

数值计算方法：原理、算法和应用

Numerical Methods: Principles, Algorithms and Applications

授课教师：周铁

北京大学数学科学学院

2021 年 11 月 15 日

1 线性方程组的迭代数值解法

- 经典迭代法
- 最速下降法
- 共轭梯度法

经典迭代法

经典迭代法

- 直接法需要对系数矩阵进行 LU 分解，一般不能保持原有稀疏性.
- 一些大规模系数矩阵并不需要事先存储，而是需要用到哪个元素时才计算出来.

设要解非奇异 $n \times n$ 线性方程组为

$$Ax = b \quad (1.1)$$

现在考虑如何构造迭代序列 $\{x^{(k)}\}$. 基本思想是：先将系数矩阵 A 分裂成

$$A = M - N, \quad (1.2)$$

其中 M 是一个可逆矩阵，再将方程组(1.1)改写为

$$x = M^{-1}Nx + M^{-1}b. \quad (1.3)$$

利用(1.3)式，可从初值 $x^{(0)}$ 出发产生迭代向量序列 $\{x^{(k)}\}$ ：

$$x^{(k+1)} = M^{-1}Nx^{(k)} + M^{-1}b, \quad k = 0, 1, \dots \quad (1.4)$$

假如迭代格式(1.4)产生的序列 $\{x^{(k)}\}$ 收敛, 则其极限必是方程组(1.3)的解, 进而也是原方程组(1.1)的解.

系数矩阵 A 不同的分裂方式可构造出不同的迭代法. 下面列举其中最基本的三个.

设系数矩阵 $A = (a_{ij})_{n \times n}$ 的对角元素都非零, 将 A 分裂为:

$$A = D - L - U,$$

其中

$$D = \text{diag}(a_{11}, a_{22}, \dots, a_{nn}), \quad a_{ii} \neq 0, \quad i = 1, 2, \dots, n$$

$$L = \begin{bmatrix} 0 & & & \\ -a_{21} & 0 & & \\ \vdots & \ddots & \ddots & \\ -a_{n-1} & \dots & -a_{n,n-1} & 0 \end{bmatrix}, \quad U = \begin{bmatrix} 0 & -a_{12} & \dots & -a_{1n} \\ & 0 & \ddots & \vdots \\ & & \ddots & -a_{n-1,n} \\ & & & 0 \end{bmatrix}.$$

Jacobi 迭代法

在系数矩阵的一般分裂式(1.2)中取 $M = D$ (可逆对角阵), $N = L + U$, 则线性方程组(1.1)化成

$$Dx = (L + U)x + b.$$

这样就得到一种迭代法:

$$x^{(k+1)} = D^{-1}(L + U)x^{(k)} + D^{-1}b, \quad k = 0, 1, 2, \dots, \quad (1.5)$$

称做 Jacobi 迭代方法, 其迭代矩阵为

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

不难写出 Jacobi 迭代法的分量形式:

$$x_i^{(k+1)} = \frac{1}{a_{ii}} \left(b_i - \sum_{j=1}^{i-1} a_{ij} x_j^{(k)} - \sum_{j=i+1}^n a_{ij} x_j^{(k)} \right), \quad i = 1, 2, \dots, n \quad (1.6)$$

Gauss-Seidel 迭代法

在系数矩阵的一般分裂式(1.2)中, 取 $M = D - L$ (可逆下三角阵), $N = U$, 则线性方程组(1.1)化成

$$(D - L)x = Ux + b.$$

这就得到 Gauss-Seidel 迭代方法:

$$x^{(k+1)} = Gx^{(k)} + (D - L)^{-1}b, \quad k = 1, 2, \dots, \quad (1.7)$$

其中的迭代矩阵

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

称为 Gauss-Seidel 迭代矩阵.

