子空间”, 并且维数为 m, 则一般地为推断出这种近似必须加 m 个约束. 描述这些约束的一个典

型的方式是强加 m 个正交性条件. 特别地, 残差向量 $\boldsymbol{b}-\boldsymbol{A}\boldsymbol{x}$ 限制为与 m 个线性无关的向量正交. 这就定义了另一个 m 维子空间 $\mathcal{L}$, 称其为“约束子空间”. 这种简单的框架称为 Petrov–Galerkin 条件, 它在许多不同的数学方法中是很普遍的.

投影方法有两大类: 正交投影方法和斜投影方法. 在正交投影技术中, 子空间 $\mathcal{L}$ 与 $\mathcal{K}$ 是相同的. 在斜投影方法中 $\mathcal{L}$ 与 $\mathcal{K}$ 不相同, 甚至彼此完全无关. 这种差别非常重要并由此给出不同类型的算法.

设 $\boldsymbol{A}\in\mathbb{R}^{n\times n}$ 且 $\mathcal{K}$ 和 $\mathcal{L}$ 为 $\mathbb{R}^n$ 的两个 m 维子空间. 正交于 $\mathcal{L}$ 的到 $\mathcal{K}$ 上的投影技术是一个过程: 通过条件 $\widetilde{\boldsymbol{x}}\in\mathcal{K}$ 并且新的残差向量正交于 $\mathcal{L}$ 来求 $\boldsymbol{A}\boldsymbol{x}=\boldsymbol{b}$ 的一个近似解 $\widetilde{\boldsymbol{x}}$, 即

$$\text{求}\ \ \widetilde{\boldsymbol{x}}\in\mathcal{K},\ \ \text{使得}\ \ \boldsymbol{b}-\boldsymbol{A}\widetilde{\boldsymbol{x}}\perp\mathcal{L}.$$

如果希望利用解的初始值 $\boldsymbol{x}^{(0)}$ 的信息, 则必须在仿射空间

$\boldsymbol{x}^{(0)} + \mathcal{K}$ 中而不是在齐次的向量空间 $\mathcal{K}$ 中寻求近似解. 这需要对上述公式进行稍微的修改. 近似解问题需要重新定义为

$$\widetilde{\boldsymbol{x}} \in \boldsymbol{x}^{(0)} + \mathcal{K}, \quad \text{使得} \quad \boldsymbol{b} - \boldsymbol{A}\widetilde{\boldsymbol{x}} \perp \mathcal{L}.$$

注意到, 如果将 $\widetilde{\boldsymbol{x}}$ 写成形式 $\widetilde{\boldsymbol{x}} = \boldsymbol{x}^{(0)} + \boldsymbol{z}$, 且初始残差向量 $\boldsymbol{r}^{(0)}$ 定义为 $\boldsymbol{r}^{(0)} = \boldsymbol{b} - \boldsymbol{A}\boldsymbol{x}^{(0)}$, 则上述方程变为

$$\boldsymbol{b} - \boldsymbol{A}(\boldsymbol{x}^{(0)} + \boldsymbol{z}) \perp \mathcal{L} \quad \text{或} \quad \boldsymbol{r}^{(0)} - \boldsymbol{A}\boldsymbol{z} \perp \mathcal{L}.$$

换言之, 可定义近似解为

$$\widetilde{\boldsymbol{x}} = \boldsymbol{x}^{(0)} + \boldsymbol{z}, \quad \boldsymbol{z} \in \mathcal{K}, \tag{2.46}$$

$$(\boldsymbol{r}^{(0)} - \boldsymbol{A}\boldsymbol{z}, \boldsymbol{w}) = 0, \quad \forall\, \boldsymbol{w} \in \mathcal{L}. \tag{2.47}$$

这是最一般形式的一个基本投影步. 大多数标准技术都连续地使用这种投影. 投影方法为科学计算中许多著名的方法提供了统一

的框架. 事实上, 几乎所有的基本迭代法都可视为投影技术. 只要通过 m 个自由度 (子空间 $\mathcal{K}$) 和 m 个约束 (子空间 $\mathcal{L}$) 来定义一个近似, 就可得到一个投影过程.

正交投影方法相应于两个子空间 $\mathcal{K}$ 和 $\mathcal{L}$ 相等的特殊情形. 这种差别在 Heimite 的情形下特别重要, 因为在此情形下, 需要保证被投影的问题是 Heimite 的. 当 $\mathcal{K} = \mathcal{L}$ 时, Petrov–Galerkin 条件也简称为 Galerkin 条件.

