第四讲 线性方程组迭代方法

4.1 迭代法基本概念

4.2 三种经典迭代方法

为什么迭代法

直接法 VS 迭代法

- 直接法运算量 $\mathcal{O}(n^3)$, 随着矩阵规模的增大, 运算量也随之 **快速增长**
- 对于大规模线性方程组, 特别是稀疏方程组, 当前的首选方法是 **迭代方法**

迭代法基本思想

$$\boxed{Ax = b}, \quad A \in \mathbb{R}^{n\times n} \text{ 非奇异}$$

迭代法基本思想

$$\boxed{Ax = b}, \quad A \in \mathbb{R}^{n\times n} \text{ 非奇异}$$

给定一个初始值 $x^{(0)}$, 通过 **一定的迭代格式** 生成一个迭代序列 $\{x^{(k)}\}_{k=0}^{\infty}$, 使得

$$\boxed{\lim_{k\to\infty} x^{(k)} = x_* \triangleq A^{-1}b}$$

迭代法基本思想

$$\boxed{Ax = b}, \qquad A \in \mathbb{R}^{n\times n} \text{ 非奇异}$$

给定一个初始值 $x^{(0)}$, 通过 **一定的迭代格式** 生成一个迭代序列 $\{x^{(k)}\}_{k=0}^{\infty}$, 使得

$$\boxed{\lim_{k\to\infty} x^{(k)} = x_* \triangleq A^{-1}b}$$

目前常用的两类迭代法

定常迭代法: 如 Jacobi, Gauss-Seidel, SOR 等.

子空间迭代法: 如 CG, MINRES, GMRES, BiCGStab 等.

4-1-1 向量序列与矩阵序列的收敛性

定义 (向量序列的收敛) 设 $\left\{x^{(k)}\right\}_{k=0}^{\infty}$ 是 $\mathbb{R}^n$ 中的一个向量序列. 如果存在向量 $x=[x_1,\, x_2, \ldots,\, x_n]^{\mathsf{T}} \in \mathbb{R}^n$ 使得

$$\lim_{k\to\infty} x_i^{(k)} = x_i, \quad i = 1, 2, \ldots, n,$$

其中 $x_i^{(k)}$ 表示 $x^{(k)}$ 的第 i 个分量. 则称 $\left\{x^{(k)}\right\}$ (按分量) 收敛 到 x, 记为

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

我们称 x 为序列 $\left\{x^{(k)}\right\}$ 的 极限.

矩阵序列的收敛性

定义 (矩阵序列的收敛) 设 $\left\{A^{(k)}=\left[a_{ij}^{(k)}\right]\right\}_{k=0}^{\infty}$ 是 $\mathbb{R}^{n\times n}$ 中的一个矩阵序列. 如果存在矩阵 $A=[a_{ij}]\in\mathbb{R}^{n\times n}$ 使得

$$\lim_{k\to\infty} a_{ij}^{(k)}=a_{ij},\quad i,j=1,2,\ldots,n,$$

则称 $A^{(k)}$ 收敛到 A, 记为

$$\lim_{k\to\infty} A^{(k)}=A.$$

我们称 A 为 $A^{(k)}$ 的 极限.

例 设 $0 < |a| < 1$, 考虑矩阵序列 $\{A^{(k)}\}$, 其中

$$A^{(k)} = \begin{bmatrix} a & 1 \\ 0 & a \end{bmatrix}^k, \quad k = 1, 2, \ldots.$$

易知当 $k \to \infty$ 时, 有

$$A^{(k)} = \begin{bmatrix} a & 1 \\ 0 & a \end{bmatrix}^k = \begin{bmatrix} a^k & ka^{k-1} \\ 0 & a^k \end{bmatrix} \to \begin{bmatrix} 0 & 0 \\ 0 & 0 \end{bmatrix}.$$

收敛性定理

定理 设向量序列 $\{x^{(k)}\}_{k=0}^{\infty} \subset \mathbb{R}^n$, 矩阵序列 $\left\{A^{(k)} = \left[a_{ij}^{(k)}\right]\right\}_{k=0}^{\infty} \subset \mathbb{R}^{n\times n}$, 则

(1) $\boxed{\lim\limits_{k\to\infty} x^{(k)} = x \Longleftrightarrow \lim\limits_{k\to\infty} \|x^{(k)} - x\| = 0}$, 其中 $\|\cdot\|$ 为任一向量范数;

(2) $\boxed{\lim\limits_{k\to\infty} A^{(k)} = A \Longleftrightarrow \lim\limits_{k\to\infty} \|A^{(k)} - A\| = 0}$, 其中 $\|\cdot\|$ 为任一矩阵范数;

(板书)

% 该结论将向量 (矩阵) 序列的收敛性转化为数列的收敛性.

