

自顶向下聚集型代数多重网格预条件的健壮性与参数敏感性研究^{*}

吴建平

(国防科技大学 海洋科学与工程研究院, 长沙 410073)

摘要: 针对自顶向下聚集型代数多重网格预条件, 首先对问题规模敏感性进行了研究, 并与基于强连接的经典聚集型算法进行了系统比较, 发现大部分情况下, 该算法具有明显优势, 特别是在采用 Jacobi 光滑时优势更显著; 之后, 对最粗网格层的分割数与每次每个子图进行分割时的分割数这两个参数进行了敏感性分析。综合分析表明, 自顶向下聚集型代数多重网格预条件具有较好的健壮性, 特别是在采用 Gauss-Seidel 光滑, 或采用九点差分离散时, 健壮性表现更加充分。

关键词: 稀疏线性代数方程组; 聚集型代数多重网格; 预条件; 克里洛夫子空间方法; 图分割

中图分类号: TP301 **文献标志码:** A **文章编号:** 1001-3695(2018)09-2617-04

doi: 10.3969/j.issn.1001-3695.2018.09.013

Research of robustness and parameter-sensitivity of aggregation based algebraic multigrid preconditioner from top to bottom

Wu Jianping

(Academy of Ocean Science & Engineering, National University of Defense Technology, Changsha 410073, China)

Abstract: For the aggregation based algebraic multigrid preconditioner from top to bottom, this paper investigated the sensitivity to the scale of the problem first and compared to the classical aggregation schemes based on strong connections systematically. The results show that this algorithm is superior in the most cases. When Jacobi smoothing is used, the privileges is more significant. Then it investigated the sensitivity to two parameters, including the number of partitions in the coarsest level and that for each sub-graph. The systematic analyses show that the aggregation based algebraic multigrid preconditioner from top to bottom is robust and when Gauss-Seidel smoothing is used or the nine-point difference scheme is used, the robustness is more sufficient.

Key words: sparse linear system; aggregation based algebraic multigrid; preconditioner; Krylov subspace method; graph partitioning

0 引言

稀疏线性代数方程组广泛用于科学与工程计算领域, 且其求解通常在总执行时间中占有很大比重。为求解这种稀疏线性方程组, Krylov 子空间迭代由于具有相对较快的收敛速度而经常采用, 但其收敛速度由系数矩阵的特征值分布决定。为改善这种分布, 只要预条件算子作用到向量上的操作相对于矩阵向量乘操作而言, 计算量比较小, 就可以采用预条件技术^[1,2]。

一般而言, 任何近似求解方法都可以用来构造预条件子。多重网格方法可以求解稀疏线性方程组, 且具有潜在的最优收敛速度。因此, 也可以用做预条件子。多重网格方法的高效性由两个过程, 即光滑与校正的互补性所决定, 这两个过程分别用于减小具有相对较高与相对较低频率的误差分量。在基于聚集的代数多重网格算法中^[3], 当采用经典的简单网格传递算子时, 校正过程由网格层次结构所决定。因此, 当给定光滑过程时, 多重网格的效率只取决于聚集方案本身。

聚集构造最经典的是基于强连接的方案^[4], 每次将最多两个节点聚集在一起。为减少多重网格的层次数量, 可以重复应用多次进行聚集, 来确定最终的聚集^[5,6]。文献[7]中提出了一种基于因子3的网格点粗化算法, 文献[8]中提出了一种

基于强耦合的方案, 其中每个孤立点都尽可能快地加入到相邻的聚集中。Chen 等人^[9]对多种聚集算法进行了比较, 发现能与各向异性相一致的方法最有效。基于这种考虑, 其建议按对应于最小特征值的特征向量, 或者与迭代矩阵谱半径相对应的量来进行聚集。Braess^[10]给出了两种聚集算法, 一种是纯粹基于已知的七种结构来选取网格点进行聚集, 另一种是先对节点按边权分组, 再按组间连接边数多少来进行聚集。为减少聚集构造时间, Deng 等人^[11]对九点有限差分方法, 给出了简单而经济的聚集实现算法。

