# 数值分析

## 第六章：线性方程组迭代法

张亚楠[1]

苏州大学数学科学学院

April 2, 2020

[1]Email: ynzhang@suda.edu.cn

## Contents

$Ax = b$

1. A进行三角分解之后$A = LU$, 解方程的工作量是$O(n^2)$.
2. 迭代法的目标是降低计算复杂度到$\boldsymbol{O}(n)$.
3. 迭代法适用于大型稀疏矩阵，迭代一次的工作量与矩阵的稀疏程度有关

偏微分方程的差分方法和有限元方法产生的代数方程，其系数矩阵都是稀疏的； 例如椭圆算子的标准五点差分格式，系数矩阵的每行只有5个非零元素，迭代一次需要5n次乘法。 若n很大且迭代收敛较快时，迭代法的运算量只有O$(n)$

记 $x^*$ 是

$$Ax = b$$

的精确解

**Remark 1.1**

1. 直接法的目标是找到 $x^*$ 本身
2. 迭代法的目标是找到 $x^*$ 的有效近似解，例如

$$\|x - x^*\| < \tau = 10^{-6}$$

则 $x$ 即是所求。 只要满足容许的误差 $\tau$ 即可

如何得到近似解 $x$? 从方程组出发构造迭代格式，迭代格式产生向量序列 $\boldsymbol{x}^k$,

$$\boldsymbol{x}^{(0)}, x^{(1)}, ..., x^{(k)}, ...$$

向量序列收敛即可！！

# 回顾向量范数

设 $\boldsymbol{x} \in \boldsymbol{R}^n$, 内积空间诱导范数一般称之为2范数。更一般的，可以定义p范数

$$\boldsymbol{x} = (x_1, x_2, ..., x_n)^{\mathrm{T}}; \quad \|\boldsymbol{x}\|_p = \left(\sum_{j=1}^{n} |x_j|^p\right)^{\frac{1}{p}}, \; 1 \le p < +\infty$$

## 本课程常用三种范数

- 1-范数，p = 1,

$$\|\boldsymbol{x}\|_1 = \sum_{j=1}^{n} |x_j|$$

- 2-范数，p = 2,

$$\|\boldsymbol{x}\|_2 = \left(\sum_{j=1}^{n} |x_j|^2\right)^{\frac{1}{2}}$$

- ∞-范数， p→ ∞

$$\|\boldsymbol{x}\|_\infty = \lim_{p\to\infty} \|\boldsymbol{x}\|_p = \lim_{p\to\infty} \left(\sum_{j=1}^{n} |x_j|^p\right)^{\frac{1}{p}} = \max_{1\le j\le n} |x_j|$$

**符号$\|\cdot\|$可以看作是一个映射，值域是$[0,+\infty)$, 满足**

(1) 正定性 $\|\boldsymbol{x}\| \geq 0$ 且 等号成立当且紧当x是零元

(2) 齐次性 $\|\alpha\,\boldsymbol{x}\| = |\alpha| \cdot \|\boldsymbol{x}\|$，$\forall \alpha \in \mathrm{C}$

(3) 三角形不等式 $\|\boldsymbol{x}+\boldsymbol{y}\| \leq \|\boldsymbol{x}\| + \|\boldsymbol{y}\|$

Tips: 范数用来衡量一个向量的大小，长度， 类似于实数或复数的绝对值； 运算结果是一个非负数.

从映射的角度看，n维向量的范数又是一个n元函数； 可以证明该函数(此时称之为范数)对每一个自变量(此时称之为向量的分量)均是连续的. 即

$$\left\| \begin{bmatrix} | \\ x_k + \varepsilon \\ | \end{bmatrix} - \begin{bmatrix} | \\ x_k \\ | \end{bmatrix} \right\| \to 0 \quad \text{as} \quad \varepsilon \to 0$$

由此，可以证明如下重要结论

**Theorem 1**

$\boldsymbol{R}^n$ 空间中任意两个范数 均是等价的. 即存在正数c, 对任意的$\boldsymbol{x} \in \boldsymbol{R}^n$, 有

$$\frac{1}{c}\|\boldsymbol{x}\|_\diamondsuit \le \|\boldsymbol{x}\|_\heartsuit \le c\|\boldsymbol{x}\|_\diamondsuit$$

注：$\boldsymbol{R}^n$是有限维空间，完备的线性赋范空间(Banach)；更多信息可参考：泛函分析！

如要说明一个向量序列收敛，则只需要在其中一个范数下证明收敛即可。

**Definition 2**

设 $\{\boldsymbol{x}^{(k)}\}$ 是 $\boldsymbol{R}^n$ 中的向量序列，$\boldsymbol{x}^* \in \boldsymbol{R}^n$, 若对每一个向量分量，都有

$$\lim_{k\to\infty} x_j^{(k)} = x_j^*, \quad 1 \le j \le n$$

则称序列$\boldsymbol{x}^{(k)}$收敛到 $\boldsymbol{x}^*$, 记为

$$\lim_{k\to\infty} \boldsymbol{x}^{(k)} = \boldsymbol{x}^*$$

**Theorem 3**

$\boldsymbol{R}^n$ 中序列收敛等价于以任何一种范数收敛

$$\lim_{k\to\infty} \boldsymbol{x}^{(k)} = \boldsymbol{x}^* \Leftrightarrow \lim_{k\to\infty} \|\boldsymbol{x}^{(k)} - \boldsymbol{x}^*\| = 0$$

Tips : 判断向量序列的收敛性可用数列(误差序列范数)是否趋于零考量！！

**需求：**

在误差分析中，矩阵范数大多时候与向量范数会同时出现，因此希望二者之间 有一定的联系和统一性。例如：F范数符合范数定义，但不好用！！

**Definition 4**

任给一个向量$x \in \boldsymbol{R}^n$, 矩阵 $A \in \boldsymbol{R}^{n\times n}$, 以及范数 $\|x\|_\spadesuit$,(如：$\spadesuit = 1, 2, \infty$), 可以按照如下方式定义一个从属于♠范数的 关于矩阵A的算子范数

$$\|A\|_\spadesuit = \max_{x\neq\vec{\mathbf{0}}} \frac{\|Ax\|_\spadesuit}{\|x\|_\spadesuit}$$

矩阵范数满足范数的定义，同时还具有如下性质

$$\|AB\| \leq \|A\| \cdot \|B\|; \quad \|Ax\| \leq \|A\| \cdot \|x\|$$

**Example 5**

证明:

$$\|AB\| \leq \|A\| \cdot \|B\|$$

提示:

$$\|AB\| = \max_{x \neq \vec{0}} \frac{\|ABx\|}{\|x\|} = \|AB * y\|, \quad \text{某个特殊的单位向量} \|y\| = 1$$

