第 40 卷 第 2 期 电 子 科 技 大 学 学 报 Vol.40 No.2
2011年3月 Journal of University of Electronic Science and Technology of China Mar. 2011

对称不定线性系统的不定预处理技术

李 良，黄廷祝

(电子科技大学数学科学学院计算科学研究所 成都 611731)

【摘要】研究求解对称不定线性系统 $\boldsymbol{A}x=b$ 的不定不完全分解预处理算法，其中$\boldsymbol{A}$为稀疏的对称不定矩阵。合适的选主元算法是成功分解不定矩阵的关键，为了加快选主元的速度，给出了松弛的有界Bunch-Kaufman (RBBK)对称选主元算法，并分析了该选主元算法的稳定性以及参数的选择范围。将RBBK算法与不完全Cholesky分解相结合，得到了一类稳定性较高的修改的不完全Cholesky分解预处理技术。MATLAB下的数值例子表明，将提出的预处理技术用于SQMR迭代算法时，得到较快的收敛速度。

关 键 词 分解； 迭代法； 线性方程组； 选主元； 预处理

中图分类号 TP301.6 **文献标识码** A doi:10.3969/j.issn.1001-0548.2011.02.026

Indefinite Preconditioning Techniques For for Symmetric Indefinite Linear Systems

LI Liang and ，HUANG Ting-zhu

(School of Mathematical Sciences, University of Electronic Science Technology of China Chengdu 611731)

Abstract In this paper, we study a class of factorized indefinite preconditioning techniques for solving linear system $\boldsymbol{A}x = b$, where **A** is sparse and symmetric indefinite. Choosing an appropriate pivoting strategy is the key for the success of factorization for an indefinite matrix. To speed up the process for selecting the pivot, we propose a relaxed bounded Bunch-Kaufman (RBBK) algorithm, analyze its stability, and derive the criteria for parameter selection. Combining RBBK algorithm with the incomplete Cholesky factorization, we obtain a kind of stable preconditioning technique via modified incomplete Cholesky factorization. Preconditioned by this kind of preconditioners, SQMR iteration converges very fast according to the presented numerical examples.

Key words factorization; iterative methods; linear equations; pivoting; preconditioning

1 背景知识介绍

科学工程计算中的很多问题，，比如计算电磁学，结构力学，、计算流体等，，最后都归结为求解如下线性方程组：

$$\boldsymbol{A}x=b \tag{1}$$

其中式中，， $A\in R^{n\times n}$ 为对称不定矩阵，， $x,b\in R^n$. $x,b\in R^n$ 。当n较小时，，可以用直接算法，，比如Aasen 算 法 [1]，] 、 Bunch-Kaufman 算 法 [2] 以 及 Bunch-Parlett算法[3].]。最近，，Ashcraft, Grimes和Lewis[4]文献[4]提出了有界的Bunch-Kaufman(BBK)算法和快速Bunch-Parlett (FBP)算法，，可提高这些直接求解器的效率，。在文献[5]中Li和Huang给出了BBK算法和FBP算法的松弛版本RBBK和RFBP，，使其有更大的灵活性. 。这些算法都是将式(1)中系数矩阵 $\boldsymbol{A}$ 重排为 $\boldsymbol{PAP}^{\mathrm{T}}$ ($\boldsymbol{P}$ 为相应的置换矩阵)。，$\boldsymbol{P}$ 为相应的置换矩阵.

但是，，随着n的增大，，直接求解器需要太多的存储量和计算时间而变得不适用，. 需要使用迭代算法来近似求解式(1).)。迭代法成功的关键是用适当的预处理技术加速其收敛速度. 。求解对称不定系统的常用Krylov子空间方法有SYMMLQ和MINRES方法[6]，]，这该二两个方法需要对称正定的预处理矩阵，，但是对不定的系数矩阵却用正定矩阵来预处理，，有一定的不合理性. 。幸运的是，，文献[7]Freud和Nachtigal提出了对称QMR算法(SQMR)，[7]，该算法可以结合使用包括不定矩阵在内的任意对称预处理矩阵. 。本文即通过SQMR迭代求解涉及的对称不定线性系统.。

令 $\boldsymbol{M}=\boldsymbol{M}^{\mathrm{T}}=M_1M_2$ 为任一可逆的n阶对称矩

收稿日期：2009 – 10 – 26; 修回日期：2010 – 10 – 16
基金项目：国家自然科学基金(60973015,11026084)；四川省科技项目(2009SPT-1，2009GZ0004，2009HH0025)
作者简介：李 良(1972 –)，男，博士，主要从事科学工程计算及应用方面的研究.