两种特殊情形

$$\lim_{k\to\infty} A^{(k)} = 0 \iff \lim_{k\to\infty} \left\| A^{(k)} \right\| = 0$$

$$\lim_{k\to\infty} A^{k} = 0 \iff \lim_{k\to\infty} \left\| A^{k} \right\| = 0$$

更多判别方法

定理 设矩阵序列 $\left\{A^{(k)} = \left[a_{ij}^{(k)}\right]\right\}_{k=0}^{\infty} \subset \mathbb{R}^{n\times n}$, 则

$$\lim_{k\to\infty} A^{(k)} = 0 \Longleftrightarrow \lim_{k\to\infty} A^{(k)} x = 0, \quad \forall\, x \in \mathbb{R}^n.$$

(板书)

更多判别方法 (续)

定理 设 $B \in \mathbb{R}^{n\times n}$, 若存在矩阵范数使得 $\|B\| < 1$, 则 $\lim\limits_{k\to\infty} B^k = 0$. (板书)

定理 设 $B \in \mathbb{R}^{n\times n}$, 则 $\lim\limits_{k\to\infty} B^k = 0$ 当且仅当 $\rho(B) < 1$. (板书)

推论 设 $B \in \mathbb{R}^{n\times n}$, 则 $\lim\limits_{k\to\infty} B^k = 0$ 的充要条件是存在某个矩阵范数 $\|\cdot\|$, 使得 $\|B\| < 1$.

怎么构造迭代法

定常 VS 非定常

- 定常迭代法: 每一步的迭代格式是不变
- 非定常迭代法: 迭代格式是可变的, 收敛性分析较复杂

定常迭代法

基本思想

直接求解 $Ax = b$ 比较困难, 我们可以求解一个近似线性方程组 $Mx = b$, 其中 M 是 A 在某种意义下的近似.

- 记 $Mx = b$ 的解为 $x^{(1)}$, 与原方程的解 $x_* = A^{-1}b$ 之间的误差满足

$$A\left(x_* - x^{(1)}\right) = b - Ax^{(1)}.$$

定常迭代法

基本思想

直接求解 $Ax = b$ 比较困难, 我们可以求解一个近似线性方程组 $Mx = b$, 其中 M 是 A 在某种意义下的近似.

- 记 $Mx = b$ 的解为 $x^{(1)}$, 与原方程的解 $x_* = A^{-1}b$ 之间的误差满足

$$A\left(x_* - x^{(1)}\right) = b - Ax^{(1)}.$$

如果 $x^{(1)}$ 已经满足精度要求, 则可以停止计算, 否则需要 **修正**.

- 记 $\Delta x \triangleq x_* - x^{(1)}$, 则 Δx 满足方程

$$A\Delta x = b - Ax^{(1)}.$$

由于直接求解比较困难, 因此我们还是求解近似方程

$$\boxed{M\Delta x = b - Ax^{(1)}}$$

得到一个近似的修正量 $\tilde{\Delta} x$. 于是修正后的近似解为

$$x^{(2)} = x^{(1)} + \tilde{\Delta} x = x^{(1)} + M^{-1}(b - Ax^{(1)}).$$

如果 $x^{(2)}$ 已经满足精度要求, 则停止计算, 否则 **继续按以上的方式进行修正**.

- 不断重复以上步骤, 于是, 我们就得到一个向量序列

$$x^{(1)}, x^{(2)}, \ldots, , x^{(k)}, \ldots.$$

它们都是真解 x_* 的近似值, 且满足下面的递推关系

$$\boxed{x^{(k+1)} = x^{(k)} + M^{-1}(b - Ax^{(k)})}, \quad k = 1, 2, \ldots.$$

这就构成了一个迭代方法.