Gauss-Seidel 迭代法写成分量形式为:

$$x_i^{(k+1)} = \frac{1}{a_{ii}} \left(b_i - \sum_{j=1}^{i-1} a_{ij} x_j^{(k+1)} - \sum_{j=i+1}^n a_{ij} x_j^{(k)} \right), \quad i = 1, 2, \dots, n \quad (1.8)$$

- Jacobi 方法和 Gauss-Seidel 方法的每个迭代步都只要做矩阵向量乘法，不破坏系数矩阵的稀疏性，计算量都是 $\mathcal{O}(n^2)$ ，如果迭代次数远低于 n ，总计算量就比直接法少.
- 从(1.6)式可以看到：Jacobi 迭代各分量的计算是独立的，这一点非常适合并行计算.
- 从(1.8)式可以看到 Gauss-Seidel 迭代与 Jacobi 迭代的关系：前者就是后者及时更新信息的版本.
- Gauss-Seidel 迭代法的明显的优点是及时更新信息并可以节省存储量，但是各个分量的计算必须按顺序进行，不能同时计算.

SOR (Successive Over-Relaxation) 迭代法

把相邻两次 Gauss-Seidel 迭代的每个分量 $x_i^{(k+1)}$ 和 $x_i^{(k)}$ 做加权平均

$$\begin{aligned}x_i^{(k+1)} &= (1 - \omega)x_i^{(k)} + \omega x_i^{(k+1)}, \quad \omega > 0 \\&= (1 - \omega)x_i^{(k)} + \frac{\omega}{a_{ii}} \left(b_i - \sum_{j=1}^{i-1} a_{ij} x_j^{(k+1)} - \sum_{j=i+1}^n a_{ij} x_j^{(k)} \right) \quad (1.9)\end{aligned}$$

改写为

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

写成矩阵形式，就得到

$$(D - \omega L)x^{(k+1)} = [(1 - \omega)D + \omega U]x^{(k)} + \omega b \quad (1.10)$$

其中的参数 $\omega > 0$ 叫做**松弛因子**.

由(1.10)式, 当 $(D - \omega L)^{-1}$ 存在时, SOR 迭代法(1.10)可以写成

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

其中

$$B_{\omega} = (D - \omega L)^{-1}[(1 - \omega)D + \omega U]$$

就是 SOR 迭代法的迭代矩阵.

SOR 迭代法也可按照(1.4)的方式来构造, 此时只需取

$$M = \omega^{-1}D - L$$

$$N = \omega^{-1}(1 - \omega)D + U$$

- $\omega = 1$ 时就是 Gauss-Seidel 迭代法.

经典迭代法的收敛性

将形如(1.4)的迭代法写成

$$x^{(k+1)} = Sx^{(k)} + c, \quad k = 0, 1, 2, \dots, \quad (1.11)$$

其中 $S = M^{-1}N$ 为迭代矩阵, $c = M^{-1}b$ 称为常数项.

记 $x_* = A^{-1}b$ 是线性方程组 $Mx = Nx + b$ ($Ax = b$) 的准确解, 则

$$e^{(k)} = x^{(k)} - x_*$$

就是误差向量.

迭代法(1.11)产生的序列 $\{x^{(k)}\}$ 收敛于 x_* $\iff \lim_{k \rightarrow \infty} e^{(k)} = 0$.

显然

$$e^{(k+1)} = Se^{(k)}, \quad k = 0, 1, 2, \dots$$

从而有

$$e^{(k)} = S^k e^{(0)}, \quad k = 0, 1, 2, \dots \quad (1.12)$$

由此可见, 要使对任意 $e^{(0)}$ 都有 $e^{(k)} \rightarrow 0$ 的充分与必要条件是 $S^k \rightarrow 0$.

利用矩阵的 Jordan 分解

$$S = T^{-1}JT$$

其中 J 为上三角阵, 且对角线上全是 S 的特征值. 有

$$S^k = T^{-1}J^k T$$

所以要使 $S^k \rightarrow 0$ 只要 S 所有特征值的模严格小于 1, 这等价于 S 的谱半径 $\rho(S) < 1$, 记 $\sigma(A) \subset \mathbb{C}$ 为 A 全体特征值集合, $\rho(S)$ 的定义为:

$$\rho(S) = \max_{\lambda \in \sigma(A)} |\lambda|.$$

迭代法(1.11)的收敛判定准则:

定理

迭代法(1.11)对任意初值 $x^{(0)}$ 都收敛的充分必要条件是 $\rho(S) < 1$.

用谱半径来判定迭代法是否收敛, 显然是不方便的.
因此, 需要给出一些方便的判别条件, 也就是比较容易计算的条件.