设 $\boldsymbol{V} = [\boldsymbol{v}_1, \cdots, \boldsymbol{v}_m]$ 是一个 $n \times m$ 阶矩阵, 其列向量形成 $\mathcal{K}$ 的一组基. 类似地, $\boldsymbol{W} = [\boldsymbol{w}_1, \cdots, \boldsymbol{w}_m]$ 也是一个 $n \times m$ 阶矩阵, 其列向量形成 $\mathcal{L}$ 的一组基. 如果将近似解写成

$$\boldsymbol{x} = \boldsymbol{x}^{(0)} + \boldsymbol{V}\boldsymbol{y},$$

则正交性条件立即导致下面关于向量 $\boldsymbol{y}$ 的方程组:

$$\boldsymbol{W}^{\mathrm{T}}\boldsymbol{A}\boldsymbol{V}\boldsymbol{y} = \boldsymbol{W}^{\mathrm{T}}\boldsymbol{r}^{(0)}.$$

如果假设 $m \times m$ 阶矩阵 $\boldsymbol{W}^{\mathrm{T}}\boldsymbol{A}\boldsymbol{V}$ 是非奇异的, 则有下述关于近似解 $\boldsymbol{x}$ 的表达式:

$$\boldsymbol{x} = \boldsymbol{x}^{(0)} + \boldsymbol{V}(\boldsymbol{W}^{\mathrm{T}}\boldsymbol{A}\boldsymbol{V})^{-1}\boldsymbol{W}^{\mathrm{T}}\boldsymbol{r}^{(0)}. \tag{2.48}$$

在许多算法中, 没有必要形成矩阵 $\boldsymbol{W}^{\mathrm{T}}\boldsymbol{A}\boldsymbol{V}$, 因此它可作为对算法的一个“副产品”供使用. 典型投影方法由下述算法表示.

算法 2.15 (投影方法的基本框架)

步 1, 选取一对子空间 $\mathcal{K}$ 和 $\mathcal{L}$.

步 2, 选取 $\mathcal{K}$ 和 $\mathcal{L}$ 的基 $\boldsymbol{V} = [\boldsymbol{v}_1, \cdots, \boldsymbol{v}_m]$ 和 $\boldsymbol{W} = [\boldsymbol{w}_1, \cdots, \boldsymbol{w}_m]$.

步 3, 计算残差 $\boldsymbol{r} = \boldsymbol{b} - \boldsymbol{A}\boldsymbol{x}$.

步 4, 解方程组 $\boldsymbol{W}^{\mathrm{T}}\boldsymbol{A}\boldsymbol{V}\boldsymbol{y}=\boldsymbol{W}^{\mathrm{T}}\boldsymbol{r}$ 得到 $\boldsymbol{y}$.

步 5, 计算近似解 $\boldsymbol{x}:=\boldsymbol{x}+\boldsymbol{V}\boldsymbol{y}$.

只有当矩阵 $\boldsymbol{W}^{\mathrm{T}}\boldsymbol{A}\boldsymbol{V}$ 是非奇异的, 近似解才是有意义的. 即使当 $\boldsymbol{A}$ 是非奇异的, 也不能保证 $\boldsymbol{W}^{\mathrm{T}}\boldsymbol{A}\boldsymbol{V}$ 是非奇异的. 看下面的例子.

例 2.8 考虑矩阵

$$\boldsymbol{A}=\begin{bmatrix}\boldsymbol{O} & \boldsymbol{I}\\ \boldsymbol{I} & \boldsymbol{I}\end{bmatrix},$$

式中: $\boldsymbol{I}$ 为 m 阶单位矩阵且 $\boldsymbol{O}$ 为 m 阶零矩阵. 设 $\boldsymbol{V}=\boldsymbol{W}=[\boldsymbol{e}_1,\boldsymbol{e}_2,\cdots,\boldsymbol{e}_m]$, 其中 $\boldsymbol{e}_i$ 是第 i 个分量为 1 而其余分量均为 0 的 $2m$ 维列向量. 尽管 $\boldsymbol{A}$ 是非奇异的, 而矩阵 $\boldsymbol{W}^{\mathrm{T}}\boldsymbol{A}\boldsymbol{V}=\boldsymbol{O}$, 因此是奇异的.

下面的命题表明, 有两种重要的特殊情形可以保证 $\boldsymbol{W}^{\mathrm{T}}\boldsymbol{A}\boldsymbol{V}$ 的非奇异性.

命题 2.1 设 $\boldsymbol{A}$, $\mathcal{K}$ 和 $\mathcal{L}$ 满足下面两个条件之一:

(1) $\boldsymbol{A}$ 是正定的且 $\mathcal{L} = \mathcal{K}$.

(2) $\boldsymbol{A}$ 是非奇异的且 $\mathcal{L} = \boldsymbol{A}\mathcal{K}$.

