

文章编号: 1000-0887(2004)09-0927-08

# 稀疏近似逆与多层块 ILU 预条件技术\*

谷同祥<sup>1,2</sup>, 迟学斌<sup>2</sup>, 刘兴平<sup>1</sup>

(1. 北京应用物理与计算数学研究所 计算物理实验室, 北京 8009 信箱, 北京 100088;  
2. 中科院 计算机网络信息中心 超级计算中心, 北京 349 信箱, 北京 100080)

(我刊原编委吴启光推荐)

**摘要:** 设计了一种求解一般稀疏线性方程组的健壮且有效的可并行化预条件子, 这种预条件子涉及在多层块 ILU 预条件子 (BILUM) 中使用稀疏近似逆 (AINV) 技术。所得的预条件子保持了 BILUM 的健壮性, 它比标准的 BILUM 预条件子有两点优势: 控制稀疏性的能力和增强了并行性。数值例子显示了新预条件子的有效性和效率。

**关键词:** 稀疏矩阵; 预条件技术; BILUM; AINV; Krylov 子空间方法  
**中图分类号:** O241.6      **文献标识码:** A

## 引 言

目前, 大型稀疏线性代数方程组迭代求解技术流行的选择是预条件 Krylov 子空间方法, 见如[1]。预条件迭代法的收敛速度通常由预条件子的质量来决定, 有了高质量的预条件子, 选择 Krylov 子空间方法加速的方式就不再是关键。这使得人们由着重寻找有效的迭代解法器转向构造健壮的预条件子<sup>[2,3]</sup>。并行和分布式处理机的出现与流行使得寻求适合高性能计算机的可并行化预条件子变得越来越重要。

构造“一般目的”的预条件子需要权衡其有效性、一般性、并行性和健壮性。对可并行化的预条件子, 近年来出现了几种稀疏近似逆技术<sup>[4-9]</sup>, 这些预条件子可提供较高的并行性且对某些类问题被证明是有效的。然而, 对一般稀疏矩阵, 它们的健壮性并不强, 未达到可与传统串行预条件子(如 ILUT<sup>[10]</sup>)相媲美的水平。另一方面, ILUT 直接实现中并行性是有限的, 特别是当需要用高精度分解时。

许多作者已经注意到稀疏近似逆技术的一些缺陷和限制<sup>[7,8,11]</sup>。其中之一是由于内在的局部耦合性使它们对小规模问题更有效<sup>[7,11]</sup>。因此, 这些作者建议在块形式整体预条件子中, 如块 SSOR 和基于 Schur 补技术的预条件子中, 将稀疏近似逆用作局部预条件子。

多层块 ILU 预条件子 (BILUM)<sup>[3,12]</sup> 推广了具有多层结构的逐次无关集排序的概念, 具有较好的并行度, 对某些类问题可以导致一个几乎与问题规模无关的收敛速度<sup>[3]</sup>。BILUM 中无

\* 收稿日期: 2002\_05\_28; 修订日期: 2004\_03\_30

基金项目: 国家自然科学基金资助项目 (60373015); 国家高技术研究发展计划基金资助项目 (2001AA111043); 计算物理国家重点实验室资助项目

作者简介: 谷同祥 (1964—), 男, 汉族, 河南焦作人, 副研究员, 博士 (联系人, Tel: + 86\_10\_62014411\_2967; Fax: + 86\_10\_62057289; E\_mail: txgu@iapcm.ac.cn)。

关块的求解使用精确逆或基于奇异值分解的正则化逆(regularized inverse), 这些策略对小的块是有效的, 然而, 求逆的花费随着块大小的增加而急剧增加。对具有大块的基于区域分解的实现, 预条件子的这些构造花费是不能容忍的。另一个相关的问题是较高的存储花费, 因为大块不再是稠密的而其逆却是稠密的。

类似于[13], 我们考察在 BILUM 的构造中用稀疏近似逆技术求大块的逆。对最终的约化系统也构造一个稀疏近似逆并用作对最终层上近似解的一个解法器。事实上, 几乎所有基于 Schur 补技术的多层方法都使用了某种稀疏近似逆技术, 其中一些是非常简单的。然而, 在一般目的的多层方法中使用一般目的的稀疏近似逆技术似乎是新的且有希望的。