阵，，M_1和M_2不一定对称，，式(1)在两边预处理后转化为新的方程组：

$$A'x' = b'$$

其中式中，$A' = M_1^{-1}AM_2^{-1}$，$A' = M_1^{-1}AM_2^{-1}$；$x' = M_2x$，$x' = M_2x$；$b' = M_1^{-1}b$．$b' = M_1^{-1}b$。注意到，，如果令$M_1 = I$或$M_2 = I$，就变为了单边预处理，。本文中，这里I表示具有适当阶数的单位矩阵。.

虽然当阶数n很大时直接法不能用于求解式(1),，但直接法却为构造好的预处理算法提供了思路.。利用直接法来构造预处理算法这种思路已成功应用于不完全Cholesky(IC)和不完全LU(ILU)分解预处理技术上[8, 9].]。本文利用RBBK算法[5]来构造预处理算法，，实际上，，该这类算法是一种修改的IC(MIC)分解，，它将A分解为：

$$PAP^{\mathrm{T}} = LDL^{\mathrm{T}} + R$$

式中，其中P P为置换矩阵，；L为单位下三角矩阵，；D为块大小为1或2的块对角矩阵，；R为剩余的残量矩阵.。

接下来，第2节描述本文采用的选主元策略；第3节在该选主元策略的基础上给出MIC分解算法；第4节给出一些数值算例以展现本文提出算法的有效性；最后在第5节给出结论.

1　2　选主元算法

对非零对称矩阵$A \in R^{n\times n}$，$A \in R^{n\times n}$，一定存在一个置换矩阵P P和数$s = 1$或2,，使得：

$$PAP^{\mathrm{T}} = \begin{array}{cc} \begin{bmatrix} E & C^{\mathrm{T}} \\ C & B \end{bmatrix} & \begin{array}{c} s \\ n-s \end{array} \\ \begin{array}{cc} s & n-s \end{array} & \end{array}$$

式中，E为$s \times s$方阵？？？，且E可逆；C为$(n-s)\times s$矩阵？？？．。若以E为主元作一次消元就有：

$$PAP^{\mathrm{T}} = \begin{bmatrix} I_s & 0 \\ CE^{-1} & I_{n-s} \end{bmatrix} \begin{bmatrix} E & 0 \\ 0 & B - CE^{-1}C^{\mathrm{T}} \end{bmatrix} \begin{bmatrix} I_s & E^{-1}C \\ 0 & I_{n-s} \end{bmatrix}$$

应用以上过程处理Schur余$S = B - CE^{-1}C^{\mathrm{T}}$，$S = B - CE^{-1}C^{\mathrm{T}}$，这样递归处理下去，就得到修改的选主元Cholesky分解(MCP)[10-, 11].]，整个过程需要O($n^3/3$)的浮点计算以及选主元所需的比较和行列交换操作．。为了减少内存开销和计算量，，需要抛弃CE^{-1}和Schur余S中的一些非零元素，，这样就得到了不完全分解.。

BP选主元[3]算法每次需要搜索整个子矩阵S,，总共需要O(n^3)次比较；；BK算法[2]只搜索S的两列，，但得不到有界的L；；BBK算法能够得到有界的L,，但是对有些情况却需要与BP算法相当的比较量．。为了减少选主元过程中需要的比较操作，，本文我们引入一个松弛因子，，采用RBBK选主元，，增加了算法的灵活性，，减少了选主元所需的比较量．。RBBK算法将挑选出分解第一步的主元，RBBK算法细节如图1描述如下：.

RBBK算法:（本算法将挑选出分解第一步的主元.）

令γ_1为第1列中非对角元元素的最大模值;；

令r为第 1 列中第一个具有最大模值元素的行标;；

If　$\gamma_1 = 0$

不需要对第1列作任何处理;；　$s = 0$;；

else if　$|a_{11}| \geqslant \alpha\gamma_1$

令a_{11} a_{11}为主元;；$s = 1$; $P = I$;

else

$\text{flag} = 0$; $i = 1$; $\gamma_i = \gamma_1$;

while (flag = 0)

令γ_r为第i列中非对角元元素的最大模值;；

令r为第 i 列中第一个拥有最大模值元素的行标;；

if　$|a_{rr}| \geqslant \alpha\gamma_r$

令a_{rr} a_{rr}为主元;；$s = 1$;　$\text{flag} = 1$;

P交换第r行和第1行;

elseif　$\gamma_r \leqslant \beta\gamma_i$

令$\begin{bmatrix} a_{ii} & a_{ri} \\ a_{ri} & a_{rr} \end{bmatrix}$为主元；；$s = 2$;　$\text{flag} = 1$; $s = 2$; $\text{flag} = 1$;