图分割也可以用于构建聚集, Wu 等人^[12]基于图分割给出了一种并行不完全分解, 并给出了一种两层网格校正型预条件, 用来求解混凝土试件细观力学模拟中所遇到的稀疏线性方程组。Kumar^[13]基于图分割提出了一种两层网格算法, 采用 ILU(0) 作为光滑过程, 并采用 ILUT 作为粗网格上的求解器。实验表明, 这种方法相对于那些基于强连接的方法而言, 具有更好的健壮性。Wu 等人^[14]基于图分割, 给出了多重网格预条件的一种自顶向下型聚集方案, 以利用全局信息来构建多重网格算法中的网格, 网格层次按自顶向下的方式自然形成。实验结果表明, 其相对于经典强连接聚集算法一般具有优势。

本文在文献[14]的基础上, 研究自顶向下聚集型多重网格预条件的健壮性与参数敏感性, 对该预条件子的有效性进行

系统验证。

1 自顶向下聚集型多重网格预条件

考虑稀疏线性方程组:

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

其中: A 是事先给定的 n 阶稀疏矩阵; b 是一个给定向量, 且 x 是未知解向量。为求解这种稀疏线性方程组, 可以采用 Krylov 子空间迭代法^[1,2]。不妨设 A 对称正定, 则可以采用共轭斜量 (CG) 方法进行求解, 但收敛速度依赖于系数矩阵 A 的特征值分布。这些特征值分布的区域相对越窄, 则收敛速度也越快。为改进收敛速度, 可以采用预条件技术, 此时将线性方程组转换为

$$MAx = Mb \quad (2)$$

其中: M 称为预条件子。之后, 可以用 CG 法来求解式 (2), 在应用中将原来的欧几里德范数替换为能量范数 $(\cdot, \cdot)_M$, 该范数定义为 $(x, y)_M = (x, My)$ 。所得到的迭代称为预条件共轭斜量 (PCG) 迭代, 具体细节参见文献 [1] 中的算法 9.1。

任何求解式 (1) 的近似方法都可以作为预条件子, 聚集型代数多重网格可以用来求解式 (1), 因此, 也可以选为预条件子 M 。算法的描述与一般的多重网格算法并无区别, 细节可以参见文献 [2] 中的算法 11.1。在该算法中, 第 0 层是最粗网格层, 第 l 层是第 $l+1$ 层的粗网格。为描述方便起见, 这里对网格层次进行倒序排列。因此, 第 0 层是最细的网格层, 对应于原始的线性方程组。本文采用 V 循环的多重网格版本, 在前后光滑时采用不同的光滑算子, 并将其分别表示为 S_l^R 与 S_l^L 。假设从第 l 层到第 $l+1$ 层的限制算子为 $R_{l+1,l}$, 且从第 $l+1$ 层到第 l 层的延拓算子为 $P_{l,l+1}$ 。算法按嵌套方式进行调用, 将误差向量限制到粗网格, 再采用该算法在粗网格上进行求解, 之后将解插值回细网格对其上的解进行校正。如果网格层数达到事先给定的门槛, 或者线性方程组的规模足够小, 则嵌套过程终止。最粗网格层上的求解可以采用任何方法进行。为确保多重网格算法可以用做 PCG 的预条件子, M 应当对称, 为此, 可以选取 $R_{l+1,l} = P_{l,l+1}^T$ 且 $S_l^L = (S_l^R)^T$ 。

在基于聚集的多重网格方法中, 粗网格从对细网格层中的网格点进行聚集得到, 本文采用文献 [14] 中提出的自顶向下型聚集算法。聚集算法与光滑算子选好后, 还需要确定延拓算子。这里只考虑经典的最简版本, 如果在第 l 层有 n_l 个点, 则延拓算子 $P_{l,l+1}$ 是一个 $n_l \times n_{l+1}$ 的矩阵。对第 l 层, 如果第 j 个聚集含有 i , 则算子 $P_{l,l+1}$ 在第 i 行第 j 列位置上的元素为 1, 所有不等于 1 的元素都为 0。