## 1 稀疏近似逆

稀疏近似逆是一个稀疏矩阵  $M$ , 它是一般稀疏矩阵  $A$  之逆  $A^{-1}$  的一个好的近似。寻求有效稀疏近似逆的主要动力是它们在并行计算中的潜在优势。而且, 有证据显示这种类型的预条件子可解一些 ILU 预条件子难以求解的问题<sup>[7]</sup>, 并且提供了传统 ILU 预条件子的一种替代。

当前的稀疏近似逆预条件技术大致有三种<sup>[14]</sup>: 基于 Frobenius 范数极小化的稀疏近似逆<sup>[6~9]</sup>、分解的稀疏近似逆<sup>[4, 15~18]</sup>、由 ILU 分解计算的稀疏近似逆<sup>[11]</sup>。每一种都有一些不同的构造方式且每一种都有其优点和缺陷。换言之, 在所有可比较的规则中(构造花费、应用花费、健壮性、有效性等), 没有哪种是绝对最优的。Benzi<sup>[14]</sup>给出了稀疏近似逆的一个全面的总结及几种已有稀疏近似逆技术的比较。这里仅给出将用于 BILUM 预条件技术中的 AINV 稀疏近似逆<sup>[15]</sup>。

考虑迭代求解大型稀疏线性方程组

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

其中  $A \in R^{n \times n}$ 。由 Benzi 等提出的 AINV 是基于不完全(双)共轭过程构造一个分解的稀疏近似逆<sup>[15]</sup>, 它计算两个相互  $A$  共轭的向量集合  $\{z_i\}_{i=1}^n$  和  $\{w_i\}_{i=1}^n$ , 使得当且仅当  $i \neq j$  时  $w_i^T A z_j = 0$ 。如果我们引入矩阵

$$Z = [z_1, z_2, \dots, z_n] \text{ 和 } W = [w_1, w_2, \dots, w_n]$$

则

$$W^T A Z = D = \text{diag}(p_1, p_2, \dots, p_n),$$

其中  $p_i = w_i^T A z_i \neq 0$ 。从而  $W$  和  $Z$  是非奇异的且

$$A^{-1} = Z D^{-1} W^T = \sum_{i=1}^n \frac{z_i w_i^T}{p_i}. \quad (2)$$

因此, 如果知道了两个完整的  $A$  共轭向量的集合, 则可得  $A$  的逆。假设  $A$  具有 LU 分解, 列  $A$  共轭的矩阵  $W$  和  $Z$  可通过应用于标准基向量  $e_1, e_2, \dots, e_n$  上的一个双共轭过程来显式地计算出。如果  $A = LDU$ , 易见  $Z = U^{-1}$  且  $W = L^{-T}$ , 其中  $L$  和  $U$  为单位下和上三角的,  $D$  为对角的。若记  $a_i^T$  为  $A$  的第  $i$  行,  $c_i$  为  $A$  的第  $i$  列, 则计算  $Z$  的双共轭过程可写为:

算法 1 (AINV 算法)

$$\text{置 } z_1^{(0)} = e_1, p_1^{(0)} = a_{11}$$

对  $i = 2, \dots, n,$

$$z_i^{(0)} = e_i,$$

对  $j = 1, 2, \dots, i-1$

$$p_i^{(j-1)} := a_j^T z_i^{(j-1)},$$

$$z_i^{(j)} := z_i^{(j-1)} - \left( \frac{p_i^{(j-1)}}{p_j^{(j-1)}} \right) z_j^{(j-1)},$$

结束循环•

$$p_i^{(i-1)} := \mathbf{a}^T z_i^{(i-1)},$$

结束循环•

除了  $\mathbf{a}_i^T$  被换成  $c_i$  外,  $W$  的计算是相同的• 注意  $Z$  和  $W$  可同时计算• 为了得到一个稀疏的预条件子,  $Z$  和  $W$  可被不完全地计算, 如扔掉那些绝对值小于事先给定阈值的元素• 这导致不完全因子  $Z \approx \tilde{Z}$ 、 $W \approx \tilde{W}$  和  $D \approx \tilde{D}$ , 且分解的近似逆形式为  $M = \tilde{Z} \tilde{D}^{-1} \tilde{W}^T$ •