则矩阵 $\boldsymbol{B} = \boldsymbol{W}^{\mathrm{T}}\boldsymbol{A}\boldsymbol{V}$ 对 $\mathcal{K}$ 和 $\mathcal{L}$ 的任意基 $\boldsymbol{V}$ 和 $\boldsymbol{W}$ 均是非奇异的.

证明 (1) 设 $\boldsymbol{V}$ 和 $\boldsymbol{W}$ 分别为 $\mathcal{K}$ 和 $\mathcal{L}$ 的任意一组基. 事实上, 由于 $\mathcal{L} = \mathcal{K}$, $\boldsymbol{W}$ 总可以表示为 $\boldsymbol{W} = \boldsymbol{V}\boldsymbol{G}$, 其中 $\boldsymbol{G}$ 是一个非奇异 m 阶矩阵, 则

$$\boldsymbol{B} = \boldsymbol{W}^{\mathrm{T}}\boldsymbol{A}\boldsymbol{V} = \boldsymbol{G}^{\mathrm{T}}\boldsymbol{V}^{\mathrm{T}}\boldsymbol{A}\boldsymbol{V}.$$

由于 $\boldsymbol{A}$ 是正定的, 所以 $\boldsymbol{V}^{\mathrm{T}}\boldsymbol{A}\boldsymbol{V}$ 正定, 从而证明了 $\boldsymbol{B}$ 是非奇异的.

(2) 设 $\boldsymbol{V}$ 和 $\boldsymbol{W}$ 分别为 $\mathcal{K}$ 和 $\mathcal{L}$ 的任意一组基. 由于 $\mathcal{L} = \boldsymbol{A}\mathcal{K}$, 此时 $\boldsymbol{W}$ 总可表示为 $\boldsymbol{W} = \boldsymbol{AVG}$, 其中 $\boldsymbol{G}$ 是一个非奇异 m 阶矩阵, 则

$$\boldsymbol{B} = \boldsymbol{W}^{\mathrm{T}}\boldsymbol{AV} = \boldsymbol{G}^{\mathrm{T}}(\boldsymbol{AV})^{\mathrm{T}}\boldsymbol{AV}.$$

由于 $\boldsymbol{A}$ 是非奇异的, $n \times m$ 阶矩阵 $\boldsymbol{AV}$ 是满秩的, 因此 $(\boldsymbol{AV})^{\mathrm{T}}\boldsymbol{AV}$ 是非奇异的. 由此及上式证明了 $\boldsymbol{B}$ 是非奇异的. 证毕. □

注 2.14 考虑特殊情形: $\boldsymbol{A}$ 是实对称的且使用正交投影, 此时, 对 $\mathcal{L}$ 和 $\mathcal{K}$ 使用相同的基, 因为它们有相同的子空间, 则投影矩阵 $\boldsymbol{B} = \boldsymbol{V}^{\mathrm{T}}\boldsymbol{AV}$ 是对称的. 另外, 如果矩阵 $\boldsymbol{A}$ 是对称正定的, 则 $\boldsymbol{B}$ 也是对称正定的.

进一步, 分别考察命题 2.1 中的两个条件所对应的最优性结果.

命题 2.2 假设 $\boldsymbol{A}$ 是对称正定的且 $\mathcal{L} = \mathcal{K}$, 初始向量为 $\boldsymbol{x}^{(0)}$.

则向量 $\widetilde{\boldsymbol{x}}$ 为到 $\mathcal{K}$ 上的一个正交投影方法的结果, 当且仅当 $\widetilde{\boldsymbol{x}}$ 在 $\boldsymbol{x}^{(0)}+\mathcal{K}$ 上极小化误差的 $\boldsymbol{A}$ 范数 (能量范数), 即当且仅当

$$\mathcal{E}(\widetilde{\boldsymbol{x}}) = \min_{\boldsymbol{x}\in\boldsymbol{x}^{(0)}+\mathcal{K}} \mathcal{E}(\boldsymbol{x}),$$

式中: $\mathcal{E}(\boldsymbol{x}) \equiv \|\boldsymbol{x}^* - \boldsymbol{x}\|_{\boldsymbol{A}} = \sqrt{(\boldsymbol{A}(\boldsymbol{x}^* - \boldsymbol{x}), \boldsymbol{x}^* - \boldsymbol{x})}$.