在自顶向下聚集算法^[14]中, 从最粗网格层开始, 按照逐步到最细网格层的顺序进行网格构建。考虑线性方程式 (1), 将矩阵 A 的邻接图表示为 G 。由于所考虑的矩阵 A 对称, 所以 G 是一个无向图。现在, 将邻接图 G 分割为 m_0 个子图 $G_i^{(1)}$ ($i = 1, 2, \dots, m_0$), 其中 m_0 事先给定。现在, 每个子图 $G_i^{(1)}$ 中的节点可以聚集为第 1 层中的一个节点, 因此, 在这一层上有 m_0 个节点。此时 $\cup \{G_i^{(1)} : i = 1, 2, \dots, m_0\} = G$, 且 $G_i^{(1)} \cap G_j^{(1)} = \emptyset, i, j = 1, 2, \dots, m_0$ 。

对每个 $G_i^{(1)}$, 又可以再次分割为 p 个子图。由于在第 1 层共有 m_0 个部分, 所以, 总共可以得到 pm_0 个子图, 每一个对应于第 2 层中的一个点。分割过程可以继续按嵌套的方式进行, 直到某第 l 层上某个子图足够小为止。第 l 层上每个子图都是原始邻接图中相应节点的一个聚集。网格构建的具体过程可以描述如图 1 所示, 其中 $p = 2, m_0 = 3$ 且 $l = 3$ 。

将 $G_i^{(j)}$ 的顶点集表示为 $V_i^{(j)}$, 则

$$V_{i+(k-1)p}^{(j+1)} \subset V_k^{(j)} \quad i = 1, 2, \dots, p$$

这意味着网格层次结构可以很自然地生成。在第 1 层共有 m_0 个节点, 对第 j ($2 \leq j \leq l-1$) 层, 共有 $m_0 p^{j-1}$ 个节点, 每个节点对应于第 $j+1$ 层中的 p 个节点。对第 l 层, $G_i^{(l)}$ 中含有 $|G_i^{(l)}|$ 个节点, 每个 $G_i^{(l)}$ 含有 G 中部分点。

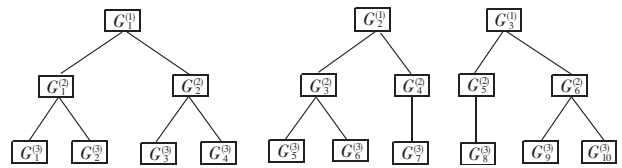


图 1 自顶向下聚集算法的示意图

当将每个图分割为子图时, 可以要求在每个子图中的节点数尽量相同的情况下, 来最小化子图之间的总连接边数。通过这种方式, 当给定参数 m_0 与 p 时, 层数可以最少。另一方面, 子图之间的边数可以衡量连接性。最小化意味着子图之间的连接性最小, 具有强连接的边移到了子图的内部, 这与经典聚集算法中的强连接概念具有相似性, 但必须注意到, 这种方法并不是以节点之间的连接性这样一种局部信息为基础, 而是以子图之间的连接性这样一种全局信息为基础。在进行分割时, 另外一个问题是对分割算法的选取, 显然可以采用任意一种分割算法。由于本文的线性方程式 (1) 来自于模型偏微分方程, 相应的坐标信息已知, 所以本文采用代价低廉的坐标分割法^[2]。

2 自顶向下聚集型代数多重网格预条件的健壮性与参数敏感性研究

本文所有实验都在一个 Intel Xeon CPU E5-2670 0@2.60 GHz (cache 20480 KB) 处理器上得到, 操作系统为 Red Hat Linux 2.6.32-279-aftms-TH, 而所用编译器为 Intel FORTRAN Version 11.1。由于所得到的线性方程组都是对称正定的, 所以总是采用 PCG 迭代。初始迭代向量选为全 0 向量, 并在当前残向量的 Euclid 范数与初始残向量的 Euclid 范数之比小于 $1E-10$ 时, 迭代终止。对基于聚集的多重网格方法, 最粗层上的线性方程组利用无填充的不完全 LU 分解预条件的 PCG 迭代进行求解, 且终止条件与外迭代相同。

实验针对六个稀疏线性方程组进行, 包括 Lin51、Lin52、Lin53、Lin54、Lin91 与 Lin92。所有线性方程组都采用有限差分, 离散二维偏微分方程 Dirichlet 边值问题得到:

$$-a_1 \partial(\rho \partial u / \partial x) / \partial x - a_2 \partial(\rho \partial u / \partial y) / \partial y + \delta u = f$$