进而

$$\|AB * y\| \leq \|A\| * \|B * y\| \leq \|A\| \cdot \|B\| \cdot \|y\| = \|A\|\|B\|$$

**常用的三种范数 公式**

- $\|A\|_\infty = \max_{1 \leq i \leq n} \sum_{j=1}^{n} |a_{ij}|$
- $\|A\|_1 = \max_{1 \leq j \leq n} \sum_{i=1}^{n} |a_{ij}|$
- $\|A\|_2 = \sqrt{\lambda_{\max}(A^{\mathrm{T}} * A)}$ $\lambda_{\max}$指最大特征值

**Definition 6**

设$A=(a_{ij})\in \boldsymbol{R}^{n\times n}$,若存在$\lambda\in \boldsymbol{C}$ 和非零向量$\boldsymbol{x}=(x_1,x_2,...,x_n)\in \boldsymbol{R}^n$, 使得

$$Ax=\lambda x$$

则称$\lambda$为A的特征值，x为特征向量，A的全体特征值称为 A的谱，记 $\rho(A)=\max\limits_{1\le j\le n}|\lambda_j|$ 为A的谱半径。

## 最常用范数

$$\|A\|_2 = \sqrt{\lambda_{\max}(A^{\mathrm{T}} * A)} \quad \lambda_{\max}\text{指最大特征值}$$

对称矩阵的2范数就是其最大(绝对值)特征值：谱半径！！

注意到$A * A^{\mathrm{T}}$与 $A^{\mathrm{T}} * A$ 有相同的特征值

1. 矩阵的2范数即是最大奇异值singular value！
2. 矩阵$A$与其转置$A^{\mathrm{T}}$有相同的2范数
3. 对称矩阵的奇异值就是特征值！！

## 思考：

满足$P^{\mathrm{T}} * P = I$ 的正交矩阵P的2范数是多少?

根据定义检验即可。

## 两个结论:

- 对任意一个矩阵算子范数，有
$$\rho(A) \le \|A\|$$
当A是对称矩阵且取2范数时，等号成立。
- $\forall \epsilon > 0$, 存在一个矩阵算子范数 $\|\cdot\|_\diamond$, 使得
$$\|A\|_\diamond \le \rho(A) + \epsilon$$

一句话：数值上，谱半径可认为是算子范数！！

## Example 7

例7：p166

$$A = \begin{bmatrix} 1 & -2 \\ -3 & 4 \end{bmatrix}$$

计算1，2，无穷范数

解答:

$$\|A\|_\infty = 7,\ \|A\|_1 = 6,\ \|A\|_2 = 5.465,\ \rho(A) = 5.3723$$

李庆扬教材 p167 例

## Example 8

计算目标

$$\begin{bmatrix}1 & 1\\ 1 & 1.0001\end{bmatrix}\begin{bmatrix}x_1\\ x_2\end{bmatrix}=\begin{bmatrix}2\\ 2\end{bmatrix}\rightarrow\begin{bmatrix}x_1\\ x_2\end{bmatrix}=\begin{bmatrix}2\\ 0\end{bmatrix}$$

实际计算时右端项存在小的扰动误差，例如

$$\begin{bmatrix}1 & 1\\ 1 & 1.0001\end{bmatrix}\begin{bmatrix}x_1\\ x_2\end{bmatrix}=\begin{bmatrix}2\\ 2.0001\end{bmatrix}\rightarrow\begin{bmatrix}x_1\\ x_2\end{bmatrix}=\begin{bmatrix}1\\ 1\end{bmatrix}$$

万分之一的右端项扰动得到百分之百的误差？ WHY?

右端项有一微小的扰动，所得结果千差万别 。 这种现象称之为“病态”；矩阵

$$A=\begin{bmatrix}1 & 1\\ 1 & 1.0001\end{bmatrix}$$

称为病态(ill-condition)矩阵

**思考：**

为什么右端项(输入)小的扰动，解(输出)的扰动这么大?

**考查：**

右端项扰动如何传递给解？

考虑$Ax=b$; 如果右端项有微小扰动$\delta b$, 则实际求解问题为

$$A(x+\delta x)=b+\delta b \quad \rightarrow A\underline{\underline{\delta x}}=\delta b$$

分析解$x$的相对误差

$$\cancel{\frac{\delta x}{x}=?} \quad \text{or} \quad \frac{\|\delta x\|}{\|x\|}=?$$

$$\delta x=A^{-1}*\delta b \Rightarrow \underline{\|\delta x\| \le \|A^{-1}\|*\|\delta b\|}$$

$$Ax=b \Rightarrow \|Ax\|=\|b\| \Rightarrow \|A\|*\|x\| \ge \|b\| \Rightarrow \underline{\frac{1}{\|x\|} \le \frac{\|A\|}{\|b\|}}$$

可得如下关系

$$\frac{\|\delta x\|}{\|x\|} \le \underline{\|A\|*\|A^{-1}\|}\frac{\|\delta b\|}{\|b\|}$$

$$\frac{\|\delta x\|}{\|x\|} \leq \underline{\|A\| * \|A^{-1}\|} \frac{\|\delta b\|}{\|b\|}$$

右端项扰动所带来的解的扰动完全体现于<u>条件数</u>

$$\boxed{cond(A) = \|A^{-1}\| * \|A\|}$$

不同的范数对应不同的条件数

- 任意条件数均大于1

$$\mathrm{cond}(\mathrm{A}) = \|\mathrm{A}^{-1}\| * \|\mathrm{A}\| \geq \|\mathrm{A}^{-1}\mathrm{A}\| = 1$$

- 任意常数$c \neq 0$

$$\mathrm{cond}(\mathrm{cA}) = \mathrm{cond}(\mathrm{A})$$

Hint: 以上两条性质按照定义检验即可

- 无穷条件数

$$\mathrm{cond}(A)_\infty = \|A^{-1}\|_\infty * \|A\|_\infty$$

- 谱(2)条件数(最大和最小奇异值比值)

$$\mathrm{cond}(A)_2 = \|A^{-1}\|_2 * \|A\|_2 = \sqrt{\frac{\lambda_{\max}(A^T A)}{\lambda_{\min}(A^T A)}} = \frac{|\sigma_1|}{|\sigma_n|}$$