由于每次迭代的格式是一样的, 因此称为 **定常迭代法**.

不断重复以上步骤，于是，我们就得到一个向量序列

$$x^{(1)}, x^{(2)}, \ldots, , x^{(k)}, \ldots .$$

它们都是真解 x_* 的近似值，且满足下面的递推关系

$$\boxed{x^{(k+1)} = x^{(k)} + M^{-1}(b - Ax^{(k)})}, \quad k = 1, 2, \ldots .$$

这就构成了一个迭代方法.

由于每次迭代的格式是一样的，因此称为 **定常迭代法**.

好的定常迭代法需要考虑的两个方面

(1) 以 M 为系数矩阵的线性方程组必须比原线性方程组 **更容易求解**

(2) M 应该是 A 的一个很好的近似：迭代序列 $\{x_k\}$ **快速收敛**

4-1-2 基于矩阵分裂的迭代法

目前一类比较常用的定常迭代法是基于矩阵分裂的迭代法, 如 Jacobi 方法, Gauss-Seidel 方法, SOR (Successive Over-Relaxation, 超松弛) 方法等.

4-1-2 基于矩阵分裂的迭代法

目前一类比较常用的定常迭代法是基于矩阵分裂的迭代法, 如 Jacobi 方法, Gauss-Seidel 方法, SOR (Successive Over-Relaxation, 超松弛) 方法等.

定义 (矩阵分裂 Matrix Splitting) 设 $A \in \mathbb{R}^{n\times n}$ 非奇异, 我们称

$$A = M - N \tag{4.1}$$

为 A 的一个**矩阵分裂**, 其中 M 非奇异.

- 给定一个矩阵分裂, 则原方程组 $Ax = b$ 就等价于 $\boxed{Mx = Nx + b}$.

- 给定一个矩阵分裂, 则原方程组 $Ax = b$ 就等价于 $\boxed{Mx = Nx + b}$.

于是我们就可以构造出以下的迭代格式

$$\boxed{Mx^{(k+1)} = Nx^{(k)} + b}, \quad k = 0, 1, \ldots, \tag{4.2}$$

或

$$\boxed{x^{(k+1)} = M^{-1}Nx^{(k)} + M^{-1}b \triangleq Bx^{(k)} + f}, \quad k = 0, 1, \ldots, \tag{4.3}$$

其中 $B \triangleq M^{-1}N$ 称为 **迭代矩阵**.

给定一个矩阵分裂, 则原方程组 $Ax = b$ 就等价于 $\boxed{Mx = Nx + b}$.

于是我们就可以构造出以下的迭代格式

$$\boxed{Mx^{(k+1)} = Nx^{(k)} + b}, \quad k = 0, 1, \ldots, \tag{4.2}$$

或

$$\boxed{x^{(k+1)} = M^{-1}Nx^{(k)} + M^{-1}b \triangleq Bx^{(k)} + f}, \quad k = 0, 1, \ldots, \tag{4.3}$$

其中 $B \triangleq M^{-1}N$ 称为 **迭代矩阵**.

这就是基于矩阵分裂的迭代方法, 选取不同的 M, 就得到不同的迭代方法.

4-1-3 迭代法的收敛性

定义 对任意初始向量 $x^{(0)}$，设 $\{x^{(k)}\}$ 是由迭代方法 4.3 生成的向量序列，如果 $\lim_{k\to\infty} x^{(k)}$ 存在，则称迭代方法 4.3 **收敛**，否则就称为 **发散**.

4-1-3 迭代法的收敛性

定义 对任意初始向量 $x^{(0)}$，设 $\{x^{(k)}\}$ 是由迭代方法 4.3 生成的向量序列，如果 $\lim\limits_{k\to\infty} x^{(k)}$ 存在，则称迭代方法 4.3 **收敛**，否则就称为 **发散**.

引理 设迭代序列 $\{x^{(k)}\}$ 收敛，且 $\lim\limits_{k\to\infty} x^{(k)} = x_*$，则 x_* 一定是原方程组的真解.

