

# 对称逐步超松弛预处理共轭梯度法的改进迭代格式<sup>\*</sup>

林绍忠

(长江科学院, 武汉)

## IMPROVED ITERATIVE FORMAT OF SYMMETRIC SUCCESSIVE OVER RELAXATION—PRECONDITIONED CONJUGATED GRADIENT METHOD

Lin Shao-zhong

(Yangtze River Scientific Research Institute, Wuhan)

### Abstract

The symmetric successive over relaxation-preconditioned conjugated gradient method (SSOR-PCG) is a very efficient iterative method for solving large sparse linear equations. In this paper an improved iterative format of the SSOR-PCG method is presented, which avoids the product operation of coefficient matrix and direction vector and thus saves computation work about 8%—50%.

### § 1. 引言

线性方程组的求解方法可分为两大类: 直接法和迭代法. 对于大型问题, 当系数矩阵为条件数较小的稀疏矩阵且右端项不多时, 迭代法的求解效率高. 尽管迭代法多种多样, 但其迭代收敛速度毫不例外地取决于迭代矩阵的条件数, 而预处理的唯一目的就是降低迭代矩阵的条件数, 从而达到减少迭代次数和计算量的目的. 共轭梯度法 (CG 法) 具有许多内在的优点, 如有限步收敛性质. 在实际计算中, 由于舍入误差的影响, 特别是由于系数矩阵的条件数常常较大, CG 法往往出现收敛慢的问题. 预处理共轭梯度法 (PCG 法) 就是在共轭梯度法中采用了预处理技术. 预处理矩阵不同, 便构成不同的预处理共轭梯度法. 最重要的预处理共轭梯度法是不完全因子分解预处理共轭梯度法 (ICCG 法) 和对称逐步超松弛预处理共轭梯度法 (SSOR-PCG 法). 至今, 预处理共轭梯度法已成为求解大型稀疏矩阵的极为有效的方法.

SSOR-PCG 法在每步迭代中, 需计算系数矩阵与方向向量的乘积. 本文提出的改进迭代格式, 避免了该乘积的计算, 比原迭代格式可节省计算量 8%—50%.

<sup>\*</sup> 1996年5月13日收到.

## § 2. SSOR-PCG 法及改进迭代格式

### 2.1. SSOR-PCG 法

设线性方程组

$$Ax = b \quad (1)$$

的系数矩阵为  $n$  阶对称正定矩阵. 设  $M = (S^T S)^{-1}$  为对称正定矩阵, 则 (1) 等价于

$$A'x' = b', \quad A' = SAS^T, \quad b' = Sb, \quad x' = S^{-T}x. \quad (2)$$

用 CG 法求解 (2), 可不必先求  $x'$ , 而是直接迭代求  $x$ . 其迭代格式为 [1, 2]

$$\left\{ \begin{array}{l} \text{置初值 } x^0, g^0 = Ax^0 - b, h^0 = M^{-1}g^0, d^0 = -h^0, k = 0, \\ R: \quad \delta = (g^k, h^k) \\ \text{如果 } \delta \leq \varepsilon, \text{ 则停止, 否则} \\ \tau_k = (g^k, h^k)/(d^k, Ad^k), \\ x^{k+1} = x^k + \tau_k d^k, \\ g^{k+1} = g^k + \tau_k Ad^k, \\ h^{k+1} = M^{-1}g^{k+1}, \\ \beta_k = (g^{k+1}, h^{k+1})/(g^k, h^k), \\ d^{k+1} = -h^{k+1} + \beta_k d^k, \\ k = k + 1, \\ \text{转到 } R. \end{array} \right. \quad (3)$$

等价问题 (2) 的收敛速度取决于  $\text{Cond}(A')$ . 若  $M$  为单位矩阵, 则 (3) 为原问题的 CG 法迭代格式, 其收敛速度取决于  $\text{Cond}(A)$ , 每步迭代的主要计算量为  $(r_a + 5)n$  次乘法运算. 这里  $r_a$  为  $A$  的各行非零元素个数的平均值. 通常  $\text{Cond}(A)$  相当大, 因此希望  $\text{Cond}(A') \ll \text{Cond}(A)$ . 如此选择的  $M$  称为  $A$  的预处理矩阵. 好的预处理矩阵要满足如下条件:

- (a)  $\text{Cond}(A') \ll \text{Cond}(A)$ ;
- (b) 相对于  $A$ , 不要求过多内存;
- (c) 解方程组  $Mh = g$  较解  $Ax = b$  远为容易.