当A为对称矩阵时，奇异值即是特征值

$$\mathrm{cond}(A)_2 = \frac{|\lambda_1|}{|\lambda_n|}$$

- 当R为正交矩阵时，

$$\mathrm{cond}(R)_2 = 1; \quad \mathrm{cond}(RA)_2 = \mathrm{cond}(AR)_2 = \mathrm{cond}(A)_2$$

证明第三条性质

$$R * R^T = I \Leftrightarrow R^{-1} = R^T \to cond(R) = \|R\|_2 * \|R^T\|_2 = \|R\|_2^2 = 1$$

$$cond(RA) = \sqrt{\frac{\lambda_{Max}\left((RA)^T * RA\right)}{\lambda_{min}\left((RA)^T * RA\right)}} = \sqrt{\frac{\lambda_{Max}\left(A^T A\right)}{\lambda_{min}\left(A^T A\right)}}$$

以最为常用的谱条件数为例

$$\text{cond}(A)_2 = \|A^{-1}\|_2 * \|A\|_2 = \sqrt{\frac{\lambda_{max}(A^T A)}{\lambda_{min}(A^T A)}} = \frac{|\sigma_1|}{|\sigma_n|}$$

**计算很困难！！**

- 矩阵A的特征值，奇异值的计算难度 远远远大于解$Ax = b$
- 实际计算时，除非特别必要，一般不计算条件数！！
- 仅仅为了解方程，又想知道条件数大小，咋办？——猜！

当矩阵的行列式绝对值接近0的时候，A的条件数可能很大, 例如李庆扬p171 例9 Hilbert矩阵条件数； 或者A的元素间数量级相差很大，且无规则，A可能病态。

给定A矩阵，如果需要多次解方程，对精度要求高，且有计划做预处理，可以按照如下步骤估计其条件数

1. 给一个随机列向量$y$, 并计算$b = A * y$
2. 求解 $Ax = b$, 得到$x$, 看 $x - y$ 的大小

注意上述过程需要解一次方程，如果目标仅仅是为了解一次方程而且对精度要求不高，上述过程无意义。

**precondition**

$$Ax = b \Leftrightarrow PAx = Pb, \ \& \ cond(PA) \ll cond(A)$$

P要足够简单且容易得到！！难！！！

理论上最好的P，$PA = I$, 实际上不可取，因为不能求逆，不能求逆！！！

# Contents

**Example 9**

求解线性方程组Ax = b

$$\begin{cases} 2x+4y-2z & =2 \\ 4x+9y-3z & =8 \\ -2x-3y+7z & =10 \end{cases}$$

改写为

$$\begin{cases} 2x & =2-\Big(4y-2z\Big) \\ 9y & =8-\Big(4x-3z\Big) \\ 7z & =10-\Big(-2x-3y\Big) \end{cases} \rightarrow \begin{cases} x & =\Big[2-\Big(4y-2z\Big)\Big]/2 \\ y & =\Big[8-\Big(4x-3z\Big)\Big]/9 \\ z & =\Big[10-\Big(-2x-3y\Big)\Big]/7 \end{cases}$$

方程组 $Ax=b$改写成等价形式

$$x=Bx+f$$

将$Ax=b$改写成等价形式$x=Bx+f$, 构造迭代格式

$$\boldsymbol{x}^{(k+1)}=B\boldsymbol{x}^{(k)}+f,\quad k=0,1,2,\ldots$$

给定初值$\boldsymbol{x}^{(0)}$, 反复作用上式可得到一个向量序列

$$\boldsymbol{x}^{(0)},\boldsymbol{x}^{(1)},\boldsymbol{x}^{(2)},\ldots,\boldsymbol{x}^{(n)}\ldots$$

**思考:**

若随着迭代次数增加，向量序列不再变化了(收敛！！)，是否得到了目标解?

**Definition 10**

对上述产生的向量序列，若

$$\lim_{k\to\infty}\boldsymbol{x}^{(k)}=\boldsymbol{x}^*$$

则称迭代法收敛，且$\boldsymbol{x}^*$即是方程组的解，否则称为发散.

## 明确几个问题

1. $Ax=b \Leftrightarrow x=Bx+f$ 等价形式一定有, 例如: $x=x-c*(b-Ax)$
2. 等价形式 $\rightarrow$ 迭代格式一定有; 但未必都收敛!!
3. 迭代序列 $\boldsymbol{x}^{(k)}$ 何时收敛? 前文的引例就不收敛

**问题**

$x^{(k+1)} = B * x^{(k)} + f$何时收敛?

记$x^*$是精确解，引进误差向量

$$\boldsymbol{\varepsilon}^{(k)} = \boldsymbol{x}^{(k)} - \boldsymbol{x}^*$$

可知<u>迭代序列是否收敛等价于</u>

$$\lim_{k\to\infty} \boldsymbol{\varepsilon}^{(k)} = \vec{\mathbf{0}}$$

注意到 $x^*$满足

$$\boldsymbol{x}^* = B * \boldsymbol{x}^* + f$$

上式与迭代格式

$$\boldsymbol{x}^{(k+1)} = B * \boldsymbol{x}^{(k)} + f$$

对应相减，即得

$$\boldsymbol{\varepsilon}^{(k+1)} = B\boldsymbol{\varepsilon}^{(k)} = \dots = B^{k+1}\boldsymbol{\varepsilon}^{(0)}$$

观察上式，$\boldsymbol{\varepsilon}^{(0)}$是初始误差，一般不为零;

$\boldsymbol{\varepsilon}^{(k+1)}$是否收敛于0向量, 完全取决于迭代矩阵$B$

**问题：**

什么样的$\boldsymbol{B}$？ 对任意初值成立

$$\lim_{k\to\infty} \boldsymbol{\varepsilon}^{(k+1)} = \lim_{k\to\infty} B^{k+1}\boldsymbol{\varepsilon}^{(0)} = \mathbf{0}$$

或者

$$\lim_{k\to\infty} \|\boldsymbol{\varepsilon}^{(k+1)}\| = \lim_{k\to\infty} \|B^{k+1}\boldsymbol{\varepsilon}^{(0)}\| = 0$$

分析：

$$\|B^{k+1}\boldsymbol{\varepsilon}^{(0)}\| \le \|B^{k+1}\| * \|\boldsymbol{\varepsilon}^{(0)}\| \le \|B\|^{k+1} * \|\boldsymbol{\varepsilon}^{(0)}\|$$

**结论：**

若存在某种矩阵范数使得$\|B\| < 1$, 则迭代收敛.