P交换第1行和第i行，，第2行和第r行;

else

$i = r$; $\gamma_i = \gamma_r$;

end

end

end

根据该算法有如下定理来保证PMIC产生的三角矩阵$\boldsymbol{L}$的稳定性. 。

定理 1 如果由RBBK算法得到的是1×11×1的主元$\boldsymbol{a}_{rr}$，a_{rr}，那么有如下不等式成立：

$$|l_{ir}| \leqslant \frac{1}{\alpha} \qquad i \neq r \tag{2}$$

如果得到的是2×2 2×2的主元, ，那么有：

图1 RBBK算法

$$\begin{cases} |l_{i'i}| \leqslant \dfrac{\alpha+1}{\beta-\alpha^2} & i' \neq i,r \\ |l_{i'r}| \leqslant \dfrac{2\alpha}{\beta-\alpha^2} & i' \neq i,r \end{cases} \tag{3}$$

RBBK算法中有两个参数α和β，β，但由式是(2)和式(3)可知, ，当$\beta = 2\alpha^2 + \alpha$时, ，$\boldsymbol{L}$下三角的元素有上界$1/\alpha$，$1/\alpha$，那么实际上RBBK算法只需要一个自由参数$\alpha$，$\alpha$，就能保证矩阵$\boldsymbol{L}$的稳定性.。

2 3 修改的不完全Cholesky分解算法

如图2.

本节给出基于RBBK选主元的修改的不完全Cholesky分解(PMIC).)。对得到的Schur余递归地使用RBBK选主元, ，并进行选主元的Gauss消元, ，将最终得到修改的Cholesky分解, ，$\boldsymbol{A} = \boldsymbol{LDL}^{\mathrm{T}}$. $\boldsymbol{A} = \boldsymbol{LDL}^{\mathrm{T}}$。称之为修改的, ，是因为矩阵$\boldsymbol{D}$为块对角矩阵而不是通常的对角矩阵. 。抛弃$\boldsymbol{L}$中的一些非零元就得到PMIC分解. 。$P_n = \{(i,j) \mid 1 \leqslant i,j \leqslant n\}$表示矩阵中元素的位置, ，假定$\boldsymbol{S}$和$\boldsymbol{L}$有非零元模式$P_S \subseteq P_n$，$\boldsymbol{P_S} \subseteq P_n$，$P_L \subseteq P_n$，$\boldsymbol{P_L} \subseteq P_n$，$P_U \subseteq P_L^{\mathrm{T}}$，$\boldsymbol{P_U} \subseteq P_L^{\mathrm{T}}$，PMIC算法将得到$\boldsymbol{A}$的选主元$\boldsymbol{LDL}^{\mathrm{T}}$分解，PMIC算法描述如下：

$$\beta = 2\alpha^2 + \alpha; i = 1;$$

PMIC **算法**:（本算法将得到 A 的选主元 LDL^T 分解）

while $i < n$

将子矩阵 $\boldsymbol{A}(i:n, i:n)$ 运行 RBBK 算法；；

得到 s 为主元类型, ，矩阵 $\boldsymbol{P}_i$ 为相应置换矩阵；；

if s=0

本步骤不需要作任何处理；； $i = i+1$; $i = i+1$;

elseif s=1

根据 $\boldsymbol{P}_i$ P_i 重排矩阵 $\boldsymbol{A}$；； $i = i+1$; $i = i+1$;

$$\boldsymbol{D}(i,i) = \boldsymbol{A}(i,i); \boldsymbol{A}(i,i) = 1;$$

$$\boldsymbol{A}(i, i+1:n) = \frac{\boldsymbol{A}(i, i+1:n)}{\boldsymbol{D}(i,i)};$$

按照非零模式 P_U 去掉 $\boldsymbol{A}(j, i+1:n)$ $A(i, i+1:n)$ 中的非零元；；

for j=i+1:n

if $(j,i) \in \boldsymbol{P_L}$

$$\boldsymbol{A}(j, i+1:n) = \boldsymbol{A}(j, i+1:n) - \boldsymbol{A}(j, i+1:n)) * \boldsymbol{A}(j,i);$$