当  $M$  取为对称逐步超松弛迭代法 (SSOR 法) 的分裂矩阵时, 即

$$M = (2 - \omega)^{-1}(D/\omega + L)(D/\omega)^{-1}(D/\omega + L)^T \quad (4)$$

可以满足上述条件, 则对 (2) 实施 CG 算法称为 SSOR-PCG 法.

式 (4) 中,  $D$  为  $A$  的对角阵,  $L$  为  $A$  的严格下三角矩阵, 即  $A = D + L + L^T$ ,  $0 < \omega < 2$  为松弛因子.

在 SSOR-PCG 法中, 计算  $h = M^{-1}g$  (即求解  $Mh = g$ ) 时, 相当于三角分解法中的前代与回代过程, 或者等价于用 SSOR 法迭代一次. 因此实际计算时, 不必对  $M$  求逆, 也不必另开辟数组存放  $M$  矩阵. 计算  $Ad$  和求解  $Mh = g$  仅对  $A$  的非零元素进行运算, 所以仅需存储  $A$  的下三角阵 (或上三角阵) 中的非零元素. 若  $h$  和  $d$  放大同一倍数, 迭代格式 (3) 仍然适用. 因此,  $M$  中的  $(2 - \omega)$  可不参加计算.

SSOR-PCG 法每步迭代的主要计算量为  $(2r_a + 6)n$  次乘法运算, 其中,  $(r_a + 1)n$  次用于计算  $h = M^{-1}g$ ,  $r_a n$  次用于计算  $Ad$ .

## 2.2. 改进的迭代格式

令

$$W = D/\omega + L, V = (2 - \omega)D/\omega, y = W^{-1}g, z = W^T d. \quad (5)$$

则

$$\left\{ \begin{array}{l} A = W + W^T - V, Ad = Wd + z - Vd, W^{-1}Ad = d + W^{-1}(z - Vd), \\ (d, Ad) = (d, 2z - Vd), (g, h) = (y, Vy), \\ g^{k+1} = g^k + \tau_k Ad^k \Leftrightarrow y^{k+1} = y^k + \tau_k (d^k + W^{-1}(z^k - Vd^k)), \\ d^{k+1} = -h^{k+1} + \beta_k d^k \Leftrightarrow z^{k+1} = -Vy^{k+1} + \beta_k z^k. \end{array} \right. \quad (6)$$

于是, 迭代格式 (3) 可改为

$$\left\{ \begin{array}{l} \text{置初值 } x^0, g^0 = Ax^0 - b, y^0 = W^{-1}g^0, z^0 = -Vy^0, d^0 = W^{-T}z^0, k = 0, \\ R: \delta = (y^k, Vy^k) \\ \text{如果 } \delta \leq \varepsilon, \text{ 则停止, 否则} \\ \tau_k = (y^k, Vy^k)/(d^k, 2z^k - Vd^k), \\ x^{k+1} = x^k + \tau_k d^k, \\ y^{k+1} = y^k + \tau_k (d^k + W^{-1}(z^k - Vd^k)), \\ \beta_k = (y^{k+1}, Vy^{k+1})/(y^k, Vy^k), \\ z^{k+1} = -Vy^{k+1} + \beta_k z^k, \\ d^{k+1} = W^{-T}z^{k+1}, \\ k = k + 1, \\ \text{转到 } R. \end{array} \right. \quad (7)$$

从迭代格式 (7) 可见, 矩阵与向量的乘积运算为  $W^{-1}(z - Vd)$ ,  $W^{-T}z$ ,  $Vy$  及  $Vd$ . 前 2 个为求解三角形方程组, 计算量为  $(r_a + 1)n$  次乘法运算. 由于  $V$  是对角阵, 计算  $Vy$  和  $Vd$  为  $2n$  次乘法运算. 迭代格式 (7) 由于避免了计算  $Ad$ , 每步迭代的主要计算量为  $(r_a + 8)n$  次乘法运算, 比原迭代格 (3) 可节省  $(r_a - 2)n$  次乘法运算. 因此,  $r_a = 3$  时, 改进的迭代格式可节省计算量约 8%;  $2r_a \gg 6$  时, 可节省计算量约 50%, 各迭代步的计算量接近于 CG 法的计算量.