说明： 存在不代表一定找得到

利用矩阵范数和谱半径之间的关系

- 对任意矩阵算子范数

$$\rho(B) \leq \|B\|$$

- $\forall \epsilon > 0$, 存在一个矩阵算子范数 $\|\cdot\|_\diamond$, 使得

$$\|B\|_\diamond \leq \rho(B) + \epsilon$$

**结论**

数值上看，最小的矩阵算子范数就是谱半径！！因而迭代是否收敛取决于迭代矩阵的谱半径是否小于1！！

**Theorem 11**

设$B \in \boldsymbol{R}^{n\times n}$, 则下面命题等价

- 迭代格式
$$\boldsymbol{x}^{(k+1)} = B\boldsymbol{x}^{(k)} + f, \quad k = 0, 1, 2, \ldots$$
对任意初值均收敛.
- $\rho(B) < 1$
- 存在某种$\diamond$范数满足 $\|B\|_\diamond < 1$
- 对任意矩阵范数
$$\lim_{k\to\infty} \|B^k\| = 0$$
即 $B^k$收敛到零矩阵, i.e.,
$$\lim_{k\to\infty} \left(B^k\right)_{ij} = 0, \quad 1 \le i, j \le n$$

如果找到某种范数满足 $\|B\|_\diamond = q < 1$, 则由迭代格式

$$\boldsymbol{\varepsilon}^{(k+1)} = B * \boldsymbol{\varepsilon}^{(k)}$$

可得(忽略下标)

$$\|\boldsymbol{\varepsilon}^{(k+1)}\| = \|B * \boldsymbol{\varepsilon}^{(k)}\| \leq \|B\| * \|\boldsymbol{\varepsilon}^{(k)}\| \to$$

$$\|\boldsymbol{\varepsilon}^{(k+1)}\| \leq \|B\|^{k+1} * \|\boldsymbol{\varepsilon}^{(0)}\| = q^{k+1}\|\boldsymbol{\varepsilon}^{(0)}\|$$

$$\|\boldsymbol{\varepsilon}^{(k+1)} - \boldsymbol{\varepsilon}^{(k)}\| = \|B\boldsymbol{\varepsilon}^{(k)} - \boldsymbol{\varepsilon}^{(k)}\| \geq \|\boldsymbol{\varepsilon}^{(k)}\| - q * \|\boldsymbol{\varepsilon}^{(k)}\|$$

即

$$\|\boldsymbol{x}^{(k+1)} - \boldsymbol{x}^{(k)}\| = \|\boldsymbol{\varepsilon}^{(k+1)} - \boldsymbol{\varepsilon}^{(k)}\| \geq (1-q) * \|\boldsymbol{\varepsilon}^{(k)}\|$$

上式第一个等号：$\boldsymbol{\varepsilon}^{(k)} = \boldsymbol{x}^{(k)} - \boldsymbol{x}^*$

由

$$\|\boldsymbol{\varepsilon}^{(k+1)}\| \leq q^{k+1}\|\boldsymbol{\varepsilon}^{(0)}\|$$

$$\|\boldsymbol{x}^{(k+1)} - \boldsymbol{x}^{(k)}\| \geq (1-q) * \|\boldsymbol{\varepsilon}^{(k)}\|$$

可得

**Theorem 12**

如果存在迭代矩阵的某种范数满足$\|B\| = q < 1$, 则有如下估计

- 先验估计 $\|x^* - x^{(k)}\| \leq q^k\|x^* - x^{(0)}\|$
- 后验估计(第一个小于号) $\|x^* - x^{(k)}\| \leq \dfrac{q}{1-q}\|x^{(k)} - x^{(k-1)}\| \leq \dfrac{q^k}{1-q}\|x^{(1)} - x^{(0)}\|$

**Remark 2.1**

上述估计式并不常用，WHY？实际应用时多用 残差、(残量、residual）来判断数值解的精度

$$r_k = b - Ax^{(k)} = Ax^* - Ax^{(k)} = A * \left(x^* - x^{(k)}\right) = A * \boldsymbol{\varepsilon}^{(k)}$$

# Contents

**问题：**

给定Ax = b, 如何选取迭代矩阵B, 且满足收敛性条件?

不容易！！

$$Ax - b = 0 \Leftrightarrow x = x - C * (Ax - b)$$

即

$$\boxed{x = \Big(I - CA\Big)x + Cb}$$

得到迭代格式

$$x^{(k+1)} = \Big(I - CA\Big)x^{(k)} + Cb, \quad k = 0, 1, 2, \ldots$$

观察上式，迭代矩阵即为$B = I - CA$, 由前文收敛性分析可知，若存在某种范数 使得 $\|I - CA\| = q < 1$, 则迭代格式收敛. 此时也可称C为A的广义逆矩阵. q = 0时，有$CA = I$

**NOTICE:**

矩阵求逆比解方程组的工作量要大很多，线性方程组数值解必须 关心时效问题； 因此，A的广义逆C必须快速求解， 不可使用$C = A^{-1}$。

$$\boxed{\boldsymbol{x} = \Big(I - CA\Big)\boldsymbol{x} + C\boldsymbol{b}}$$

考虑特殊情况，A的对角线元素数量级比该行其它元素都大. 形式上$A \approx D = diag(A)$, 即

$$A = \begin{bmatrix} a_{11} & a_{12} & \dots & a_{1n} \\ a_{21} & a_{22} & \dots & a_{2n} \\ \vdots & \vdots & \ddots & \vdots \\ a_{n1} & a_{n2} & \dots & a_{nn} \end{bmatrix} \approx \begin{bmatrix} a_{11} & 0 & \dots & 0 \\ 0 & a_{22} & \dots & 0 \\ \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & \dots & a_{nn} \end{bmatrix} = D$$

则 $C = D^{-1}$, 进而迭代矩阵

$$B = I - CA = I - D^{-1} * A$$

此时构造迭代格式，称之为Jacobi迭代！！

## Jacobi迭代矩阵形式$B = I - D^{-1} * A$

将系数矩阵写成如下形式$A = D + L + U$

$$A = \begin{bmatrix} a_{11} & 0 & \cdots & 0 \\ 0 & a_{22} & \ddots & \vdots \\ \vdots & \ddots & \ddots & 0 \\ 0 & \cdots & 0 & a_{nn} \end{bmatrix} + \begin{bmatrix} 0 & 0 & \cdots & 0 \\ a_{21} & 0 & \ddots & \vdots \\ \vdots & \ddots & \ddots & 0 \\ a_{n1} & \cdots & a_{nn-1} & 0 \end{bmatrix} + \begin{bmatrix} 0 & a_{12} & \cdots & a_{1n} \\ 0 & 0 & \ddots & \vdots \\ \vdots & \ddots & \ddots & a_{n-1n} \\ 0 & \cdots & 0 & 0 \end{bmatrix}$$