定义域是 $(0, c) \times (0, c)$ 。函数 f 与边界条件从真解 $u = 1$ 得到。在每个维度上有 $n+2$ 个点, 且对任意连续函数 u , 将值 $u(x_i, y_j)$ 定义为 $u_{i,j}$, 其中 x_i 与 y_j 分别沿 x 与 y 方向上一致分布。

对线性方程组 Lin51, $a_1 = a_2 = \rho = 1, \delta = 0$ 且采用下述离散形式:

$$-u_{i,j-1} - u_{i-1,j} + 4u_{i,j} - u_{i+1,j} - u_{i,j+1} = h^2 f_{i,j}$$

对线性方程组 Lin52, $a_1 = a_2 = 1$, 且采用的离散形式为

$$-\rho_{i,j-1/2} u_{i,j-1} - \rho_{i-1/2,j} u_{i-1,j} + \lambda_{i,j} u_{i,j} - \rho_{i+1/2,j} u_{i+1,j} - \rho_{i,j+1/2} u_{i,j+1} = h^2 f_{i,j}$$

其中:

$$\lambda_{i,j} = h^2 \delta_{i,j} + \rho_{i,j-1/2} + \rho_{i-1/2,j} + \rho_{i+1/2,j} + \rho_{i,j+1/2}$$

当 (x, y) 处于 R_k ($k = 1, 2, 3$) 中时, $\rho = \rho_k$ 且 $\delta = \delta_k$:

$$R_1 = \{(2, 2, 1) \times (1, 2, 1)\} \cup \{(2, 2, 1) \times (1, 2, 1)\}, R_2 = (1, 2) \times (1, 2)$$

$$R_3 = \{[0, 2, 1] \times [0, 1]\} \cup \{[0, 1] \times [0, 2, 1]\}$$

$$\rho_1 = 1, \rho_2 = 2 \times 10^3, \rho_3 = 3 \times 10^5, \delta_1 = 0.02, \delta_2 = 3, \delta_3 = 500$$

对线性方程组 Lin53, 离散形式与 Lin52 相同, 但当 (x, y) 处于 $R_k (k = 1, 2, 3)$ 中时, $\rho = x\rho_k$ 且 $\delta = x\delta_k$ 。

对线性方程组 Lin54, $\rho = 1, \delta = 0, a_1 = 1, a_2 = 100$, 且所采用的离散形式为

$$-100u_{i,j-1} - u_{i-1,j} + 202u_{i,j} - u_{i+1,j} - 100u_{i,j+1} = h^2 f_{i,j}$$

对稀疏线性方程组 Lin91 与 Lin92, $a_1 = a_2 = \rho = 1, \delta = 0$ 且所采用的离散形式分别为

$$-u_{i-1,j-1} - u_{i,j-1} - u_{i+1,j-1} - u_{i-1,j} + 8u_{i,j} - u_{i+1,j} - u_{i-1,j+1} - u_{i,j+1} - u_{i+1,j+1} = h^2 f_{i,j}$$

与

$$-u_{i-1,j-1} - 4u_{i,j-1} - u_{i+1,j-1} - 4u_{i-1,j} + 20u_{i,j} - 4u_{i+1,j} - u_{i-1,j+1} - 4u_{i,j+1} - u_{i+1,j+1} = h^2 f_{i,j}$$

首先来研究自顶向下聚集型代数多重网格预条件子对问题规模变化的敏感性, 并与基于强连接的聚集型代数多重网格预条件进行比较, 采用 Jacobi 光滑与 Gauss-Seidel 光滑时, 求解各个线性方程组所需要的迭代次数分别如图 2 ~ 13 所示, 其中自顶向下聚集算法中参数 m_0 选为 100, 基于强连接的聚集算法中当某层上的节点个数不大于 100 时终止聚集过程, 且该层即为最粗网格层。图中 strg 表示基于强连接的聚集方案, 后缀表示每次进行聚集时的点数; coor 表示基于坐标信息进行分割的新方法, 后缀表示在进行分割时对每个子图所采用的分割数。