(板书)

定常迭代法基本收敛性定理

定理 对任意初始向量 $x^{(0)}$, 迭代方法 4.3 收敛的充要条件是 $\rho(B) < 1$.

(板书)

充分条件

定理 若存在算子范数使得 $\|B\| < 1$, 则迭代方法 4.3 收敛

充分条件

定理 若存在算子范数使得 $\|B\| < 1$, 则迭代方法 4.3 收敛

注记

- 由于计算 $\rho(B)$ 通常比较复杂, 而 $\|B\|_1$, $\|B\|_\infty$ 相对比较容易计算, 因此在判别迭代方法收敛性时, 可以先验算一下迭代矩阵的 1-范数或 ∞-范数是否小于 1.
- 上述定理中的结论是充分条件, 但不是必要条件, 因此判断一个迭代方法不收敛仍然需要使用基本收敛定理.

例 讨论迭代方法 $x^{(k+1)} = Bx^{(k)} + f$ 的收敛性, 其中 $B = \begin{bmatrix} 0.9 & 0 \\ 0.3 & 0.8 \end{bmatrix}$.

解. 由于 B 是下三角矩阵, 因此其特征值分别为 $\lambda_1 = 0.9$, $\lambda_2 = 0.8$.

所以 $\rho(B) = 0.9 < 1$. 故迭代方法收敛. $\square$

误差估计

定理 若存在算子范数使得 $q \triangleq \|B\| < 1$, 则

(1) $\|x^{(k)} - x_*\| \le q^k \|x^{(0)} - x_*\|$;

(2) $\|x^{(k)} - x_*\| \le \dfrac{q}{1-q} \|x^{(k)} - x^{(k-1)}\|$;

(3) $\|x^{(k)} - x_*\| \le \dfrac{q^k}{1-q} \|x^{(1)} - x^{(0)}\|$.

(板书)

4-1-4 迭代方法的收敛速度

第 k 步的误差为

$$\varepsilon^{(k)} \triangleq x^{(k)} - x_* = B^k(x^{(0)} - x_*) = B^k\varepsilon^{(0)}.$$

所以

$$\frac{\left\|\varepsilon^{(k)}\right\|}{\left\|\varepsilon^{(0)}\right\|} \leq \|B^k\|.$$

因此平均每次迭代后误差的压缩率为 $\|B^k\|^{1/k}$.

定义 基于矩阵分裂的定常迭代法的 **平均收敛速度** 定义为

$$R_k(B) \triangleq -\ln \|B^k\|^{\frac{1}{k}},$$

渐进收敛速度 定义为

$$R(B) \triangleq \lim_{k\to\infty} R_k(B) = -\ln \rho(B).$$

定义 基于矩阵分裂的定常迭代法的 平均收敛速度 定义为

$$R_k(B) \triangleq -\ln \|B^k\|^{\frac{1}{k}},$$

渐进收敛速度 定义为

$$R(B) \triangleq \lim_{k\to\infty} R_k(B) = -\ln \rho(B).$$

- 如果 $0 < \rho(B) < 1$, 则迭代方法 **线性** 收敛.
- 一般来说, $\rho(B)$ 越小, 迭代方法的收敛速度越快.

定义 基于矩阵分裂的定常迭代法的 **平均收敛速度** 定义为

$$R_k(B) \triangleq -\ln \|B^k\|^{\frac{1}{k}},$$

渐进收敛速度 定义为

$$R(B) \triangleq \lim_{k\to\infty} R_k(B) = -\ln \rho(B).$$

- 如果 $0 < \rho(B) < 1$, 则迭代方法 **线性** 收敛.
- 一般来说, $\rho(B)$ 越小, 迭代方法的收敛速度越快.