然而需要注意的是: 当主元素为负或零时, 算法 1 将出现中断• Benzi<sup>[15]</sup> 指出, 如果  $A$  为一个  $H$  矩阵, 则  $p_i = z_i^T A z_i > 0$ , 即不会发生中断, 从而算法 1 的不完全过程是有定义的• 当不发生中断时, 所得的稀疏近似逆通常是非常有效的<sup>[14]</sup>• 在一般情形中可能发生中断, 中断对预条件子的质量有着关键的影响• 虽然在实际中遇到零主元的情况非常少, 但是可能遇到非常小的主元, 这将导致  $Z$  和  $W$  中的元素非常大, 出现很高的填充并引起数值不稳定• 为了避免中断的发生, 通过分析主元出现负或零的原因并避免之, [16] 提出了一种稳定的(且健壮的) AINV 预条件技术, 即 SAINV•

## 2 BILUM 与 AINV

BILUM 预条件技术使用了块无关集的概念• 一个块无关集定义为一个未知量组(块)的集合, 使得任意两个不同组(块)的未知量之间不耦合<sup>[3]</sup>• 传统的点无关集被认为是一个大小均为 1 之块的块无关集, 有多种策略可用来寻找具有不同性质的块无关集<sup>[3, 19]</sup>•

先排与无关集相应的未知量, 然后再排其它未知量• 与此排序相应的置换阵  $P_\alpha$  将原矩阵转化成一个  $2 \times 2$  块结构矩阵

$$A_\alpha \sim P_\alpha A_\alpha P_\alpha^T = \begin{pmatrix} D_\alpha & F_\alpha \\ E_\alpha & C_\alpha \end{pmatrix}, \tag{3}$$

其中  $D_\alpha$  是一个  $m_\alpha$  维的块对角矩阵,  $C_\alpha$  为一个  $n_\alpha - m_\alpha$  维方阵• 这里  $\alpha$  表示层数且  $A_0 = A$ • 为简单起见, 我们将置换前和置换后的矩阵都表示成  $A_\alpha$ •

在代数多层预条件技术中, 约化系统被递归地构造成关于  $D_\alpha$  或  $C_\alpha$  的 Schur 补• 在 BILUM 中, 这种构造实际上是执行如下形式的一个块 LU 分解<sup>[3, 19]</sup>

$$\begin{pmatrix} D_\alpha & F_\alpha \\ E_\alpha & C_\alpha \end{pmatrix} = \begin{pmatrix} I_\alpha & \mathbf{0} \\ E_\alpha D_\alpha^{-1} & I_\alpha \end{pmatrix} \times \begin{pmatrix} D_\alpha & F_\alpha \\ \mathbf{0} & A_{\alpha+1} \end{pmatrix}, \tag{4}$$

其中  $A_{\alpha+1}$  是关于  $C_\alpha$  的 Schur 补,  $I_\alpha$  为第  $\alpha$  层上的单位矩阵• 注意  $n_{\alpha+1} = m_\alpha$ ,  $D_\alpha = \text{diag}(D_{\alpha,1}, D_{\alpha,2}, \dots, D_{\alpha,l})$  为块对角矩阵• 我们用某种扔掉策略来控制填充量, 即或者扔掉绝对值小的量, 或者另外还限制  $L$  和  $U$  的每一行中允许的元素总数<sup>[3, 19, 20]</sup>• 两种策略分别被称为单扔掉策略和双扔掉策略• 所得的多层块 ILU 分解用作某种 Krylov 子空间方法的预条件子•

在[3]的 BILUM 的实现中, 块对角阵  $D_\alpha$  包含的是小块• 这些小块通常是稠密的且使用精确求逆技术通过求每一个小块的逆来计算  $D_\alpha^{-1}$ • 在[12]中, 用基于奇异值分解的正则化逆技术来近似地求这些(几乎奇异的)块的逆• 正如我们在引言中注意到的, 这些直接求逆方法通常导致稠密的逆矩阵, 尽管原来块高度稀疏•

大多数多层方法利用  $D_\alpha$  的某些特殊性来构造关于  $D_\alpha$  的一个稀疏近似逆  $M_\alpha$ • 我们的新

方法使用 Benzi 等的 AINV 算法对每一个块  $D_{\alpha,i}$  计算一个稀疏近似逆  $M_{\alpha,i}$  且用  $M_{\alpha} = \text{diag}(M_{\alpha,1}, M_{\alpha,2}, \dots, M_{\alpha,l})$  作为  $D_{\alpha}$  的一个稀疏近似逆。这至少比直接计算  $D_{\alpha}$  的一个稀疏近似逆更便于并行计算, 而且, 我们的计算不损害块之间任何已存在的  $D_{\alpha}^{-1}$  的耦合。另外, 稀疏近似逆的计算可并行执行, 这也是我们除了对最终约化系统稀疏近似逆的构造外, 不需要稀疏近似逆算法具有并行性的原因。