由图 2 ~ 7 可见, 在采用 Jacobi 迭代进行光滑时, 当每次进行聚集的点数与进行子图分割时的分割数相等时, 自顶向下聚集算法总是优于强连接聚集算法, 由于每次迭代所用时间对两种聚集算法而言几乎相同, 所以从迭代时间来看, 自顶向下聚集算法优势也必然很明显。此外, 正如文献[14]所言, 自顶向下聚集算法的设置时间比强连接算法要长, 但测试结果表明, 对本文待求解的线性方程组而言, 设置时间在求解线性方程组的总时间中所占比重很小, 因此迭代次数的优劣也基本上反映了总求解时间的优劣。

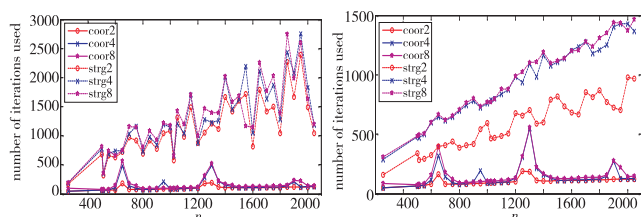


图2 采用Jacobi光滑算子时求解Lin51所需要的迭代次数

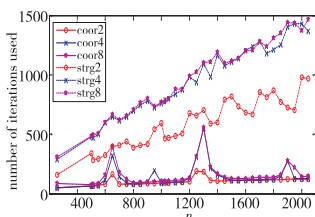


图3 采用Jacobi光滑算子时求解Lin52所需要的迭代次数

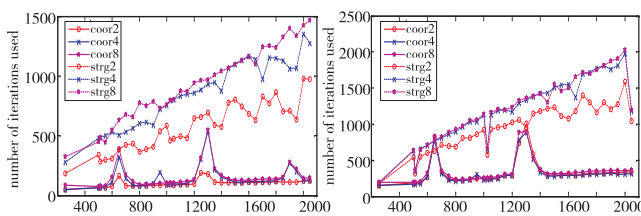


图4 采用Jacobi光滑算子时求解Lin53所需要的迭代次数

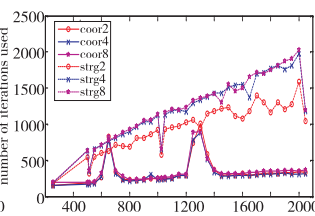


图5 采用Jacobi光滑算子时求解Lin54所需要的迭代次数

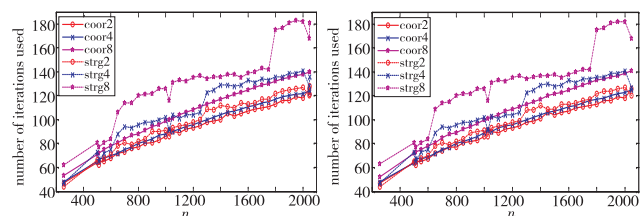


图6 采用Jacobi光滑算子时求解Lin91所需要的迭代次数

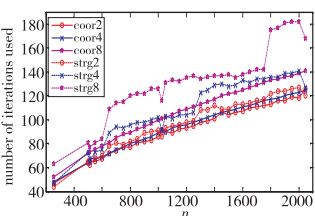


图7 采用Jacobi光滑算子时求解Lin92所需要的迭代次数

从图 2 ~ 7 还可以看出, 基于强连接的聚集算法随着待求解问题规模的变化, 所需要的求解次数波动相对比较剧烈, 当 n 等于 2 的幂次时, 相对于相邻的 n 而言, 迭代次数一般会出现一定程度的减少。对自顶向下型聚集算法, 除了对采用五点差分离散所得的线性方程组, 在 $n = 650, 950, 1300, 1900$ 附近出现了较大波动外, 在其他情形下所需要的迭代次数非常稳定, 这充分说明自顶向下聚集型代数多重网格预条件具有相对较好的健壮性。此外, 对采用五点差分离散得到的线性方程组, 自顶向下聚集算法的优势更显著, 迭代次数随问题规模的增长也远没有强连接性聚集算法那样迅速。如表 1 所示的列表数据从迭代次数与迭代时间上对结论进行了进一步印证。