利用矩阵分析中谱半径和矩阵算子范数之间的关系，即

$$\rho(A) \leq \|A\|,$$

其中 $\|A\|$ 是方阵 A 的任意一种由向量范数诱导的范数 (又称算子范数)，可得迭代法(1.11)收敛的一个充分条件和误差估计如下：

定理

如果迭代矩阵 S 满足 $\|S\| < 1$ ，则迭代法(1.11)对任意初值都收敛，而且第 k 次迭代向量 $x^{(k)}$ 与精确解 x_* 之间的误差有估计式

$$\|x^{(k)} - x_*\| \leq \frac{\|S\|^k}{1 - \|S\|} \|x^{(1)} - x^{(0)}\|,$$

这里的矩阵范数 $\|S\|$ 是由向量范数 $\|x\|$ 诱导的算子范数。

- 只是一个充分条件，如果迭代矩阵比较容易计算用起来就比较方便 (特别对 $\|S\|_1, \|S\|_\infty$)。
- 迭代矩阵的算子范数越小，迭代法收敛就越快，如果 $\|S\| \approx 1$ 迭代法收敛就很慢。

直接从系数矩阵 A 的性质来推断迭代法的收敛性有以下两个结果.

定理

若 $Ax = b$ 的系数矩阵 $A \in \mathbb{R}^{n,n}$ 是对称正定的, 则

- Jacobi 迭代法收敛的充分必要条件是 $2D - A$ 正定
- Gauss-Seidel 迭代法一定是收敛的
- 当 $0 < \omega < 2$ 时, SOR 迭代法是收敛的

定理

若系数矩阵 $A = (a_{ij})_{n \times n}$ 是按行严格对角占优的, 即

$$|a_{ii}| > \sum_{j=1}^{i-1} |a_{ij}| + \sum_{j=i+1}^n |a_{ij}|, \quad i = 1, 2, \dots, n$$

则 Jacobi 迭代法和 Gauss-Seidel 迭代法都收敛; 当 $0 < \omega < 1$ 时 SOR 迭代法也收敛.

经典迭代法的收敛速率

假设 n 阶迭代矩阵 S 有 n 个线性无关的特征向量 u_1, u_2, \dots, u_n , 并且其特征值满足:

$$1 > |\lambda_1| > |\lambda_2| \geq \dots \geq |\lambda_n|.$$

将初始误差 $e^{(0)} = x^{(0)} - x_*$ 用特征向量 u_1, u_2, \dots, u_n 线性表示为

$$e^{(0)} = \sum_{j=1}^n \alpha_j u_j,$$

则有

$$e^{(k)} = S^k e^{(0)} = \sum_{j=1}^n \alpha_j S^k u_j = \sum_{j=1}^n \alpha_j \lambda_j^k u_j$$

由此可以看出: 随着迭代次数 k 的增加, 误差向量衰减最慢的部分是在 u_1 方向的坐标 $\alpha_1 \lambda_1^k$, 并且有

$$\lim_{k \rightarrow \infty} \frac{\|e^{(k+1)}\|}{\|e^{(k)}\|} = |\lambda_1| = \rho(S) < 1$$

所以经典迭代法是一阶收敛的. 同样是一阶收敛, 迭代矩阵的谱半径不同, 收敛的快慢也有区别. 例如, 要使 $\|e^{(k)}\|$ 缩小为 $\|e^{(0)}\|$ 的 $1/10$, 需要

$$[\rho(S)]^k \leq \frac{1}{10}$$

由此推出

$$k \geq \frac{-1}{\log_{10} \rho(S)} = \frac{1}{-\log_{10} \rho(S)}$$

这说明 $-\log_{10} \rho(S)$ 可以定量刻画经典迭代法的收敛快慢.

定义

经典迭代法

$$x^{(k+1)} = Sx^{(k)} + c, \quad k = 0, 1, 2, \dots,$$

的渐进收敛速率为

$$R(S) = -\log_{10} \rho(S).$$

渐进收敛速率 $R(S)$ 的含义是: 在迭代次数比较大时, 迭代解每取得 1 位十进制有效数字需要迭代的次数.

定理

解方程组 $Ax = b$ 的 SOR 迭代法收敛的必要条件是松弛因子满足 $0 < \omega < 2$.

SOR 迭代矩阵中有一个可以调节的参数 ω . 如果能够找到 ω_{opt} , 使得谱半径 $\rho(B_\omega)$ 达到最小, 就起到加速收敛的作用.