其中D是对角矩阵，L是下三角部分，U是上三角部分. 则

$$B = D^{-1} * D - D^{-1} * (D + L + U) = -D^{-1} * (L + U)$$

于是Jacobi迭代格式 $x = \underset{\sim}{B}\, x + \underline{\underline{f}}$为

$$x^{(k+1)} = \underset{\sim}{-D^{-1}(L+U)} * x^{(k)} + \underline{\underline{D^{-1}b}}$$

或者(计算时用）求解对角方程

$$Dx^{(k+1)} = \left(b - (L+U) * x^{(k)}\right)$$

**Example 13**

计算每步迭代的乘除法次数

$$Dx^{(k+1)} = \left(b - (L+U) * x^{(k)}\right)$$

matlab可快速计算矩阵乘法，应用时需要注意计算效率，不可随意取逆(包括$D^{-1}$)。

矩阵形式简洁，求和形式如下

$$\sum_{j=1}^{n} a_{ij} x_j = b_i, \quad i = 1, 2, ..., n$$

则

$$x_i^{(k+1)} = \frac{1}{a_{ii}} \left[ b_i - \sum_{j<i} a_{ij} * x_j^{(k)} - \sum_{j>i} a_{ij} * x_j^{(k)} \right], \quad i = 1, 2, ..., n$$

若A是稠密矩阵，则每迭代一步需要$n^2$次乘法。

观察Jacobi迭代格式发现，若按照$i = 1, 2, ..., n$顺序计算，当计算第(k+1)步迭代所对应的第i个分量 $x_i^{(k+1)}$时，已经计算出$x_j^{(k+1)}$($j < i$). 为何不用呢?

# Contents

Jacobi迭代格式

$$x_i^{(k+1)} = \frac{1}{a_{ii}} \left[ b_i - \sum_{j<i} a_{ij} * x_j^{(k)} - \sum_{j>i} a_{ij} * x_j^{(k)} \right], \quad i = 1, 2, ..., n$$

及时利用已更新分量，可得Gauss-Seidel迭代

$$x_i^{(k+1)} = \frac{1}{a_{ii}} \left[ b_i - \sum_{j<i} a_{ij} * x_j^{(k+1)} - \sum_{j>i} a_{ij} * x_j^{(k)} \right], \quad i = 1, 2, ..., n$$

比较可知，Gauss-Seidel迭代和Jacobi迭代计算复杂度几乎一样； 但是直观上GS效率优于Jacobi.

**思考：**

根据简单迭代推导公式

$$\boldsymbol{x} = \boldsymbol{x} + C(b - A\boldsymbol{x}) = (I - CA)\boldsymbol{x} + Cb$$

GS迭代的广义逆$C$是什么？

GS迭代的矩阵形式

$$Dx^{(k+1)} = b - Lx^{(k+1)} - Ux^{(k)}$$

$$\Big(D + L\Big)x^{(k+1)} = b - Ux^{(k)}$$

即

$$x^{(k+1)} = \underline{-(D+L)^{-1}U}x^{(k)} + \underline{\underline{(D+L)^{-1}}}b$$

若记G为系数矩阵A的下三角矩阵(连同对角线), 上式即为

$$x^{(k+1)} = -G^{-1}Ux^{(k)} + G^{-1}b = G^{-1}\Big(G - A\Big)x^{(k)} + G^{-1}b$$

也即是此时取A的广义逆C为下三角矩阵G的逆.

**NOTICE:**

上式仅仅用于分析和书写，不用于 实际计算。因为按照下三角矩阵求逆和乘法的计算量是$\boldsymbol{O}(n^3)$。 而按照表达式顺序计算时工作量与Jacobi一致.

Jacobi

$$Dx^{(k+1)} = \Big(b - (L+U) * x^{(k)}\Big)$$

Gauss-Seidel

$$\Big(D + L\Big)x^{(k+1)} = b - Ux^{(k)}$$

**结论：**

1. 计算复杂度一样 都是 $n^2$
2. 计算效果GS更优，符合理论：下三角比对角线包含了矩阵的更多信息

想一想：Jacobi迭代为什么没有淘汰， 仅仅是教学需要？和GS比起来Jacobi 一无是处？没有任何优势？ No !!

验证迭代矩阵的谱半径是否小于1比较麻烦，有兴趣的读者自己试试看。 这里给出一般结论:

- 严格对角占优

$$|a_{ii}| > \sum_{j \geq i,\ j=1}^{n} |a_{ij}|, \quad i = 1, 2, ..., n$$

  此时Jacobi , Gauss-Seidel均收敛.

- A对称正定, 则Gauss-Seidel收敛
- A以及(2D-A)均对称正定，Jacobi收敛

# Contents

回忆几个名词：刘辉割圆术的松弛技术，Romberg 积分，Richardson 外推； 共同特点：加权平均.

## SOR：GS的加权平均

已知第k步迭代值$x^{(k)}$

1. 利用GS迭代计算得到中间值 $\overline{x^{(k+1)}}$
2. 取两步迭代的平均值

$$x^{(k+1)} = (1-w)x^{(k)} + w\overline{x^{(k+1)}} = x^{(k)} + w\left(\overline{x^{(k+1)}} - x^{(k)}\right)$$

需要指出的是：不像梯形公式有具体的收敛阶表达式，可以精确选取 松弛因子w; 若对系数矩阵和迭代矩阵没有足够的分析，好的w只能靠经验和反复试验得到。

$$A = D + L + U$$

SOR迭代的矩阵形式如下:

$$x^{(k+1)} = (1-w)x^{(k)} + w\left[D^{-1} * \left(b - Lx^{(k+1)} - Ux^{(k)}\right)\right]$$

简单推导可得:

$$\begin{aligned} Dx^{(k+1)} &= (1-w)Dx^{(k)} + w\left(b - Lx^{(k+1)} - Ux^{(k)}\right) \\ \Rightarrow (D+wL)x^{(k+1)} &= \left[(1-w)D - wU\right]x^{(k)} + wb \\ \Rightarrow x^{(k+1)} &= \underbrace{(D+wL)^{-1}\left[(1-w)D - wU\right]}x^{(k)} + \underline{\underline{w(D+wL)^{-1}}}b \end{aligned}$$