表1 采用 Jacobi 光滑算子时求解 Lin51 所需迭代时间/s

n	coor2	strg2	coor4	strg4	coor8	strg8
256	0.810	3.209	0.702	2.427	1.073	2.193
500	4.391	47.81	3.610	39.03	3.528	37.03
512	4.487	23.80	4.095	19.00	3.756	17.26
550	4.954	55.34	4.402	42.35	4.638	40.92
600	5.541	62.37	7.794	53.61	10.11	43.12
650	20.83	84.99	40.15	62.66	42.26	57.72
700	9.868	133.1	13.63	101.6	15.56	102.5
750	11.06	143.1	10.76	131.1	13.12	117.5
800	11.71	123.4	9.547	93.33	14.31	93.56
850	13.52	185.3	10.87	140.6	16.12	140.3
900	14.65	174.22	13.33	137.3	17.99	134.6
950	16.02	262.9	33.39	216.7	19.60	197.4
1000	23.00	304.6	19.95	229.3	21.79	213.0
1024	25.23	173.4	22.34	141.3	23.45	127.7

从图 8 ~ 13 可见, 当采用 Gauss-Seidel 光滑时, 对沿不同坐标轴具有强各向异性的问题 Lin54, 自顶向下聚集算法所需要的迭代次数多于基于强连接的聚集算法。对其他稀疏线性方程组, 包括具有间断系数的问题 Lin52、Lin53, 自顶向下聚集算法都具有明显优势。同时, 虽然此时两类算法所需要的迭代次数随问题规模的增长, 相对于采用 Jacobi 光滑而言, 变化都更为平缓, 但此时自顶向下聚集算法所需要迭代次数随问题规模的变化也比基于强连接算法更为平缓, 表明其健壮性依然更好。