一般来讲最佳松弛参数的选取是相当困难的. 但对某些特殊情况可以得到的最佳松弛参数的表达式. 另外, 在实际应用中, 常常可通过计算过程找到近似的最佳松弛参数.

梯度下降法

Gradient Descend Method

梯度下降法

针对 $n \times n$ **实对称正定** 线性方程组 $Ax = b$ 的求解问题, 定义 n 元二次函数 $\varphi(x)$, $x = (x_1, x_2, \dots, x_n)^T$ 为:

$$\varphi(x) = \frac{1}{2}x^T Ax - b^T x. \quad (2.1)$$

可以证明: 求方程组 $Ax = b$ 的解就等价于求二次函数(2.1)的极小点. 于是求解 $Ax = b$ 的问题就转化为求二次函数 $\varphi(x)$ 的极小点的问题. 求二次函数极小点可以采用如下的下降算法:

- ① 选取向量 $x^{(0)}$ 和下降方向 $p^{(0)}$, 解一维优化问题 $\min_{t>0} \varphi(x^{(0)} + tp^{(0)})$ 得到 t_0 , 然后令 $x^{(1)} = x^{(0)} + t_0 p^{(0)}$.
- ② 选择下降方向 $p^{(1)}$, 解一维优化问题 $\min_{t>0} \varphi(x^{(1)} + tp^{(1)})$ 得到 t_1 , 然后令 $x^{(2)} = x^{(1)} + t_1 p^{(1)}$.

\vdots

如此不断执行下去就得到两个序列

$$t_0, t_1, t_2, \dots \quad \text{和} \quad p^{(0)}, p^{(1)}, p^{(2)}, \dots,$$

我们称 $p^{(k)}$ 为搜索方向, t_k 为搜索步长. 在迭代过程中先在 $x^{(k)}$ 点处选择下降方向 $p^{(k)}$, 再在直线 $x = x^{(k)} + tp^{(k)}$ 上确定步长 t_k 使

$$t_k = \arg \min_{t \geq 0} \varphi(x^{(k)} + tp^{(k)})$$

$$x^{(k+1)} = x^{(k)} + t_k p^{(k)}$$

选择不同的搜索方向 $p^{(k)}$ 和步长 t_k , 就给出各种不同的算法.

先考虑如何确定步长 t_k . 设从 $x^{(k)}$ 出发, 已经选定方向 $p^{(k)}$. 现在要在直线 $x = x^{(k)} + tp^{(k)}$ 上确定 t_k 使得 $\varphi(x)$ 在 $x^{(k+1)} = x^{(k)} + t_k p^{(k)}$ 达到最小.

$$\begin{aligned} f(t) &= \varphi(x^{(k)} + tp^{(k)}) \\ &= \frac{1}{2}(x^{(k)} + tp^{(k)})^T A(x^{(k)} + tp^{(k)}) - b^T(x^{(k)} + tp^{(k)}) \\ &= \frac{1}{2}t^2 (p^{(k)})^T A p^{(k)} - t(r^{(k)})^T p^{(k)} + \varphi(x^{(k)}), \end{aligned}$$

其中 $r^{(k)} = b - Ax^{(k)}$.

由微分学理论，方程

$$f'(t) = t(p^{(k)})^T A p^{(k)} - (r^{(k)})^T p^{(k)} = 0$$

所确定的 t 即为所求的步长 t_k ，即

$$t_k = \frac{(r^{(k)})^T p^{(k)}}{(p^{(k)})^T A p^{(k)}}, \quad t_k > 0 \iff (r^{(k)})^T p^{(k)} > 0. \quad (2.2)$$

事实上，对任意 $t > 0$ ，我们有：

$$\begin{aligned} \varphi(x^{(k)} + t p^{(k)}) - \varphi(x^{(k)}) &= \frac{1}{2} t^2 (p^{(k)})^T A p^{(k)} - t (r^{(k)})^T p^{(k)} \\ &= t \left[\frac{1}{2} t (p^{(k)})^T A p^{(k)} - (r^{(k)})^T p^{(k)} \right] \end{aligned}$$

因此，只要

$$(r^{(k)})^T p^{(k)} > 0, \quad 0 < t < 2(r^{(k)})^T p^{(k)} / (p^{(k)})^T A p^{(k)}$$

就有 $\varphi(x^{(k)} + t p^{(k)})$ 就会下降.