因此, 分解(4)被替代为

$$\begin{pmatrix} D_{\alpha} & F_{\alpha} \\ E_{\alpha} & C_{\alpha} \end{pmatrix} \approx \begin{pmatrix} I_{\alpha} & \mathbf{0} \\ E_{\alpha}M_{\alpha} & I_{\alpha} \end{pmatrix} \times \begin{pmatrix} M_{\alpha}^{-1} & F_{\alpha} \\ \mathbf{0} & A_{\alpha+1} \end{pmatrix} = L_{\alpha}U_{\alpha}, \tag{5}$$

其中近似 Schur 补  $A_{\alpha+1} = C_{\alpha} - E_{\alpha}M_{\alpha}F_{\alpha} \cdot M_{\alpha}^{-1}$  的使用仅是为了符号方便, 在预条件过程中, 我们仅需要  $M_{\alpha}$ 。通常, 可对  $A_{\alpha+1}$  应用一个附加的扔掉策略以保持预条件子所期望的稀疏性。

图 1 是具有稀疏近似逆的一层 BILUM 分解的一个解释。

BILUM 预条件子基于 ILU 分解(5)。在每层  $\alpha$  上, 执行一个块 ILU 分解并形式如(5)的近似约化系统, 整个过程: 寻找块无关集、置换矩阵、执行并对 ILU 分解块  $A_{\alpha+1}$  重复进行。这种递归当最终约化系统  $A_{\mathcal{L}}$  足够小, 能构造一个有效的 AINV 预条件子时停止。然后, 用 AINV 算法为  $A_{\mathcal{L}}$  构造一个稀疏近似逆  $M_{\mathcal{L}}$ 。然而, 我们在任何层上都不存储任何约化系统, 包括最后一个, 代之, 我们存储稀疏矩阵

$$\begin{pmatrix} M_{\alpha} & F_{\alpha} \\ E_{\alpha} & \end{pmatrix}, 0 \leq \alpha \leq \mathcal{L} - 1$$

以及  $M_{\mathcal{L}}$ 。事实上, (5) 中的积  $E_{\alpha}M_{\alpha}$  仅对近似 Schur 补计算而不存储。在预条件期间  $L_{\alpha}$  中  $E_{\alpha}M_{\alpha}$  的作用由被替代为应用  $M_{\alpha}$  于一个向量, 接着再应用  $E_{\alpha}$ 。

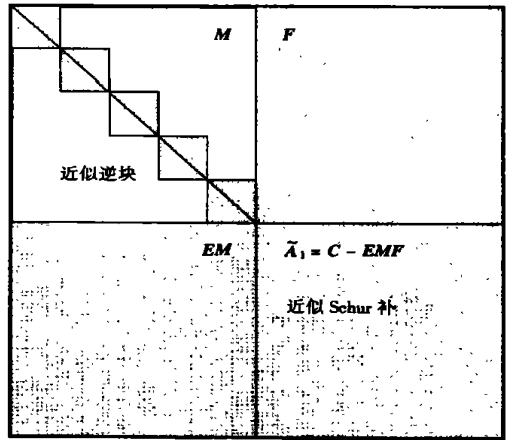


图 1 具有稀疏近似逆的一层 BILUM 分解的一个解释

在最终层上, 近似解通过应用  $M_{\mathcal{L}}$  于相应的向量而得到。这不同于[3]中 BILUM 的实现, [3]中最终约化系统由 ILUT 预条件的一个 Krylov 方法在某种精度下求解, 故利用了一个内-外迭代方案且需要使用 GMRES 的一个灵活形式, 即 FGMRES<sup>[21]</sup>, 它允许使用可变的预条件子。

假设右端项  $b$  和解向量  $x$  依据(3)中的无关集排序来划分, 在每一层我们有

$$x_{\alpha} = \begin{pmatrix} x_{\alpha,1} \\ x_{\alpha,2} \end{pmatrix}, b_{\alpha} = \begin{pmatrix} b_{\alpha,1} \\ b_{\alpha,2} \end{pmatrix}.$$