波浪线部分为迭代矩阵B, 此时系数矩阵A的广义逆

$$C = w(D+wL)^{-1}$$

$w$取值介于(0,2) 之间，以1为界，分别称为<u>低松弛和超松弛</u>

1. 分别利用Jacobi, Gauss-Seidel, SOR迭代法 计算Ax = b,  n = 15；并画出向量x的图像

$$\begin{bmatrix} \frac{5}{2} & -1 & 0 & \cdots & 0 & -1 \\ -1 & \frac{5}{2} & -1 & 0 & \cdots & 0 \\ & \ddots & \ddots & \ddots & & \\ -1 & 0 & \cdots & 0 & -1 & \frac{5}{2} \end{bmatrix}_{n\times n} * \begin{bmatrix} x_1 \\ x_2 \\ \vdots \\ x_n \end{bmatrix} = \begin{bmatrix} 1 \\ 0 \\ \vdots \\ 0 \end{bmatrix}_{n\times 1}$$

依次给出代码

1. Jacobi 迭代
2. Gauss-Seidel 迭代
3. 以GS为代码基础，利用加权平均写 SOR

# Contents

**Example 14**

已知$A$对称正定， 求n元二次函数

$$f(x) = \frac{1}{2}x^T Ax - x^T b, \quad x \in \boldsymbol{R}^n$$

的极小值

多元函数极值的为驻点，先计算梯度(导数)

$$f(x) = \frac{1}{2}\sum_{j=1}^{n}\sum_{k=1}^{n} a_{kj}x_k x_j - \sum_{j=1}^{n} x_j b_j$$

$$\frac{\partial f}{\partial x_l} = \frac{1}{2}\sum_{k=1}^{n} a_{kl}x_k + \frac{1}{2}\sum_{j=1}^{n} a_{lj}x_j - b_l = \sum_{k=1}^{n} a_{kl}x_k - b_l$$

$$\rightarrow \boxed{\nabla f = \text{grad } f = Ax - b}$$

1. $A\boldsymbol{x}-b=0$的解是多元函数驻点

$$f(x)=\frac{1}{2}x^{\mathrm{T}}Ax-x^{\mathrm{T}}b$$

2. 由于$A$正定，则二次函数$f(x)$的极小值点必存在且为驻点

$$f(x^*)=\min f(x) \quad \Leftrightarrow \quad \nabla f(x^*)=Ax^*-b=0$$

**结论和新问题**

1. 线性方程组$Ax=b$, 转化为多元函数的极小值问题
2. 如何找到这个极小值?
3. 任给一点，函数$f(x)$在该点的负梯度方向下降最快！

想象自己在一个环绕的山腰，准备出发到山底！？

**想象自己在一个环绕的山腰，准备出发到山底？**

1. 找好下降方向 $\rightarrow$ 负梯度方向
2. 沿着选定方向直走
3. 走到选定方向的最低点，停下来，拐弯（找新的下降方向）

翻译成数学语言

1. 给定任意初值$x_0$,(人当前所在位置) , 计算残量 $r_0 = b - Ax_0$. (最陡的下降方向)
2. 沿着选定的方向 P前进，例如$P = r_0$方向

$$x_1 = x_0 + \alpha * P, \quad \text{怎样选择合适的}\alpha$$

$\alpha = ?$ （沿着负梯度方向走多远？）当前方向P的最低点！满足

$$\frac{df}{d\alpha}\Big|_{x_1} = \nabla f\Big|_{x=x_1} \cdot P = \Big(r_1, P\Big) = \Big(b - Ax_1, P\Big) = 0$$

$$\rightarrow \Big(b - Ax_0 - \alpha AP, P\Big) = \Big(r_0 - \alpha AP, P\Big) = 0 \rightarrow \alpha = \frac{\Big(r_0, P\Big)}{\Big(AP, P\Big)}$$

3. repeat (环视一周，找好新的前进方向，走到底)

A-(1) 给定任意初值$x_0$, 计算residual $r_0 = b - Ax_0$.

A-(2) 选择 $P = r_0$为前进方向， 计算

$$\alpha = \frac{\left(r_0, r_0\right)}{\left(Ar_0, r_0\right)} \qquad \text{update} \quad x_1 = x_0 + \alpha r_0$$

A-(3) repeat

**思考:**

上述方法下山是否是最优选择?

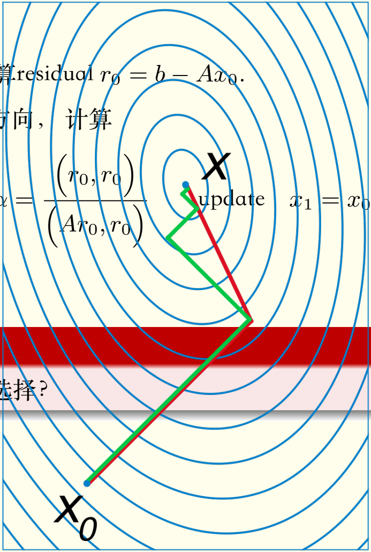
局部最优，非全局最优！

A-(1) 给定任意初值$x_0$, 计算residual $r_0 = b - Ax_0$.

A-(2) 选择 $P = r_0$为前进方向，计算

$$\alpha = \frac{(r_0, r_0)}{(Ar_0, r_0)} \quad \text{update} \quad x_1 = x_0 + \alpha r_0$$

A-(3) repeat




**思考：**

上述方法下山是否是最优选择？

局部最优，非全局最优！

**Definition 15**

A对称正定，u,v是n维向量，则可定义A-内积

$$\Big(u,v\Big)_A=(Au,v)=v^TAu=\sum_{j=1}^{n}\sum_{k=1}^{n}a_{jk}u_jv_k$$

若$\Big(v,u\Big)_A=0$, 则该A内积对应的角度正交，此时称为A共轭。简言之， A-共轭即是A内积下正交！！

**Theorem 16**

给定一组非零向量

$$p_1,p_2,...,p_k,\quad p_j\in\boldsymbol{R}^n,\ \&\ k\le n$$

若

$$(p_i,p_j)_A=0,\ i\neq j$$

则

$$\sum_{l=1}^{k}\alpha_lp_l=0\Rightarrow\alpha_l=0$$

即p彼此线性无关.