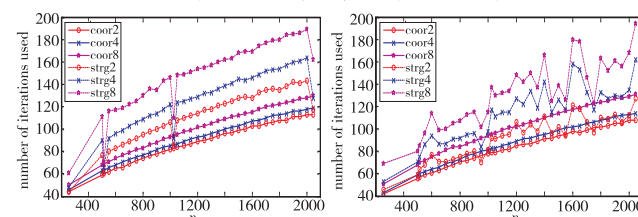


图8 采用Gauss-Seidel光滑算子时求解Lin51所需要的迭代次数

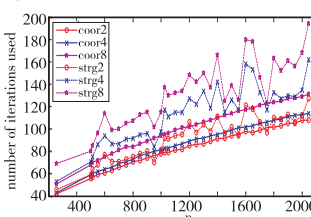


图9 采用Gauss-Seidel光滑算子时求解Lin52所需要的迭代次数

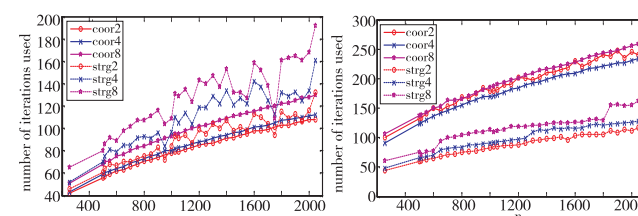


图10 采用Gauss-Seidel光滑算子时求解Lin53所需要的迭代次数

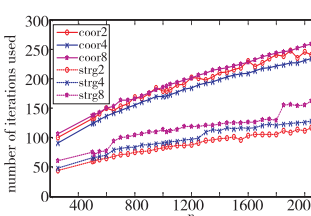


图11 采用Gauss-Seidel光滑算子时求解Lin54所需要的迭代次数

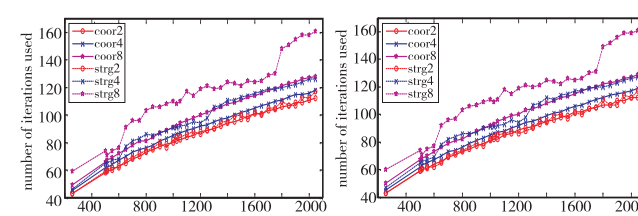


图12 采用Gauss-Seidel光滑算子时求解Lin91所需要的迭代次数

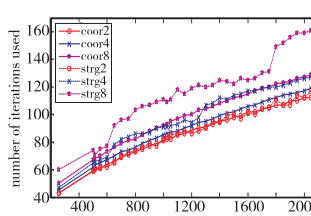


图13 采用Gauss-Seidel光滑算子时求解Lin92所需要的迭代次数

现在来看自顶向下聚集型代数多重网格预条件对最粗层网格上点数选取的敏感性,图 14~19 给出了 $n=1024$ 时迭代次数随 m_0 的变化情况。图中对每个线性方程组,柱状图中自左至右依次给出了当 m_0 取 10~90(间距 10)与 100~900(间距为 100)时所需要的迭代次数。当采用 Jacobi 光滑时,对从五点差分离散得到的稀疏线性方程组,所需要的迭代次数对参数 m_0 比较敏感。由于当 p 固定时,对所有这四个从五点差分离散得到的线性方程组,都具有完全类似的结果,所以,很可能是算法在与离散方式相结合时,存在的固有问题,具体原因有待未来进行进一步的研究。对从九点差分离散得到的稀疏线性方程组,或者在采用 Gauss-Seidel 光滑时,所需迭代次数随 m_0 的增加稳定地减少。

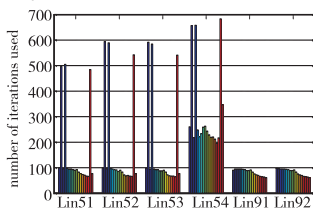


图 14 采用 Jacobi 光滑且 $p=2$ 时迭代次数随 m_0 的变化情况

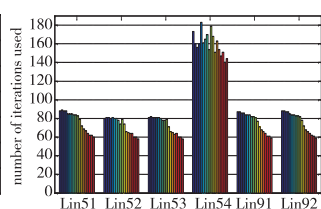


图 15 采用 Gauss-Seidel 光滑且 $p=2$ 时迭代次数随 m_0 的变化情况

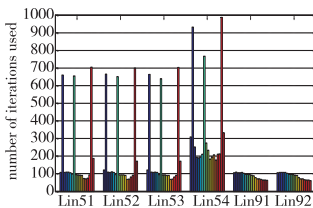


图 16 采用 Jacobi 光滑且 $p=4$ 时迭代次数随 m_0 的变化情况

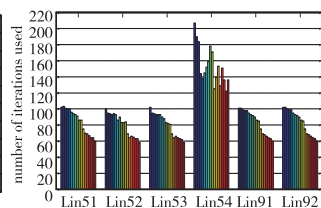


图 17 采用 Gauss-Seidel 光滑且 $p=4$ 时迭代次数随 m_0 的变化情况

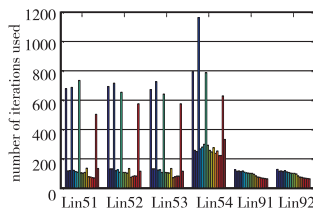


图 18 采用 Jacobi 光滑且 $p=8$ 时迭代次数随 m_0 的变化情况

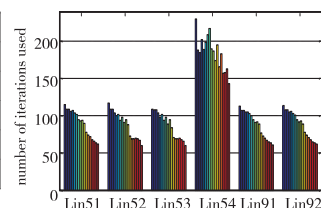


图 19 采用 Gauss-Seidel 光滑且 $p=8$ 时迭代次数随 m_0 的变化情况

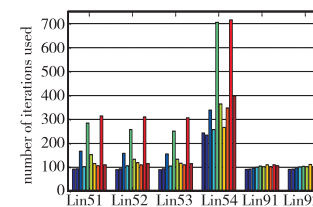


图 20 在 $m_0=100$ 时所需要的迭代次数随 p 的变化情况 (Jacobi 光滑)

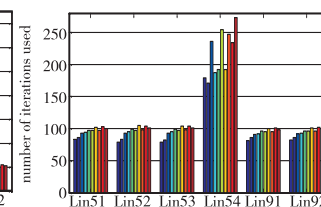


图 21 在 $m_0=100$ 时所需要的迭代次数随 p 的变化情况 (Gauss-Seidel 光滑)