向前消去即求解一个临时向量  $y_{\alpha}$ , 亦即对  $\alpha = 0, 1, \dots, \mathcal{L} - 1$ , 求解

$$\begin{pmatrix} I_{\alpha} & \mathbf{0} \\ E_{\alpha}M_{\alpha} & I_{\alpha} \end{pmatrix} \begin{pmatrix} y_{\alpha,1} \\ y_{\alpha,2} \end{pmatrix} = \begin{pmatrix} b_{\alpha,1} \\ b_{\alpha,2} \end{pmatrix}, \begin{matrix} \text{(F1): } y_{\alpha,1} = M_{\alpha}b_{\alpha,1}, \\ \text{(F2): } y_{\alpha,2} = b_{\alpha,2} - E_{\alpha}y_{\alpha,1}. \end{matrix}$$

注意: 如上讨论, 由于我们不计算乘积矩阵  $E_{\alpha}M_{\alpha}$ , 所以分离了  $M_{\alpha}$  和  $E_{\alpha}$  的作用。

计算最终约化系统的一个近似解为  $x_{\mathcal{L}} = M_{\mathcal{L}}y_{\mathcal{L}}$

向后回代即对  $\alpha = \mathcal{L} - 1, \dots, 1, 0$  求解

$$\begin{pmatrix} M_{\alpha}^{-1} & F_{\alpha} \\ \mathbf{0} & \mathbf{0} \end{pmatrix} \begin{pmatrix} x_{\alpha,1} \\ x_{\alpha,2} \end{pmatrix} = \begin{pmatrix} y_{\alpha,1} \\ y_{\alpha,2} \end{pmatrix}, \begin{matrix} \text{(B1): } x_{\alpha,1} = y_{\alpha,1} - F_{\alpha}x_{\alpha,2}, \\ \text{(B2): } x_{\alpha,1} = M_{\alpha}x_{\alpha,1}. \end{matrix}$$

由于  $x_{\alpha, 2} = x_{\alpha+1, 1}$  且  $x_{\alpha, 1, 2} = x_{\alpha}$  回代可以进行。

在上面的求解过程的讨论中, 我们省略了各层上运算前后必需的置换和逆置换, 只需在计算完成后置换解向量, 这也是我们程序设计中所用的方法。另一方面, 我们也可以在构造阶段的每层上置换矩阵, 此时, 仅在预条件子的应用前后需要整体置换<sup>[19]</sup>。基于区域分解, 具有大块(大无关集)的 BILUM 的实现中, 大多数节点在前几步约化中已经被分解出, 对粗层(短)向量的附加后置换是很小的, 后置换的冗余花费是可忽略的。

注意: 所有计算部分仅为矩阵向量乘运算或向量校正, 因此在每层内完全可并行化的, 即方法没有传统 ILU 类分解的串行缺陷。

### 3 数值试验

BILUM 的标准实现在[3, 19]中已有描述, BILUM\_AINV 的实现与 BILUM 的实现不同之处在于: 不用内迭代过程求解最终的约化方程组。因此是一个固定的近似解过程。下面的例子来自[13], 我们的试验使用了 Zhang Jun 教授网页上公布的 BILUM 程序库(见 <http://www.cs.uky.edu/~jzhang/bilum.html>)。

考虑定义在单位正方形上的对流扩散问题

$$-u_{xx} - u_{yy} - Re(\sin x \cos \pi y \cdot u_x - \cos \pi x \sin y \cdot u_y) = 0, \tag{6}$$

这里  $Re$  为 Reynolds 数, 使用 Dirichlet 边界条件。用一个九点四阶紧凑有限差分离散方案离散方程(6), 其中一致网格  $h = 1/101$ 。对不同的  $Re$  值, 所得的矩阵具有 10 000 个未知量和 88 804 个非零元素, 称之为九点矩阵。随着  $Re$  的增大, 矩阵的对角占优性越来越弱<sup>[23]</sup>。

解向量设为分量全为 1 并相应生成右端项, 初始猜测为一些随机数组成的向量。当残差的 2 范数下降一个因子  $10^7$  时终止计算。除非有特殊说明, 我们对预条件迭代解法器使用缺省参数: 50 步再启动 GMRES 作为加速器; 允许最大约化层数为 10, 即  $\mathcal{L} = 10$ ; 扔掉阈值为  $\tau = 10^{-4}$ 。