对上式依次与$p_i$做A内积即得结论。

如何理解$Ax=b$的解存在唯一?

$$\begin{bmatrix} | & | & \cdots & | \\ a_1 & a_2 & \cdots & a_n \\ | & | & \cdots & | \end{bmatrix} \begin{bmatrix} x_1 \\ \vdots \\ x_n \end{bmatrix} = \begin{bmatrix} | \\ b \\ | \end{bmatrix}$$

1. 矩阵列满秩, 列向量可以选做一组基向量

$$span\{a_1, a_2, ..., a_n\} = \boldsymbol{R}^n$$

2. 解方程等价于: 寻找一组坐标 $x_1, ..., x_n$使得$b$可以由列向量线性表示!!

$$b = x_1 * a_1 + x_2 * a_2 + ... + x_n * a_n$$

若A是正交矩阵或者列向量$a_k$ 彼此正交，则容易计算出系数$x_k$

$$x_k = \Big(a_k, b\Big), \quad k = 1 \to n$$

**思考:**

已知 正交基很有效！如果有一组A共轭的向量组选做一组基，$x_k$ 是不是也容易计算? ?

给定$\boldsymbol{R}^n$ 空间中一组彼此A共轭的向量组（可作为一组基）

$$P_1, P_2, \cdots, P_n$$

则目标向量$x = A^{-1}b$ 可由表示成这组基的线性组合

$$x = \sum_{j=1}^{n} \alpha_j P_j \ \rightarrow \ Ax = b = \sum_{j=1}^{n} \alpha_j AP_j \ \rightarrow \ P_k^T b = \alpha_k P_k^T AP_k$$

即：

$$\alpha_k = \frac{(P_k, b)}{\left(P_k, P_k\right)_A} = \frac{P_k^T b}{P_k^T AP_k}$$

如果给定了一组共轭向量组$P_k$，则可按照上式求出对于系数$\alpha_k$

**主要任务：**

如何有效的给出一组A共轭向量组$P_k$??

CG方法 !!

CG方法即可以顺序给出A共轭向量组

$$P_0, P_1, ..., P_k, ...$$

则得到精确解

$$\boldsymbol{x} = \sum_{j=0}^{n-1} \alpha_j P_j, \quad \alpha_k = \frac{P_k^{\mathrm{T}} b}{P_k^{\mathrm{T}} A P_k}$$

**WHY Iteration ?**

$$\begin{cases} x^{(0)} = \mathbf{0} \quad \text{Or other initial guess} \\ x^{(k+1)} = x^{(k)} + \alpha_k P_k = \alpha_0 P_0 + \alpha_1 P_1 + ... + \alpha_k P_k \end{cases}$$

P 是比最速下降法的$r_k$更有效的前进方向！！不必要$k = n$, 精度满足要求即可！！

理论上是直接法求精确解，实用上是迭代法近似解！一般 $k \ll n$, 也是子空间方法！！

**Example 17**

设$y_0, y_1, ..., y_k \in \boldsymbol{R}^n$是一组线性无关的向量，尝试构造一组A-共轭的向量组

$$P_0, P_1, ..., P_k,$$

满足

$$span\{y_0, y_1, ..., y_k\} = span\{P_0, P_1, ..., P_k\}$$

Gram-Schmidt

$$\begin{cases} P_0 = y_0 \\ P_j = y_j - \displaystyle\sum_{l=0}^{j-1} \frac{(AP_l, y_j)}{(AP_l, P_l)} P_l, \quad j \geq 1 \end{cases}$$

若只有两个向量，则

$$\boxed{P_1 = y_1 - \beta P_0, \quad \beta = \frac{(AP_0, y_1)}{(AP_0, P_0)}}$$

为了书写方便，$\boldsymbol{x}_k$代表第k步的迭代向量，不指向量的第k个分量

## 算法

1. 给定初值$\boldsymbol{x}_k = \mathbf{0}$(若无其它有效信息，初值取零向量)；计算残差（residual）

$$r_0 = b - A\boldsymbol{x}_0$$

2. 取 $P_0 = r_0$, 计算

$$\boldsymbol{x}_1 = \boldsymbol{x}_0 + \alpha_0 P_0, \qquad \alpha_0 = \frac{(r_0, P_0)}{(AP_0, P_0)}$$

3. 由$\boldsymbol{x}_1$得到新的残量$r_1 = b - A\boldsymbol{x}_1$. 以$r_1$为基础，对$P_0$做A投影， 得到新的前进方向$P_1$

$$P_1 = r_1 - \beta_1 P_0, \quad \beta_1 = \frac{(AP_0, r_1)}{(AP_0, P_0)}$$

4. repeat

$$\boxed{\boldsymbol{x}_{k+1} = \boldsymbol{x}_k + \alpha_k P_k} \rightarrow \boxed{r_{k+1} \rightarrow P_{k+1}} \rightarrow \boxed{\boldsymbol{x}_{k+2}}$$

$$
\begin{cases} r_0 = b - A\boldsymbol{x}_0, \quad P_0 = r_0 \\ \boxed{\boldsymbol{x}_{k+1} = \boldsymbol{x}_k + \alpha_k P_k} \to \boxed{r_{k+1} \to P_{k+1}} \end{cases} \Rightarrow \begin{cases} r_0, r_1, ..., r_n \\ P_0, P_1, ..., P_n \end{cases}
$$

## Theorem 18

上述算法产生的向量序列$r_j$，$P_j$满足

$$r_l \perp span\{r_0, r_1, \cdots r_{l-1}\} \tag{1}$$

$$P_l \perp^{\mathrm{A}} span\{P_0, P_1, \cdots P_{l-1}\} \tag{2}$$

$$span\{r_0, r_1, \cdots r_{l-1}\} = span\{P_0, P_1, \cdots P_{l-1}\} = span\{r_0, Ar_0, ..., A^{l-1} r_0\} \tag{3}$$

## 思考：

若$l = n$, 由上定理知残差序列

$$r_0, r_1, ..., r_n$$

彼此正交！(n+1)组n维向量彼此正交，说明什么?