再考虑如何选择下降方向 $p^{(k)}$.

因为 $\varphi(x)$ 增加最快的方向是其梯度方向, 所以负梯度方向是 $\varphi(x)$ 减小最快的方向, 于是最直接的做法是选取 $p^{(k)} = -\nabla\varphi(x^{(k)}) = r^{(k)}$.

解实对称正定线性方程组的最速下降法:

给一个初值 $x^{(0)}$

$r^{(0)} = b - Ax^{(0)}$; $k = 0$

while $r^{(k)} \neq 0$

$k = k + 1$

$t_{k-1} = (r^{(k-1)})^T r^{(k-1)} / (r^{(k-1)})^T A r^{(k-1)}$

$x^{(k)} = x^{(k-1)} + t_{k-1} r^{(k-1)}$

$r^{(k)} = b - Ax^{(k)}$

end

定理

设 $0 < \lambda_1 \leq \lambda_2 \leq \cdots \leq \lambda_n$ 为正定矩阵 A 的特征值, $\|x\|_A = \sqrt{x^T A x}$, 则有

$$\|x^{(k)} - x_*\|_A \leq \left(\frac{\lambda_n - \lambda_1}{\lambda_n + \lambda_1} \right)^k \|x^{(0)} - x_*\|_A$$

共轭梯度法

Conjugate Gradient Method

共轭梯度法 (Conjugate Gradient Method)

对最速下降法做仔细分析就会发现：负梯度方向虽从局部来看是最快下降方向，但从整体来看并非最佳. 这就促使人们去寻求更好的下降方向，当然还希望每步确定下降方向所付出的计算代价不要太大. 共轭梯度法就是根据这一思想设计的，其具体计算过程如下：

给定初值 $x^{(0)}$ ，第一步选负梯度方向为下降方向 $p^{(0)} = r^{(0)} = b - Ax^{(0)}$ ，于是有

$$t_0 = \frac{(r^{(0)})^T p^{(0)}}{(p^{(0)})^T A p^{(0)}}, \quad x^{(1)} = x^{(0)} + t_0 p^{(0)}, \quad r^{(1)} = b - Ax^{(1)}.$$

对以后各步，例如第 k 步 ($k \geq 1$)，下降方向就不再取 $r^{(k)}$ ，而是在过点 $x^{(k)}$ 由向量 $r^{(k)}$ 和 $p^{(k-1)}$ 所张成的超平面

$$\pi_2 = \left\{ x = x^{(k)} + \xi r^{(k)} + \eta p^{(k-1)} : \xi, \eta \in \mathbb{R} \right\}$$

内找出使函数 $\varphi(x)$ 下降最快的方向作为新的下降方向 $p^{(k)}$.

考虑 $\varphi(x)$ 在 π_2 上的限制:

$$\begin{aligned}\psi(\xi, \eta) &= \varphi(x^{(k)} + \xi r^{(k)} + \eta p^{(k-1)}) \\ &= \frac{1}{2}(x^{(k)} + \xi r^{(k)} + \eta p^{(k-1)})^T A(x^{(k)} + \xi r^{(k)} + \eta p^{(k-1)}) - \\ &\quad - b^T(x^{(k)} + \xi r^{(k)} + \eta p^{(k-1)})\end{aligned}$$

直接计算可得

$$\begin{aligned}\frac{\partial \psi}{\partial \xi} &= \xi(r^{(k)})^T A r^{(k)} + \eta(r^{(k)})^T A p^{(k-1)} - (r^{(k)})^T r^{(k)}, \\ \frac{\partial \psi}{\partial \eta} &= \xi(r^{(k)})^T A p^{(k-1)} + \eta(p^{(k-1)})^T A p^{(k-1)},\end{aligned}$$

其中第二个式用到了 $(r^{(k)})^T p^{(k-1)} = 0$, 这可由 $r^{(k)}$ 的定义直接验证.
令

$$\frac{\partial \psi}{\partial \xi} = \frac{\partial \psi}{\partial \eta} = 0,$$