证明 由推论 2.2, 为使 $\widetilde{\boldsymbol{x}}$ 极小化 $\mathcal{E}(\boldsymbol{x})$, 当且仅当 $\boldsymbol{x}^* - \widetilde{\boldsymbol{x}}$ 与 $\mathcal{K}$ 的所有向量均 $\boldsymbol{A}$ 正交. 由此得

$$(\boldsymbol{A}(\boldsymbol{x}^* - \widetilde{\boldsymbol{x}}), \boldsymbol{v}) = 0, \quad \forall\, \boldsymbol{v} \in \mathcal{K},$$

或等价地, 有

$$(\boldsymbol{b} - \boldsymbol{A}\widetilde{\boldsymbol{x}}, \boldsymbol{v}) = 0, \quad \forall\, \boldsymbol{v} \in \mathcal{K},$$

这正是对近似解 $\widetilde{\boldsymbol{x}}$ 定义一个正交投影的 Galerkin 条件. 证毕. □




Back
Close

命题 2.3 设 $\boldsymbol{A}$ 是任意方阵且 $\mathcal{L} = \boldsymbol{A}\mathcal{K}$, 初始向量为 $\boldsymbol{x}^{(0)}$. 则向量 $\widetilde{\boldsymbol{x}}$ 为正交于 $\mathcal{L}$ 的到 $\mathcal{K}$ 上一个斜投影方法的结果, 当且仅当 $\widetilde{\boldsymbol{x}}$ 在 $\boldsymbol{x} \in \boldsymbol{x}^{(0)} + \mathcal{K}$ 上极小化残差向量 $\boldsymbol{r}(\boldsymbol{x}) = \boldsymbol{b} - \boldsymbol{A}\boldsymbol{x}$ 的 2–范数, 即当且仅当

$$\|\boldsymbol{r}(\widetilde{\boldsymbol{x}})\|_2 = \min_{\boldsymbol{x} \in \boldsymbol{x}^{(0)} + \mathcal{K}} \|\boldsymbol{r}(\boldsymbol{x})\|_2.$$

证明 如前所见, 为使 $\widetilde{\boldsymbol{x}}$ 极小化 $\|\boldsymbol{r}(\boldsymbol{x})\|_2$, 其充分必要条件是 $\boldsymbol{b} - \boldsymbol{A}\widetilde{\boldsymbol{x}}$ 与所有形如 $\boldsymbol{v} = \boldsymbol{A}\boldsymbol{y}$ 的向量均正交, 其中 $\boldsymbol{y} \in \mathcal{K}$. 亦即

$$(\boldsymbol{b} - \boldsymbol{A}\widetilde{\boldsymbol{x}}, \boldsymbol{v}) = 0, \quad \forall\, \boldsymbol{v} \in \boldsymbol{A}\mathcal{K},$$

这正是定义近似解 $\widetilde{\boldsymbol{x}}$ 的 Petrov–Galerkin 条件. 证毕. □

值得指出的是, 上述命题中 $\boldsymbol{A}$ 不需要是非奇异的. 当 $\boldsymbol{A}$ 为奇异时, 可能有无穷多个向量 $\widetilde{\boldsymbol{x}}$ 满足最优性条件.

下面给出用投影算子术语的解释. 对 $\mathcal{L} = \mathcal{K}$ 和 $\mathcal{L} = \boldsymbol{A}\mathcal{K}$ 的




情形, 可容易地用在初始残差或初始误差上正交投影算子的作用来解释投影方法的结果. 先考虑 $\mathcal{L} = \boldsymbol{A}\mathcal{K}$ 的情形, 设初始残差为 $\boldsymbol{r}^{(0)} = \boldsymbol{b} - \boldsymbol{A}\boldsymbol{x}^{(0)}$, 且 $\widetilde{\boldsymbol{r}} = \boldsymbol{b} - \boldsymbol{A}\widetilde{\boldsymbol{x}}$ 为投影过程之后所得的残差, 则

$$\widetilde{\boldsymbol{r}} = \boldsymbol{b} - \boldsymbol{A}(\boldsymbol{x}^{(0)} + \boldsymbol{z}) = \boldsymbol{r}^{(0)} - \boldsymbol{A}\boldsymbol{z}.$$

其中 $\boldsymbol{z}$ 由强加条件 $\boldsymbol{r}^{(0)} - \boldsymbol{A}\boldsymbol{z} \perp \boldsymbol{A}\mathcal{K}$ 而得到. 因此, 向量 $\boldsymbol{A}\boldsymbol{z}$ 是向量 $\boldsymbol{r}^{(0)}$ 到子空间 $\boldsymbol{A}\mathcal{K}$ 上的正交投影. 从而有下面的命题.