中间必有零向量！即CG方法不超过n步迭代必能得到精确解！

## Theorem 19

$$r_l \perp span\{r_0, r_1, \cdots r_{l-1}\} \tag{4}$$

$$P_l \perp^{A} span\{P_0, P_1, \cdots P_{l-1}\} \tag{5}$$

$$span\{r_0, r_1, \cdots r_{l-1}\} = span\{P_0, P_1, \cdots P_{l-1}\} = span\{r_0, Ar_0, ..., A^{l-1} r_0\} \tag{6}$$

归纳法证明。给定初值为0向量，$P_0 = r_0 = b$. 记号$\perp^{A}$表示A共轭。证明过程会用到以下等式

$$r_{k+1} = r_k - \alpha A P_k \qquad \Leftarrow x_{k+1} = x_k + \alpha P_k \tag{7}$$

$$(r_{k+1}, P_k) = 0 \qquad \Leftarrow \frac{df}{d\alpha} = 0 \tag{8}$$

$$P_k = r_k - \beta P_{k-1} \qquad \Leftarrow P_{k+1} \perp^{A} P_k \tag{9}$$

1. 验证 $r_1 \perp P_0, \quad r_1 \perp r_0, \quad \& \quad P_1 \perp^{A} P_0$
2. 归纳假设 $l \leq k$时，r 相互正交，P 彼此共轭，
$$span\{r_0, r_1, \cdots r_{l-1}\} = span\{r_0, Ar_0, ..., A^{l-1} r_0\}, \quad 0 \leq l \leq k$$
3. 推导目标
$$r_{k+1} \perp r_j, \quad P_{k+1} \perp^{A} P_j, \quad j = 0, ..., k$$
$$span\{r_0, r_1, \cdots r_k\} = span\{r_0, Ar_0, ..., A^{k} r_0\}$$

## $r_{k+1} \perp r_j, \quad j = 0, ..., k$ 的证明

已知指标$i, j \leq k$时, $(r_i, r_j) = 0; \ (AP_i, P_j) = 0$, 即

$$r_l \perp span\{r_0, r_1, \cdots r_{l-1}\}, \quad P_l \perp^{A} span\{P_0, P_1, \cdots P_{l-1}\}, \quad l \leq k$$

且

$$r_{k+1} = r_k - \alpha A P_k \qquad \Leftarrow x_{k+1} = x_k + \alpha P_k \tag{10}$$

$$(r_{k+1}, P_k) = 0 \qquad \Leftarrow \frac{df}{d\alpha} = 0 \tag{11}$$

$$P_k = r_k - \beta P_{k-1} \qquad \Leftarrow P_{k+1} \perp^{A} P_k \tag{12}$$

1. $r_{k+1} \perp r_k$ :

$$(r_{k+1}, r_k) = (r_{k+1}, P_k + \beta P_{k-1}) = \beta(r_{k+1}, P_{k-1}) = \beta(r_k - \alpha A P_k, P_{k-1}) = 0$$

2. $r_{k+1} \perp r_j, \ j \leq k-1$ :

$$(r_{k+1}, r_j) = (r_k - \alpha_k A P_k, r_j) = -\alpha(A P_k, r_j) = -\alpha(A P_k, P_j + \beta P_{j-1}) = 0$$

以上说明(等号因为二者可相互线性表示)

$$r_{k+1} \perp \mathrm{span}\{r_0, r_1, ..., r_k\} = \mathrm{span}\{P_0, P_1, ..., P_k\}$$

# $P_{k+1} \perp^{A} P_j, \quad j = 0, ..., k$ 的证明

已知指标$i, j \le k$时, $(r_i, r_j) = 0; \ (AP_i, P_j) = 0$, 即

$$r_l \perp span\{r_0, r_1, \cdots r_{l-1}\}, \quad P_l \perp^{A} span\{P_0, P_1, \cdots P_{l-1}\}, \quad l \le k$$

且

$$r_{k+1} = r_k - \alpha A P_k \qquad \Leftarrow x_{k+1} = x_k + \alpha P_k \tag{13}$$

$$(r_{k+1}, P_k) = 0 \qquad \Leftarrow \frac{df}{d\alpha} = 0 \tag{14}$$

$$P_k = r_k - \beta P_{k-1} \qquad \Leftarrow P_{k+1} \perp^{A} P_k \tag{15}$$

1. $P_{k+1} \perp^{A} P_k$ : $P_{k+1}$是$r_{k+1}$投影得到，自然成立, (15)
2. $P_{k+1} \perp P_j, \ j \le k-1$ :

$$(P_{k+1}, AP_j) = (r_{k+1} - \beta P_k, AP_j) = \frac{1}{\alpha}(r_{k+1}, r_{j+1} - r_j) = 0$$

以上说明$P_{k+1} \perp^{A} P_l, \ l \ne k$, 则

$$P_{k+1} \perp^{A} \mathrm{span}\{P_0, P_1, ..., P_k\} = \mathrm{span}\{r_0, r_1, ..., r_k\}$$

归纳假设

$$span\{r_0, r_1, \cdots r_{k-1}\} = span\{r_0, Ar_0, ..., A^{k-1} r_0\}$$

且已推导和证明

$$r_{k+1} = r_k - \alpha A P_k \qquad \Leftarrow x_{k+1} = x_k + \alpha P_k$$

$$r_{k+1} \perp \text{span}\{P_0, P_1, ..., P_k\} = \text{span}\{r_0, r_1, ..., r_k\}$$

证明方法：两集合相互包含

1. $\forall \boldsymbol{x} \in \mathcal{K}^k(r_0)$, 有 $\boldsymbol{x} \in span\{r_0, r_1, \cdots r_k\}$

$$\boldsymbol{x} = y + z = \underbrace{y \in \mathcal{K}^{k-1}(r_0)} + c * A^k r_0 = \boxed{\cancel{y}} + A * (\boxed{c * A^{k-1} r_0})$$

记$w = \boxed{c * A^{k-1} r_0}$, 只要说明$A * w \in \text{span}\{r_0, r_1, ..., r_k\}$即可。

$$w \in \mathcal{K}^{k-1}(r_0) = span\{r_0, ..., r_{k-1}\} = span\{P_0, ..., P_{k-1}\} \to w = \sum_{j=0}^{k-1} c_j P_j$$

$$Aw = \sum_{j=0}^{k-1} c_j A P_j \to Aw = \sum_{j=0}^{k} \tilde{c}_j r_j$$

2. $\forall \boldsymbol{x} \in span\{r_0, r_1, \cdots, r_k\}$ 有 $\boldsymbol{x} \in \mathcal{K}^k(r_0)$. (读者自证。)

## 小结

1. 概念：矩阵范数、条件数
2. 迭代法基本原理
3. 简单迭代（不动点迭代）: Jacobi, Gauss-Seidel, SOR
4. 最速(梯度)下降法：按照负梯度方向找函数的极小点，有广泛的应用
5. 共轭梯度法（CG）：
    1. 理论上是直接法求精确解
    2. 实用上是迭代法：多用PCG（预处理共轭梯度法）
    3. 也可看作是Krylov子空间方法(20世纪十大算法之一！)