即知 φ 在 π_2 内有惟一的极小点

$$\tilde{x} = x^{(k)} + \xi_0 r^{(k)} + \eta_0 p^{(k-1)},$$

其中 ξ_0 和 η_0 满足

$$\begin{cases} \xi_0(r^{(k)})^T A r^{(k)} + \eta_0(r^{(k)})^T A p^{(k-1)} = (r^{(k)})^T r^{(k)} \\ \xi_0(r^{(k)})^T A p^{(k-1)} + \eta_0(p^{(k-1)})^T A p^{(k-1)} = 0 \end{cases} \quad (3.1)$$

从上式知道, 如果 $r^{(k)} \neq 0$ 必有 $\xi_0 \neq 0$, 因此可取

$$p^{(k)} = \frac{1}{\xi_0}(\tilde{x} - x^{(k)}) = r^{(k)} + \frac{\eta_0}{\xi_0}p^{(k-1)} \quad (3.2)$$

作为新的下降方向. 显然, 这是在平面 π_2 内最佳的下降方向. 令 $\beta_{k-1} = \eta_0/\xi_0$, 则由(3.1)的第二个方程得

$$\beta_{k-1} = -\frac{(r^{(k)})^T A p^{(k-1)}}{(p^{(k-1)})^T A p^{(k-1)}}. \quad (3.3)$$

从(3.2)式看到, 第 k 步迭代的下降方向是当前残差向量与前一次迭代的下降方向的线性组合, 组合系数为 β_{k-1} .

我们先不用(3.3)式，而是令：

$$p^{(k)} = r^{(k)} + \beta p^{(k-1)},$$

再用条件

$$(p^{(k-1)})^T A(r^{(k)} + \beta p^{(k-1)}) = 0 \quad (3.4)$$

来确定组合系数 β . 显然，这样得到的 β_{k-1} 与(3.3)式完全相同.

A 是一个 n 阶实对称正定矩阵，我们可以在 \mathbb{R}^n 中定义另一个内积

$$(x, y)_A = x^T A y, \quad \forall x, y \in \mathbb{R}^n$$

不妨称之为 “ A - 内积” . 在 A - 内积下正交的向量就称为 “共轭” .

等式(3.4)已经表明 $p^{(k)}$ 与 $p^{(k-1)}$ 是共轭的.

实际上还可以证明：

$$(p^{(k)})^T A p^{(j)} = 0, \quad j = 0, 1, \dots, k$$

即 $p^{(k)}$ 与它前 $k-1$ 个下降方向都是共轭的. 共轭梯度法的名称来源于此！

一旦 $p^{(k)}$ 确定, t_k 的计算仍用前面的公式(2.2).

总结上面的讨论, 可得如下的计算公式:

$$t_k = \frac{(r^{(k)})^T p^{(k)}}{(p^{(k)})^T A p^{(k)}} \quad (3.5)$$

$$x^{(k+1)} = x^{(k)} + t_k p^{(k)}$$

$$r^{(k+1)} = b - A x^{(k+1)}$$

$$\beta_k = -\frac{(r^{(k+1)})^T A p^{(k)}}{(p^{(k)})^T A p^{(k)}} \quad (3.6)$$

$$p^{(k+1)} = r^{(k+1)} + \beta_k p^{(k)}$$

这就是著名的共轭梯度法 (CG) 的基本迭代格式. 每迭代一步要产生三个向量 $x^{(k)}$, $r^{(k)}$, $p^{(k)}$, 分别称为**近似解向量**, **残差向量**和**下降方向向量**, 它们具有如下性质:

- 下降方向相互共轭, 即 $(p^{(i)})^T A p^{(j)} = 0, i \neq j$.
- 残差向量相互正交, 即 $(r^{(i)})^T r^{(j)} = 0, i \neq j$.
- 下降方向与残差向量相互正交, 即 $(p^{(j)})^T r^{(k)} = 0, 0 \leq j < k$.
- 下降方向与残差向量是子空间 $K_k(A, r_0)$ 的两组基向量, 即

$$\begin{aligned} K_k(A, r_0) &= \text{span}\{r^{(0)}, Ar^{(0)}, \dots, A^{k-1}r^{(0)}\} \\ &= \text{span}\{r^{(0)}, r^{(1)}, \dots, r^{(k-1)}\} \\ &= \text{span}\{p^{(0)}, p^{(1)}, \dots, p^{(k-1)}\} \end{aligned}$$