最后来看自顶向下聚集型代数多重网格预条件对子图分割时分割数选取的敏感性,图 20、21 给出了在 $m_0=100$ 与 $n=1024$ 时迭代次数随 p 的变化情况。由图可见,当采用 Jacobi 光滑时,对从五点差分离散得到的稀疏线性方程组,所需要的迭代次数对参数 p 比较敏感,在 p 分别取 6、10 与 18 时,所需迭代次数远大于其他情形。这同样很可能是算法在与离散方式相结合时,存在的固有问题,具体原因有待进一步研究。对从九点差分离散得到的稀疏线性方程组,或者在采用 Gauss-Seidel 光滑时,所需迭代次数随 p 的增加具有逐渐增加的趋势,但波动幅度不大。同时,可以看到,一般 p 取 2、4、8 时所需要的迭代次数相对比较少。

3 结 束 语

本文对自顶向下聚集型代数多重网格预条件的健壮性与参数敏感性进行了研究,结果表明,除对沿坐标轴具有强各向异性的问题离散所得稀疏线性方程组,在采用 Gauss-Seidel 光滑时,该算法的有效性不如经典强连接聚集算法之外,对其他情形,该算法均优于经典强连接聚集算法。同时,该算法随问题规模的变化,求解所需要的迭代次数变化更为平缓,具有更好的健壮性。此外,对利用五点差分离散得到的稀疏线性方程组,当采用 Jacobi 光滑时,该算法所需要的迭代次数对最粗层网格中的网格点数与每次对子图进行分割时的分割数比较敏感,而当切换为 Gauss-Seidel 光滑时,敏感性大大降低,健壮性更好。对采用九点差分离散得到的稀疏线性方程组,算法具有更弱的参数敏感性与更强的健壮性。

参考文献:

- [1] Saad Y. Iterative methods for sparse linear systems [M]. Boston: PWS Pub. Co., 1996.
- [2] 吴建平,王正华,李晓梅. 稀疏线性方程组的高效求解与并行计算 [M]. 长沙: 湖南科学技术出版社, 2004.
- [3] Notay Y. Aggregation-based algebraic multilevel preconditioning [J]. SIAM Journal on Matrix Analysis and Applications, 2006, 27 (4): 998-1018.
- [4] Kim H, Xu Jinchao, Zikatanov L. A multigrid method based on graph matching for convection-diffusion equations [J]. Numerical Linear Algebra Applications, 2003, 10 (1-2): 181-195.
- [5] Notay Y. Aggregation-based algebraic multigrid for convection-diffusion equations [J]. SIAM Journal on Scientific Computing, 2012, 34 (4): A2288-A2316.
- [6] D'Ambra P, Buttari A, Di Serafino D, et al. A novel aggregation method based on graph matching for algebraic multigrid preconditioning of sparse linear systems [C]//Proc of International Conference on Preconditioning Techniques for Scientific & Industrial Applications. 2011.
- [7] Jr Dendy J E, Moulton J D. Black box multigrid with coarsening by a factor of three [J]. Numerical Linear Algebra with Applications, 2010, 17 (2-3): 577-598.
- [8] Vanek P, Mandel J, Brezina M. Algebraic multigrid by smoothed aggregation for second order and fourth order elliptic problems [J]. Computing, 1996, 56: 179-196.
- [9] Chen Menghuo, Greenbaum A. Analysis of an aggregation-based algebraic two-grid method for a rotated anisotropic diffusion problem [J]. Numerical Linear Algebra with Applications, 2015, 22 (4): 681-701.
- [10] Braess D. Towards algebraic multigrid for elliptic problems of second order [J]. Computing, 1995, 55: 379-393.
- [11] Deng Liangjian, Huang T Z, Zhao Xilei, et al. An economical aggregation algorithm for algebraic multigrid (AMG) [J]. Journal of Computational Analysis and Applications, 2014, 16 (1): 181-198.
- [12] Wu Jianping, Song Junqiang, Zhang Weimin, et al. Coarse grid correction to domain decomposition based preconditioners for meso-scale simulation of concrete [J]. Applied Mechanics and Materials, 2012, 204-208: 4683-4687.
- [13] Kumar P. Aggregation based on graph matching and inexact coarse grid solve for algebraic two grid [J]. International Journal of Computer Mathematics, 2014, 91 (5): 1061-1081.
- [14] Wu Jianping, Yin Fukang, Peng Jun, et al. An algebraic multigrid preconditioner based on aggregation from top to bottom [C]//Proc of the 4th Annual International Conference on Information Technology and Applications. 2017.