命题 2.4 设 $\widetilde{\boldsymbol{x}}$ 为由正交于 $\mathcal{L} = \boldsymbol{A}\mathcal{K}$ 的到 $\mathcal{K}$ 上的一个投影过程所得的近似解, 且设 $\widetilde{\boldsymbol{r}} = \boldsymbol{b} - \boldsymbol{A}\widetilde{\boldsymbol{x}}$ 为相应的残差, 则

$$\widetilde{\boldsymbol{r}} = (\boldsymbol{I} - \boldsymbol{P})\boldsymbol{r}^{(0)}, \tag{2.49}$$

式中: $\boldsymbol{P}$ 为子空间 $\boldsymbol{A}\mathcal{K}$ 上的正交投影算子.

命题 2.4 说明一个投影步后所得残差的 2–范数将不超过初始

残差的 2–范数, 即

$$\|\widetilde{\boldsymbol{r}}\|_2 \leqslant \|\boldsymbol{r}^{(0)}\|_2,$$

这是一个已经建立了的结果. 这类方法称为*残差投影方法*.

现在考虑 $\mathcal{L} = \mathcal{K}$ 且 $\boldsymbol{A}$ 是对称正定的情形. 设 $\boldsymbol{e}^{(0)} = \boldsymbol{x}^* - \boldsymbol{x}^{(0)}$ 为初始误差, 类似地, 设 $\widetilde{\boldsymbol{e}} = \boldsymbol{x}^* - \widetilde{\boldsymbol{x}}$, 其中 $\widetilde{\boldsymbol{x}} = \boldsymbol{x}^{(0)} + \boldsymbol{z}$ 为由投影步所得的近似解. 由式 (2.49), 得

$$\boldsymbol{A}\widetilde{\boldsymbol{e}} = \boldsymbol{A}(\boldsymbol{x}^* - \widetilde{\boldsymbol{x}}) = \widetilde{\boldsymbol{r}} = \boldsymbol{A}(\boldsymbol{e}^{(0)} - \boldsymbol{z}) = \boldsymbol{r}^{(0)} - \boldsymbol{A}\boldsymbol{z},$$

式中: $\boldsymbol{z}$ 是由限制残差向量 $\boldsymbol{r}^{(0)} - \boldsymbol{A}\boldsymbol{z} \perp \mathcal{K}$ 而得到, 即

$$(\boldsymbol{r}^{(0)} - \boldsymbol{A}\boldsymbol{z}, \boldsymbol{w}) = 0, \ \forall\, \boldsymbol{w} \in \mathcal{K} \iff (\boldsymbol{A}(\boldsymbol{e}^{(0)} - \boldsymbol{z}), \boldsymbol{w}) = 0, \ \forall\, \boldsymbol{w} \in \mathcal{K}.$$

由于 $\boldsymbol{A}$ 是对称正定的, 它定义了一种通常表示为 $(\cdot,\cdot)_{\boldsymbol{A}}$ 的内积, 则条件变为

$$(\boldsymbol{e}^{(0)} - \boldsymbol{z}, \boldsymbol{w})_{\boldsymbol{A}} = 0, \ \ \forall\, \boldsymbol{w} \in \mathcal{K}.$$




Back
Close

此条件可解释为: 向量 z 是初始误差向量 $e^{(0)}$ 到子空间 $\mathcal{K}$ 上的 A 正交投影.

命题 2.5 设 $\widetilde{x}$ 为到 $\mathcal{K}$ 上的一个正交投影过程所得的近似解, 且设 $\widetilde{e} = x^* - \widetilde{x}$ 为相应的误差, 则

$$\widetilde{e} = (I - P_A)e^{(0)},$$

式中: P_A 为子空间 $\mathcal{K}$ 上的投影算子, 它关于 A 内积是正交的.

命题 2.5 表明一个投影步后所得的误差向量的 A 范数不超过初始误差的 A 范数, 即

$$\|\widetilde{e}\|_A \leqslant \|e^{(0)}\|_A,$$

这正是所期望的, 因为已知误差的 A 范数在 $x^{(0)} + \mathcal{K}$ 中被极小化. 这类方法称为误差投影方法.

§2.5.3 一维投影方法

这里给出几种基于一维投影过程的投影方法. 以下用 $\boldsymbol{r}$ 表示当前近似解 $\boldsymbol{x}$ 的残差向量 $\boldsymbol{r} = \boldsymbol{b} - \boldsymbol{A}\boldsymbol{x}$. 为避免下标, 用 "$:=$" 表示向量校正, 即 $\boldsymbol{x} := \boldsymbol{x} + \alpha\boldsymbol{r}$ 意味着 "计算 $\boldsymbol{x} + \alpha\boldsymbol{r}$ 并用结果覆盖当前的 $\boldsymbol{x}$".