以上  $D$  为对角阵, 即  $M$  是点 SSOR 的分裂矩阵. 若取  $D$  为  $A$  的块对角阵,  $L$  为  $A$  的与  $D$  相应划分的严格块下三角阵, 则  $M$  为块 SSOR 的分裂矩阵, 迭代格式 (7) 仍然适用, 但所能节省的计算量与块的划分及子方程组的解法有关.

### § 3. 算 例

本文提出的改进迭代格式已用于求解固体力学有限元方程组. 这里举一实例说明方法的效率和精度.

在某实际工程结构的三维有限元应力分析中, 共划分空间 8 结点等参单元 1957 个, 结点 2597 个, 方程组的阶数为 7176, 在 16M 内存的 586/90 微机上进行计算. 用直接法 (三角分解法) 求解方程组时, 经带宽优化后, 按一维半带宽存储的系数矩阵容量为 3452728 个 (按单精度存储和计算, 加上其它内存开销, 几乎用到了全部内存, 但没有动用硬盘). 用 SSOR-PCG 法求解时, 仅存储下三角系数矩阵中的非零元素, 其个数为 236912 个, 由于非零元素的分布没有规律, 还用了一个同等大小的索引矩阵指示非零元素的列号, 取  $x^0 = 0$ , 以  $(y^k, Vy^k)/(y^0, Vy^0) \leq \varepsilon$  作为收敛控制条件. 两种方法求解方程组所用时间及计算结果的比较列于表 1. 由于舍入误差的影响, 两种方法的计算结果都存在误差. 为此, 表中还列出余差  $r = b - Ax$  的范数  $\|r\|_2$  ( $\|b\|_2 = 14323.5$ ).

从算例可见, 无论是存储量, 还是计算时间, SSOR-PCG 法都比直接法少. 在表中控制误差  $\varepsilon$  的取值情况下, SSOR-PCG 法的误差也小于直接法. 从表中还可以看出, 松弛因子  $\omega$  对收敛速度有一定的影响, 因此最佳松弛因子的选择是值得探讨的.

表 1 SSOR-PCG 法与直接法的比较

方法及参数		迭代次数	计算时间 (秒)	$\max  x_i  (10^{-3})$	$\ r\ _2$	
S	$\varepsilon = 10^{-8}$	$\omega = 0.5$	207	38	2.7124	13.8
		$\omega = 1.0$	158	29	2.7124	11.9
		$\omega = 1.5$	201	36	2.7124	14.7
R	$\varepsilon = 10^{-12}$	$\omega = 0.5$	301	55	2.7125	13.9
		$\omega = 1.0$	202	37	2.7125	10.9
		$\omega = 1.5$	292	53	2.7125	13.0
P	$\varepsilon = 10^{-16}$	$\omega = 0.5$	364	67	2.7125	13.8
$\omega = 1.0$		270	49	2.7125	11.0	
$\omega = 1.5$		349	64	2.7125	12.7	
直 接 法			645	2.6939	29.5	

#### §4. 结 束 语

SSOR-PCG 法是一种求解大型稀疏线性方程组的快速迭代解法, 而且所需内存少. 本文改进了 SSOR-PCG 法的迭代格式, 比原迭代格式可节省计算量 8%—50%. 笔者已将改进的迭代格式用于求解大型三维固体力学有限元方程组, 比三角分解法一般快几倍至十几倍.

#### 参 考 文 献

- [1] 吕涛, 石济民, 林振宝, 区域分解算法—偏微分方程数值解法新技术, 北京, 科学出版社, 1992, 5.
- [2] 张丽君, 金绥更, 向量算法与并行算法, 北京, 国防工业出版社, 1993, 10.