如果事先给定一个精度要求, 比如要求相对误差满足 $\boxed{\dfrac{\|x^{(k)} - x_*\|}{\|x^{(0)} - x_*\|} < \varepsilon}$,

则可根据下面的公式估计所需迭代步数 k:

$$\|B^k\| < \varepsilon \Longrightarrow \ln \|B^k\|^{1/k} \le \frac{1}{k}\ln(\varepsilon) \Longrightarrow k \ge \frac{-\ln(\varepsilon)}{-\ln \|B^k\|^{1/k}} \approx \frac{-\ln(\varepsilon)}{R(B)}.$$

三种经典定常迭代法

$$\boldsymbol{A} = \boldsymbol{D} - \boldsymbol{L} - \boldsymbol{U}$$

4.2 三种经典迭代方法

4-2-1 Jacobi 迭代法

取 $\boldsymbol{M = D,\ N = L + U}$

- 可得 **Jacobi 迭代** 法:

$$\boxed{x^{(k+1)} = D^{-1}(L+U)x^{(k)} + D^{-1}b}, \quad k = 0, 1, \ldots.$$

4-2-1 Jacobi 迭代法

取 $\boldsymbol{M = D,\ N = L + U}$

- 可得 **Jacobi 迭代** 法:

$$\boxed{x^{(k+1)} = D^{-1}(L+U)x^{(k)} + D^{-1}b}, \quad k = 0, 1, \ldots.$$

- 对应的迭代矩阵

$$\boxed{J = D^{-1}(L+U)}$$

4-2-1 Jacobi 迭代法

取 $\boldsymbol{M = D,\ N = L + U}$

- 可得 **Jacobi 迭代** 法:

$$\boxed{x^{(k+1)} = D^{-1}(L+U)x^{(k)} + D^{-1}b}, \qquad k = 0, 1, \ldots.$$

- 对应的迭代矩阵

$$\boxed{J = D^{-1}(L+U)}$$

- 分量形式 (便于理解与编程实现)

$$\boxed{x_i^{(k+1)} = \frac{1}{a_{ii}}\left(b_i - \sum_{j=1, j\neq i}^{n} a_{ij}x_j^{(k)}\right)}, \qquad i = 1, 2, \ldots, n.$$

算法 4-2.1 Jacobi 迭代

Given an initial guess $x^{(0)}$

while not converge **do** % 停机准则

  for $i = 1$ to n **do**

$$x_i^{(k+1)} = \left(b_i - \sum_{j=1, j\neq i}^{n} a_{ij} x_j^{(k)} \right) \Big/ a_{ii}$$

  end for

end while

几点说明

- Jacobi 迭代中 $x_i^{(k+1)}$ 的更新顺序与 i 无关, 因此非常适合并行计算.
- 在编程实现该算法时, “停机准则” 一般是要求相对残量满足一定的精度, 即

$$\frac{\|b - Ax^{(k)}\|}{\|b - Ax^{(0)}\|} < \text{tol},$$

其中 tol 是一个事前给定的精度要求, 如 10^{-6} 或 10^{-8} 等.

- Jacobi 迭代格式也可以写为

$$x^{(k+1)} = x^{(k)} + D^{-1}(b - Ax^{(k)}) = x^{(k)} + D^{-1} r_k$$

其中 $r_k \triangleq b - Ax^{(k)}$. 这表明, $x^{(k+1)}$ 是通过对 $x^{(k)}$ 做一个修正得到的.

4-2-2 Gauss-Seidel (G-S) 迭代法

取 $M = D - L, \; N = U$

- 可得 **G-S 迭代法**

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

4-2-2 Gauss-Seidel (G-S) 迭代法

取 $\boldsymbol{M = D - L,\ N = U}$

- 可得 **G-S 迭代法**

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

- 对应的迭代矩阵

$$G = (D-L)^{-1} U$$

- 分量形式

$$Dx^{(k+1)} = Lx^{(k+1)} + Ux^{(k)} + b \iff x_i^{(k+1)} = \frac{1}{a_{ii}} \left(b_i - \sum_{j=1}^{i-1} a_{ij} x_j^{(k+1)} - \sum_{j=i+1}^{n} a_{ij} x_j^{(k)} \right)$$

算法 4-2.2 Gauss-Seidel 迭代法

Given an initial guess $x^{(0)}$

while not converge **do**

    for $i = 1$ to n **do**

$$x_i^{(k+1)} = \frac{1}{a_{ii}}\left(b_i - \sum_{j=1}^{i-1} a_{ij}x_j^{(k+1)} - \sum_{j=i+1}^{n} a_{ij}x_j^{(k)}\right)$$

    end for

end while

G-S 迭代的优点

G-S 迭代充分利用了已经获得的最新数据, 有望获得更快的收敛速度.