一维投影过程可由下面定义, 即

$$\mathcal{K} = \text{span}\{\boldsymbol{v}\} \quad \text{且} \quad \mathcal{L} = \text{span}\{\boldsymbol{w}\},$$

式中: $\boldsymbol{v}$ 和 $\boldsymbol{w}$ 为两个向量.

此时, 新的近似取为 $\boldsymbol{x} := \boldsymbol{x} + \alpha\boldsymbol{v}$ 且 Petrov–Galerkin 条件 $\boldsymbol{r} - \boldsymbol{A}\boldsymbol{z} \perp \boldsymbol{w}$ 导致

$$\alpha = \frac{(\boldsymbol{r}, \boldsymbol{w})}{(\boldsymbol{A}\boldsymbol{v}, \boldsymbol{w})}.$$

下面考虑三种流行的选择.

1) 最速下降法

最速下降法对矩阵 $\boldsymbol{A}$ 为对称正定的情形来定义. 它由每一步取 $\boldsymbol{v} = \boldsymbol{r}$ 和 $\boldsymbol{w} = \boldsymbol{r}$ 组成 (即一维正交投影). 这导致下面描述的一个迭代.

算法 2.16 (最速下降法)

步 1, $\boldsymbol{r} := \boldsymbol{b} - \boldsymbol{A}\boldsymbol{x}$.

步 2, $\alpha \leftarrow (\boldsymbol{r}, \boldsymbol{r})/(\boldsymbol{A}\boldsymbol{r}, \boldsymbol{r})$.

步 3, $\boldsymbol{x} := \boldsymbol{x} + \alpha \boldsymbol{r}$.

此迭代的每一步在所有形式为 $\boldsymbol{x} + \alpha \boldsymbol{r}$ 的向量上极小化

$$f(\boldsymbol{x}) = \|\boldsymbol{x} - \boldsymbol{x}^*\|_{\boldsymbol{A}}^2 = (\boldsymbol{A}(\boldsymbol{x} - \boldsymbol{x}^*), \boldsymbol{x} - \boldsymbol{x}^*),$$

式中: $\boldsymbol{r}$ 为负梯度方向 $-\nabla f(\boldsymbol{x})$.

负梯度方向是使 $f(\boldsymbol{x})$ 局部最快速下降的方向. 其次, 证明当 $\boldsymbol{A}$ 是对称正定的时, 收敛性是有保证的. 它是下面著名的 Kantorovich (康托洛维奇) 不等式的一个结论.

引理 2.2 (Kantorovich 不等式) 设 $\boldsymbol{B}$ 为任意对称正定实矩阵且 $\lambda_{\max}$, $\lambda_{\min}$ 为其最大、最小特征值, 则

$$\frac{(\boldsymbol{B}\boldsymbol{x},\boldsymbol{x})(\boldsymbol{B}^{-1}\boldsymbol{x},\boldsymbol{x})}{(\boldsymbol{x},\boldsymbol{x})^2} \leqslant \frac{(\lambda_{\max}+\lambda_{\min})^2}{4\lambda_{\max}\lambda_{\min}}, \quad \forall\, \boldsymbol{x} \neq \boldsymbol{0}. \tag{2.50}$$

定理 2.14 设 $\boldsymbol{A}$ 是对称正定的. 则由算法 2.16 生成的误差向量 $\boldsymbol{e}^{(k)} = \boldsymbol{x}^* - \boldsymbol{x}^{(k)}$ 的 $\boldsymbol{A}$ 范数满足关系

$$\|\boldsymbol{e}^{(k+1)}\|_{\boldsymbol{A}} \leqslant \frac{\lambda_{\max}-\lambda_{\min}}{\lambda_{\max}+\lambda_{\min}}\|\boldsymbol{e}^{(k)}\|_{\boldsymbol{A}}, \tag{2.51}$$

且算法 2.16 对任意初始向量 $\boldsymbol{x}^{(0)}$ 都是收敛的.




证明 首先注意到 $\|\boldsymbol{e}^{(k+1)}\|_{\boldsymbol{A}}^2 = (\boldsymbol{A}\boldsymbol{e}^{(k+1)}, \boldsymbol{e}^{(k+1)}) = (\boldsymbol{r}^{(k+1)}, \boldsymbol{e}^{(k+1)})$, 则由简单的代入有

$$\|\boldsymbol{e}^{(k+1)}\|_{\boldsymbol{A}}^2 = (\boldsymbol{r}^{(k+1)}, \boldsymbol{e}^{(k)} - \alpha_k \boldsymbol{r}^{(k)}).$$