在下面所有具有数值结果的表中, 块大小取为一致的, 时间  $t$  (单位为 s), 表示 CPU 时间, 稀疏性比率即预条件子中非零元素的个数与原矩阵非零元素的个数之比, “-”表示经最大迭代次数后仍未收敛。

表 1 求解具有不同  $Re$  值九点矩阵之预条件子的比较

$Re$	ILUT			AINV			BILUM_AINV		
	迭代次数	预条件子构造时间 (t/s)	求解时间 (t/s)	迭代次数	预条件子构造时间 (t/s)	求解时间 (t/s)	迭代次数	预条件子构造时间 (t/s)	求解时间 (t/s)
0	10	0.99	1.09	102	57.51	8.76	15	14.91	2.37
1	10	0.99	1.09	118	58.42	10.46	15	14.78	2.36
10	11	0.97	1.22	209	56.94	18.24	18	14.87	2.74
100	10	0.92	1.06	227	56.82	19.92	19	14.82	2.93
1 000	7	0.89	0.75	243	57.28	23.64	10	14.64	1.46
10 000	14	1.05	1.55	2 052	59.78	187.44	10	14.88	1.47
100 000	19	1.11	2.17	-	-	-	27	16.65	4.44
1 000 000	21	1.12	3.31	-	-	-	30	14.94	5.06

我们的试验给出用不完全 LU 分解(ILUT)、稀疏近似逆(AINV)及具有稀疏近似逆的多层块 ILU(BILUM\_AINV)作为 GMRES 之预条件子的性能比较。对 ILUT 和 BILUM\_AINV, 用双扔掉策略。表 1 给出对不同  $Re$  和固定稀疏比率约为 4.5 的求解九点矩阵预条件子的性能比较。在这种特殊情形中, AINV 的最大迭代数设为 5 000。我们发现随着矩阵对角占优性变差,

AINV 退化得非常快, 当  $Re \geq 10^5$  时不收敛。ILUT 在 8 种情形中有 7 种比 BILUM\_AINV 有效得多, 两者之间的差别并不是太显著, 如果两者都在并行计算机上实现, 情况可能相反。然而, AINV 和 BILUM\_AINV 的差别是显著的, 可见, BILUM\_AINV 在有效性和并行性之间提供了一个好的协调。

我们进一步检测用不同的扔掉策略和不同的块大小的 BILUM\_AINV 求解  $Re = 100$  的方程(6)。对双扔掉策略, 我们在每个 L 和 U 因子和每个块中保留至多 20 个(依绝对值的)最大元素。我们将块大小从 20 变到 90, 相应的块填充从 100% 到 22%, 结果在表 2 中给出。可以见到, 随着块大小的增加(具有定量的填充), 迭代量增加而稀疏比率下降。这正是我们所期望的, 有了这种灵活性我们可根据不同并行机的结构与配置调整预条件子。除了几乎全填充的情形, 双和单扔掉策略结果相类似。这说明 BILUM\_AINV 总的稀疏性受块之稀疏近似逆的稀疏性所影响。

表 2 用不同扔掉策略的 BILUM\_AINV 求解九点矩阵之比较 ( $Re = 100$ )

块大小	单扔掉策略				双扔掉策略			
	迭代次数	预条件子构造时间 (t/s)	求解时间 (t/s)	稀疏性比率	迭代次数	预条件子构造时间 (t/s)	求解时间 (t/s)	稀疏性比率
20	24	21.34	3.01	6.46	28	5.89	2.74	3.27
30	33	10.65	3.76	4.34	35	7.34	3.64	3.05
40	45	8.57	4.75	3.14	46	7.92	4.72	2.86
50	45	9.30	4.73	3.12	46	8.54	4.68	2.82
60	56	9.24	5.97	2.80	57	8.71	5.97	2.73
70	79	9.38	7.81	2.65	79	9.15	8.05	2.63
80	88	9.59	9.03	2.62	89	9.25	9.12	2.60
90	102	9.64	10.78	2.56	102	9.42	10.86	2.56

表 3 用不同块大小及单扔掉策略的 BILUM\_AINV 和 BILUM 求解九点矩阵之比较 ( $Re = 1000$ )