4-2-3 SOR 迭代法

基本思想

将 G-S 迭代法中的第 $k+1$ 步近似解与第 k 步近似解做一个加权平均, 从而给出一个 更好 的近似解.

4-2-3 SOR 迭代法

基本思想

将 G-S 迭代法中的第 $k+1$ 步近似解与第 k 步近似解做一个加权平均, 从而给出一个 更好 的近似解.

SOR 迭代法

$$x^{(k+1)} = (1-\omega)x^{(k)} + \omega D^{-1}\left(Lx^{(k+1)} + Ux^{(k)} + b\right)$$

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

其中 ω 称为**松弛参数**.

几点说明

- 当 $\omega = 1$ 时, SOR 即为 G-S 方法,

  当 $\omega < 1$ 时, 称为**低松弛**方法,

  当 $\omega > 1$ 时, 称为**超松弛**方法.

  在大多数情况下, 当 $\omega > 1$ 时会取得比较好的收敛效果.

- SOR 方法曾经在很长一段时间内是科学计算中求解线性方程组的首选方法.

SOR 的迭代矩阵

$$S_\omega = (D - \omega L)^{-1}\big((1-\omega)D + \omega U\big)$$

对应的矩阵分裂

$$M = \frac{1}{\omega}D - L, \quad N = \frac{1-\omega}{\omega}D + U$$

分量形式为

$$\begin{aligned} x_i^{(k+1)} &= (1-\omega)x_i^{(k)} + \frac{\omega}{a_{ii}}\left(b_i - \sum_{j=1}^{i-1} a_{ij}x_j^{(k+1)} - \sum_{j=i+1}^{n} a_{ij}x_j^{(k)}\right) \\ &= x_i^{(k)} + \frac{\omega}{a_{ii}}\left(b_i - \sum_{j=1}^{i-1} a_{ij}x_j^{(k+1)} - \sum_{j=i}^{n} a_{ij}x_j^{(k)}\right) \end{aligned}$$

算法 4-2.3 求解线性方程组的 SOR 迭代方法

Given an initial guess $x^{(0)}$ and parameter ω

while not converge **do**

  for $i = 1$ to n **do**

$$x_i^{(k+1)} = (1-\omega)x_i^{(k)} + \frac{\omega}{a_{ii}}\left(b_i - \sum_{j=1}^{i-1} a_{ij}x_j^{(k+1)} - \sum_{j=i+1}^{n} a_{ij}x_j^{(k)}\right)$$

  end for

end while

几点说明

- SOR 方法最大的优点是引入了松弛参数 ω, 增加了算法的自由度, 同时通过选取适当的 ω 可以大大提高方法的收敛速度.
- 如何确定 SOR 的最优松弛因子是一件非常困难的事!

例 分别用 Jacobi, G-S 和 SOR($\omega = 1.1$) 迭代方法求解线性方程组 $Ax = b$, 其中

$$A = \begin{bmatrix} 2 & -1 & 0 \\ -1 & 3 & -1 \\ 0 & -1 & 2 \end{bmatrix}, \quad b = \begin{bmatrix} 1 \\ 8 \\ -5 \end{bmatrix}.$$

初始向量设为 $x^{(0)} = [0, 0, 0]^{\intercal}$, 迭代过程中保留小数点后四位.

(Iter_Jacobi_GS_SOR_01.m)

例 编程实践：分别用 Jacobi, G-S 和 SOR($\omega = 1.5$) 迭代方法求解线性方程组 $Ax = b$, 其中

$$A = \begin{bmatrix} 2 & -1 & & \\ -1 & \ddots & \ddots & \\ & \ddots & \ddots & -1 \\ & & -1 & 2 \end{bmatrix}, \quad b = \begin{bmatrix} 1 \\ 0 \\ \vdots \\ 0 \end{bmatrix}.$$

初始向量设为 $x^{(0)} = [0, 0, 0]^{\mathsf{T}}$.

(Iter_Jacobi_GS_SOR_02.m)