由于新残差向量 $\boldsymbol{r}^{(k+1)}$ 必须正交于搜索方向 $\boldsymbol{r}^{(k)}$, 上式右端项中第 2 项为零. 因此

$$\begin{aligned}
\|\boldsymbol{e}^{(k+1)}\|_{\boldsymbol{A}}^2 &= (\boldsymbol{r}^{(k)} - \alpha_k \boldsymbol{A}\boldsymbol{r}^{(k)}, \boldsymbol{e}^{(k)}) \\
&= (\boldsymbol{r}^{(k)}, \boldsymbol{A}^{-1}\boldsymbol{r}^{(k)}) - \alpha_k (\boldsymbol{r}^{(k)}, \boldsymbol{r}^{(k)}) \\
&= \|\boldsymbol{e}^{(k)}\|_{\boldsymbol{A}}^2 \left(1 - \frac{(\boldsymbol{r}^{(k)}, \boldsymbol{r}^{(k)})}{(\boldsymbol{r}^{(k)}, \boldsymbol{A}\boldsymbol{r}^{(k)})} \cdot \frac{(\boldsymbol{r}^{(k)}, \boldsymbol{r}^{(k)})}{(\boldsymbol{r}^{(k)}, \boldsymbol{A}^{-1}\boldsymbol{r}^{(k)})}\right).
\end{aligned}$$

应用 Kantorovich 不等式 (2.50) 即得定理的结论. 证毕. □

2) 极小残差 (MR) 法

现在假设 $\boldsymbol{A}$ 不必为对称的而只是正定的, 即其对称部分 $\boldsymbol{A}+\boldsymbol{A}^{\mathrm{T}}$ 是对称正定的. 在每一步取 $\boldsymbol{v}=\boldsymbol{r}$ 和 $\boldsymbol{w}=\boldsymbol{A}\boldsymbol{r}$, 可得下述迭代过程.

算法 2.17 (极小残差法)

步 1, $\boldsymbol{r}:=\boldsymbol{b}-\boldsymbol{A}\boldsymbol{x}$.

步 2, $\alpha:=(\boldsymbol{A}\boldsymbol{r},\boldsymbol{r})/(\boldsymbol{A}\boldsymbol{r},\boldsymbol{A}\boldsymbol{r})$.

步 3, $\boldsymbol{x}:=\boldsymbol{x}+\alpha\boldsymbol{r}$.

这里, 每一步在方向 $\boldsymbol{r}$ 上极小化 $\|\boldsymbol{b}-\boldsymbol{A}\boldsymbol{x}\|_2^2$. 如下定理所述, 迭代在 $\boldsymbol{A}$ 为正定时收敛.

定理 2.15 设 $\boldsymbol{A}$ 为实正定矩阵, 且设

$$\mu=\lambda_{\min}(\boldsymbol{A}+\boldsymbol{A}^{\mathrm{T}})/2,\quad \sigma=\|\boldsymbol{A}\|_2.$$

则由算法 2.17 产生的残差向量满足关系

$$\|\boldsymbol{r}^{(k+1)}\|_2 \leqslant \left(1-\frac{\mu^2}{\sigma^2}\right)^{1/2}\|\boldsymbol{r}^{(k)}\|_2, \tag{2.52}$$

且算法 2.17 对任意初始向量 $\boldsymbol{x}^{(0)}$ 均收敛.

证明 类似于最速下降法来处理, 有关系式

$$\begin{aligned}\|\boldsymbol{r}^{(k+1)}\|_2^2 &= (\boldsymbol{r}^{(k)}-\alpha_k\boldsymbol{A}\boldsymbol{r}^{(k)}, \boldsymbol{r}^{(k)}-\alpha_k\boldsymbol{A}\boldsymbol{r}^{(k)})\\ &= (\boldsymbol{r}^{(k)}-\alpha_k\boldsymbol{A}\boldsymbol{r}^{(k)}, \boldsymbol{r}^{(k)})-\alpha_k(\boldsymbol{r}^{(k)}-\alpha_k\boldsymbol{A}\boldsymbol{r}^{(k)}, \boldsymbol{A}\boldsymbol{r}^{(k)}).\end{aligned}$$