- 近似解向量是方程组 $Ax = b$ 之解在超平面 $x^{(0)} + K_k(A, r^{(0)})$ 上的最佳逼近, 即

$$\|x^{(k)} - x_*\|_A = \min\{\|x - x_*\|_A : x \in x^{(0)} + K_k(A, r^{(0)})\},$$

利用上述性质, 又可以导出

$$t_k = \frac{(r^{(k)})^T r^{(k)}}{(p^{(k)})^T A p^{(k)}}, \quad \beta_k = \frac{(r^{(k+1)})^T r^{(k+1)}}{(r^{(k)})^T r^{(k)}}.$$

这两个公式可使 CG 方法的运算减少.

综合上面的讨论, 可得解实对称正定线性方程组的共轭梯度法:

给一个初值 $x^{(0)}$, 计算 $r^{(0)} = b - Ax^{(0)}$, $p^{(0)} = r^{(0)}$, 对 $k = 0, 1, 2, \dots$,

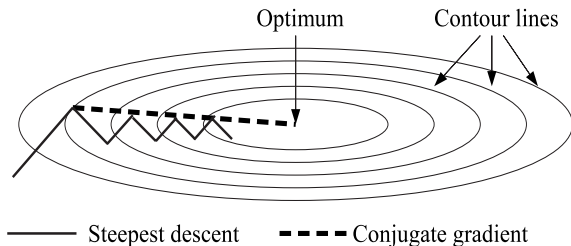
$$\begin{aligned}t_k &= \frac{(r^{(k)})^T r^{(k)}}{(p^{(k)})^T A p^{(k)}} \\x^{(k+1)} &= x^{(k)} + t_k p^{(k)} \\r^{(k+1)} &= r^{(k)} - t_k A p^{(k)} \\\beta_k &= \frac{(r^{(k+1)})^T r^{(k+1)}}{(r^{(k)})^T r^{(k)}} \\p^{(k+1)} &= r^{(k+1)} + \beta_k p^{(k)}\end{aligned}$$

注意共轭梯度法 (CG 算法) 每迭代一次的主要工作量就是计算一次系数矩阵 A 乘列向量 $p^{(k)}$, 其计算量为 $O(n^2)$.

定理

设实对称正定 $n \times n$ 线性方程组的系数矩阵为 A . 如果计算没有误差, 则 CG 方法最多迭代 n 步就得到准确解. 每步迭代的误差 $e^{(k)} = x^{(k)} - x_*$ 都与以前各步的下降方向 $p^{(0)}, p^{(1)}, \dots, p^{(k-1)}$ 正交, 并且有

$$\|e^{(k)}\|_A \leq 2 \left(\frac{\sqrt{\kappa_2} - 1}{\sqrt{\kappa_2} + 1} \right)^k \|e^{(0)}\|_A, \quad \kappa_2 = \|A\|_2 \|A^{-1}\|_2$$



from google image

- 从共轭梯度法的基本性质可以看出, 在没有误差的情况下, 最多 n 步就可得到线性方程组的精确解, 因此理论上共轭梯度法是一种直接方法.
- 在实际使用时, 希望 $k \ll n$ 时, $x^{(k)}$ 就是 x_* 的一个很好的近似. 因此共轭梯度法通常是作为一种迭代法来使用的.
- 理论分析和实际使用的经验表明, 共轭梯度法收敛的快慢与系数矩阵 A 的特征值分布有关. 当 A 的特征值凝聚于少数几个数附近或其条件数不大时, 共轭梯度法收敛的很快. 而当 A 的特征值散布在一个很长的区间上时, 收敛很慢.
- 在使用共轭梯度法之前, 先对原方程作一些适当的预处理使其系数矩阵的谱有某种凝聚性出现, 则可大大地加快共轭梯度法的收敛速度, 这就是预优 (预条件, preconditioning) 共轭梯度法所讨论的问题.

- ① Gene H. Golub, Charles F. Van Loan,
Matrix Computations (4th Edition),
Johns Hopkins University Press, 2012.
- ② James W. Demmel,
Applied Numerical Linear Algebra,
SIAM, 1997.
- ③ Alfio Quarteroni, Riccardo Sacco, Fausto Saleri,
Numerical Mathematics(2nd Edition),
Springer, 2007.