块大小	BILUM_AINV				BILUM			
	迭代次数	预条件子构造时间 (t/s)	求解时间 (t/s)	稀疏性比率	迭代次数	预条件子构造时间 (t/s)	求解时间 (t/s)	稀疏性比率
5	52	14.74	5.26	3.50	3	11.87	1.02	5.56
10	49	8.79	4.64	3.00	3	9.10	0.84	5.48
15	37	9.07	3.35	3.26	3	9.46	0.80	5.79
20	37	8.11	3.48	3.07	3	6.84	0.56	5.37
30	32	7.96	2.87	3.38	3	6.35	0.55	6.16
40	27	9.51	2.53	3.67	3	5.36	0.59	6.79
50	23	11.22	2.14	4.18	3	6.16	0.68	7.99
60	20	12.32	2.01	4.40	3	6.03	0.74	8.78
70	19	14.31	2.12	4.70	3	5.77	0.79	9.63
80	12	16.28	2.07	5.09	3	5.61	0.77	10.6

表 3 给出了用不同块大小的 BILUM\_AINV 与 BILUM 求解九点矩阵的比较, 其中  $Re = 1000$ , 两者均用单扔掉策略。在 BILUM\_AINV 情形中, 我们仅保留块之稀疏近似逆的大约一半的元素。我们见到, BILUM 更有效和快速, 但是需用的存储空间是 BILUM\_AINV 所需的两倍。事实上, 当块大小大于 50 时, BILUM 就溢出了预先设置的工作数组, 我们必须调整存储分配。此检测说明两者各有优缺点, 例如, 若存储有限, 则 BILUM\_AINV 更灵活。

## 6 结 论

我们研究了用稀疏近似逆与多层块 ILU 预条件技术(BILUM) 结合求解一般稀疏矩阵。新方法提供了控制 ILU 分解期间的填充量的灵活性, 此时基于多层方法实现的区域分解可使用较大的块。

我们的数值试验显示: 我们所提出的组合方式确实具有预期的灵活性与有效性。作为可并行化的高精度预条件子, BILUM\_AINV 比标准的 AINV 技术更有效。BILUM\_AINV 比标准 BILUM 的优点是节省存储。这种新的可并行化预条件子继承了 ILU 预条件的有效性和健壮性, 以及多层结构和稀疏近似逆技术的并行性。事实上, 它提供了有效性和并行性之间的一个好的平衡。

BILUM\_AINV 比 AINV 优越在于健壮性和有效性, 但对 BILUM\_AINV、BILUM 和 ILUT 的相互优越性未有明显的结论, 作为一般目的的预条件子, 它们中任何一个都不绝对优于其它两个。这个结论是正常的, 因为一种共识是没有任何一种一般目的的预条件子对任何类型的问题都是最佳的, 研究者的任务是鉴别对某一大类问题有效的特殊预条件子。从这一观点来看, 我们见到 BILUM\_AINV 提供了对 BILUM 和 ILUT 很好的弥补, 因为它可以处理后难于处理的一些问题。

致谢 在本文完成之际, 我们要特别感谢美国 Kentucky 大学的 Zhang Jun 教授, 我们的研究思路源自他的一些论文并且在我们的数值试验中使用了来自其网页的 BILUM 程序库。

### [参 考 文 献]

- [1] Saad Y. Iterative Methods for Sparse Linear Systems [M]. New York: PWS Publishing, 1996.
- [2] Golub G H, van der Vorst H A. Closer to the solution: iterative linear solver[A]. In: Duff I S, Watson G A Eds. The State of the Art in Numerical Analysis [C]: Oxford: Clarendon Press, 1997, 63—92.
- [3] Saad Y, ZHANG Jun. BILUM: block versions of multi\_elimination and multi\_level ILU preconditioner for general sparse linear systems[J]. SIAM J Sci Comput, 1999, 20(6): 2103—2121.
- [4] Benzi M, Tuma M. A sparse approximate inverse preconditioner for nonsymmetric linear systems[J]. SIAM J Sci Comput, 1998, 19(3): 968—994.
- [5] Chan T F, TANG Wei\_pai, Wan W L. Wavelet sparse approximate inverse preconditioners[J]. BIT, 1997, 37(3): 644—660.
- [6] Chow E, Saad Y. Approximate inverse techniques for block\_partitioned matrices[J]. SIAM J Sci Comput, 1997, 18(6): 1657—1675.
- [7] Chow E, Saad Y. Approximate inverse preconditioner via sparse\_sparse iterations[J]. SIAM J Sci Comput, 1998, 19(3): 995—1023.
- [8] Gould N I M, Scott J A. Sparse approximate inverse preconditioners using norm minimization techniques[J]. SIAM J Sci Comput, 1998, 19(2): 605—625.
- [9] Grote M, Huckle T. Parallel preconditioning with sparse approximate inverses[J]. SIAM J Sci Comput, 1997, 18(3): 838—853.
- [10] Saad Y. ILUT: a dual threshold incomplete ILU preconditioner[J]. Numer Linear Algebra Appl, 1994, 1(4): 387—402.
- [11] TANG Wei\_pai, Wan W L. Sparse approximate inverse smoother for multigrid[J]. SIAM J Matrix Anal Appl, 2000, 21(4): 1236—1252.
- [12] Saad Y, ZHANG Jun. Enhanced multi\_level block ILU preconditioning strategies for general linear systems[J]. J Comput Appl Math, 2001, 130(1/2): 99—118.