由构造, 新的残差向量 $\boldsymbol{r}^{(k)}-\alpha_k\boldsymbol{A}\boldsymbol{r}^{(k)}$ 必须正交于搜索方向 $\boldsymbol{A}\boldsymbol{r}^{(k)}$, 因此, 上式右端项中第 2 项为零, 且得

$$\|\boldsymbol{r}^{(k+1)}\|_2^2 = (\boldsymbol{r}^{(k)} - \alpha_k \boldsymbol{A}\boldsymbol{r}^{(k)}, \boldsymbol{r}^{(k)}) = (\boldsymbol{r}^{(k)}, \boldsymbol{r}^{(k)}) - \alpha_k (\boldsymbol{A}\boldsymbol{r}^{(k)}, \boldsymbol{r}^{(k)})$$

$$= \|\boldsymbol{r}^{(k)}\|_2^2 \left(1 - \frac{(\boldsymbol{A}\boldsymbol{r}^{(k)}, \boldsymbol{r}^{(k)})}{(\boldsymbol{r}^{(k)}, \boldsymbol{r}^{(k)})} \cdot \frac{(\boldsymbol{A}\boldsymbol{r}^{(k)}, \boldsymbol{r}^{(k)})}{(\boldsymbol{A}\boldsymbol{r}^{(k)}, \boldsymbol{A}\boldsymbol{r}^{(k)})}\right)$$

$$= \|\boldsymbol{r}^{(k)}\|_2^2 \left(1 - \frac{(\boldsymbol{A}\boldsymbol{r}^{(k)}, \boldsymbol{r}^{(k)})^2}{(\boldsymbol{r}^{(k)}, \boldsymbol{r}^{(k)})^2} \cdot \frac{\|\boldsymbol{r}^{(k)}\|_2^2}{\|\boldsymbol{A}\boldsymbol{r}^{(k)}\|_2^2}\right). \tag{2.53}$$

因为 $\boldsymbol{A}$ 为实正定矩阵, 从而有

$$\frac{(\boldsymbol{A}\boldsymbol{x}, \boldsymbol{x})}{(\boldsymbol{x}, \boldsymbol{x})} = \frac{1}{2}\frac{((\boldsymbol{A} + \boldsymbol{A}^{\mathrm{T}})\boldsymbol{x}, \boldsymbol{x})}{(\boldsymbol{x}, \boldsymbol{x})} \geqslant \mu > 0, \tag{2.54}$$

式中: $\mu = \lambda_{\min}(\boldsymbol{A} + \boldsymbol{A}^{\mathrm{T}})/2$.

由不等式 $\|\boldsymbol{A}\boldsymbol{r}^{(k)}\|_2 \leqslant \|\boldsymbol{A}\|_2 \|\boldsymbol{r}^{(k)}\|_2$ 立即可得定理的结论. 证毕. □

3) 残差范数最速下降法

在残差范数最速下降法中, 只需 $\boldsymbol{A}$ 为非奇异矩阵. 在每一步, 算法使用 $\boldsymbol{v} = \boldsymbol{A}^{\mathrm{T}}\boldsymbol{r}$ 和 $\boldsymbol{w} = \boldsymbol{A}\boldsymbol{v}$, 给出下面运算序列:

$$\begin{aligned} &\boldsymbol{r} := \boldsymbol{b} - \boldsymbol{A}\boldsymbol{x}, \quad \boldsymbol{v} = \boldsymbol{A}^{\mathrm{T}}\boldsymbol{r}, \\ &\alpha := \|\boldsymbol{v}\|_2^2 / \|\boldsymbol{A}\boldsymbol{v}\|_2^2, \\ &\boldsymbol{x} := \boldsymbol{x} + \alpha\boldsymbol{v}. \end{aligned}$$

注意到基于上述运算序列的算法需要三个矩阵向量乘积, 这是本节其他算法的 3 倍. 通过不同的方式计算残差, 每步中矩阵向量乘积的个数可降为两个. 这种变形如下:

算法 2.18 (残差范数最速下降)

步 1, 计算 $\boldsymbol{r} := \boldsymbol{b} - \boldsymbol{A}\boldsymbol{x}$.

步 2, $\boldsymbol{v} := \boldsymbol{A}^{\mathrm{T}}\boldsymbol{r}$.

步 3, 计算 $\boldsymbol{Av}$ 和 $\alpha := \|\boldsymbol{v}\|_2^2/\|\boldsymbol{Av}\|_2^2$.

步 4, $\boldsymbol{x} := \boldsymbol{x} + \alpha\boldsymbol{r}$.

步 5, $\boldsymbol{r} := \boldsymbol{r} - \alpha\boldsymbol{Av}$.

这里, 每一步在方向 $-\nabla f(\boldsymbol{x})$ 上极小化 $\|\boldsymbol{b}-\boldsymbol{Ax}\|_2^2$. 由此得出, 这等价于最速下降算法应用于法方程. 因为当 $\boldsymbol{A}$ 是非奇异时, $\boldsymbol{A}^{\mathrm{T}}\boldsymbol{Ax} = \boldsymbol{A}^{\mathrm{T}}\boldsymbol{b}$ 是正定的, 则根据定理 2.15, 只要 $\boldsymbol{A}$ 是非奇异的, 方法将收敛.