- [13] ZHANG Jun. Sparse approximate inverse and multilevel block ILU preconditioning techniques for general sparse matrices[J]. Appl Numer Math, 2000, **35**(1): 67—86.
- [14] Benzi M, Tuma M. A comparative study of sparse approximate inverse preconditioners[J]. Appl Numer Math, 1999, **30**(2/3): 305—340.
- [15] Benzi M, Meyer C D, Tuma M. A sparse approximate inverse preconditioner for the conjugate gradient method[J]. SIAM J Sci Comput, 1996, **17**(5): 1135—1149.
- [16] Benzi M, Cullum J K, Tuma M. Robust approximate inverse preconditioning for the conjugate gradient method[J]. SIAM J Sci Comput, 2000, **22**(4): 1318—1332.
- [17] Benzi M, Tuma M. Orderings for factorized sparse approximate inverse preconditioners[J]. SIAM J Sci Comput, 2000, **21**(5): 1851—1868.
- [18] Kolotina L Y, Yeremin A Y. Factorized sparse approximate inverse preconditioning I : theory[J]. SIAM J Matrix Anal Appl, 1993, **14**(1): 45—58.
- [19] Saad Y. ILUM: a multi-elimination LU preconditioner for general sparse matrices[J]. SIAM J Sci Comput, 1996, **17**(4): 830—847.
- [20] Saad Y, Sosonkina M, ZHANG Jun. Domain decomposition and multi\_level type techniques for general sparse linear systems[A]. In: Mandel J, Farhat C, Cai X C, Eds. Domain Decomposition Methods 10 [C]. number 218 in Contemporary Mathematics, Providence, RI, AMS, 1998, 174—190.
- [21] Saad Y. A flexible inner\_outer preconditioned GMRES algorithm[J]. SIAM J Sci Comput, 1993, **14**(2): 461—469.
- [22] Saad Y, ZHANG Jun. BILUTM: a domain\_based multi\_level block ILUT preconditioner for general linear systems[J]. SIAM J Matrix Anal Appl, 1999, **21**(1): 279—299.
- [23] ZHANG Jun. On convergence of iterative methods with a fourth\_order compact scheme[J]. Appl Math Lett, 1997, **10**(2): 49—55.

## AINV and BILUM Preconditioning Techniques

GU Tong\_xiang<sup>1, 2</sup>, CHI Xue\_bin<sup>2</sup>, LIU Xing\_ping<sup>1</sup>

(1. Laboratory of Computational Physics, Institute of Applied Physics and Computational Mathematics, P. O. Box 8009, Beijing 100088, P. R. China;

2. Supercomputing Center of Computer Network Information Center, Chinese Academy of Science, P. O. Box 349, Beijing 100080, P. R. China)

**Abstract:** It was proposed that a robust and efficient parallelizable preconditioner for solving general sparse linear systems of equations, in which the use of sparse approximate inverse (AINV) techniques in a multi\_level block LU (BILUM) preconditioner were investigated. The resulting preconditioner retains robustness of BILUM preconditioner and has two advantages over the standard BILUM preconditioner: the ability to control sparsity and increased parallelism. Numerical experiments are used to show the effectiveness and efficiency of the new preconditioner.

**Key words:** sparse matrix; preconditioning technique; BILUM; AINV; Krylov subspace method