依照 $\boldsymbol{P_S}$ 去掉 $\boldsymbol{A}(j, i+1:n)$ $A(j, i+1:n)$ 中的非零元；；

end

$\boldsymbol{A}(j,i) = 0;$

end

else if s=2

end

elseif s=2

根据 $\boldsymbol{P}_i$ 重排矩阵 $\boldsymbol{A}$： $i = i+2$;

$$\boldsymbol{D}(i:i+1, i:i+1) = \boldsymbol{A}(i:i+1, i:i+1);$$

$$\boldsymbol{A}(i:i+1, i:i+1) = I_2;$$

$$\boldsymbol{A}(i:i+1, i+2:n) = \boldsymbol{A}(i:i+1, i+2:n)) * \boldsymbol{D}^{-1}(i:i+1, i:i+1);$$

根据 $\boldsymbol{P_U}$ 去掉 $\boldsymbol{A}(i:i+1, i+2:n)$ 中的非零元；；

for j=i+2:n

if $(j,i) \in \boldsymbol{P_L}$ or $(j,i+1) \in \boldsymbol{P_L}$

$$\boldsymbol{A}(j, i+2:n) = \boldsymbol{A}(j, i+2:n) - \boldsymbol{A}(j, i:i+1) * \boldsymbol{A}(i:i+1, i+2:n);$$

依照 $\boldsymbol{P_S}$ 去掉 $\boldsymbol{A}(j, i+2:n)$ 中的非零元；；

end

$\boldsymbol{A}(j, i:i+1) = 0;$

end

最后, 剩下的矩阵 $\boldsymbol{A}$ 即为要求的 $\boldsymbol{L}^{\mathrm{T}}$ 。

知道适当的非零模式 P_S， $\boldsymbol{P}_S$、 $\boldsymbol{P}_L$， $\boldsymbol{P}_L$， 因此我们根据元素的模值大小动态地决定矩阵的非零模式，，采用阈值ILU中的方法[9, 12].]，给定一个阈值τ，τ， 对向量$\boldsymbol{u}$,抛弃模值小于$\tau\|\boldsymbol{u}\|_2$ 的元素. ，这样得到的PMIC称为PMIC(τ). (τ)。本文中，这里，抛弃的意义是将本来非零的元素令为0，，以减少分解过程中所需的存储量和计算量.。

图2　PMIC算法

43　数值例子

本节文给出两个数值算例来说明PMIC(τ)预处理算法的有效性. 。在PMIC(τ)算法中，，$\boldsymbol{L}$和Schur余$\boldsymbol{S}$的稀疏模式只由其元素的相对大小确定，，所有的数值实验都由MATLAB实现，，SQMR迭代终止准则为残量$\|r^k\|_2 < 10^{-6}\|r^0\|_2$， $\|r^k\|_2 < 10^{-6}\|r^0\|_2$，$r^k$ 表示第k步的残量，， r^0 表示初始残量. 。我们调整右端项，，使得准确解为 $\boldsymbol{e} = (1,2,\cdots,1)^{\mathrm{T}}$， $\boldsymbol{e} = (1,2,\cdots,1)^{\mathrm{T}}$，所有的初始解向量都令为0. 。主元选择的参数用$\alpha$表示，，PMIC($\tau$)算法中的非零元抛弃阈值用$\tau$ 表示.。

34.1　BCSSTK19矩阵

BCSSTK19矩阵来源于矩阵市场[13],], 从力学工程中的动态分析中产生. ，其阶数为817，，有6 853个非零元，，为对称不定矩阵. 。一般的IC分解用于该矩阵时，，算法失败. 。表1给出了在不同的α 和τ取值下，，矩阵$\boldsymbol{L}$的非零元个数($nz(\boldsymbol{L})$)和预处理SQMR算法所需的迭代次数(its).)。

从表1可以看出，，当τ 固定，, $\boldsymbol{L}$的非零元个数随着α 的减小而减少. 。通常，，$\boldsymbol{L}$越稠密所需的迭代次数越少，，但是从表1看出，，这个该结论不是一定成立.。

表1　BCSSTK19矩阵

α	$\tau\tau=10^{-5}$		$\tau\tau=10^{-6}$	
	$nz(\boldsymbol{L})$	its	$nz(\boldsymbol{L})$	its
0.5	11 664	42	16 006	7
0.1	10 081	18	14 158	7
0.05	9 647	25	13 737	6
0.01	7 264	198	12 813	39
0.001	5 797	175	10 980	37

图1给出了当$\tau=10^{-5}$时，，在不同α 取值下选主元对矩阵稀疏模式的影响. 。可我们发现，，当α 较大时，，需要较多的行列交换操作，，随着α 的减小，，选主元对矩阵稀疏模式的影响就越小. 。因为当α 很小时，，几乎不用交换矩阵的行列，，就能很容易地找到满足要求的主元，。但同时从表1可以看出，此时得到的预处理矩阵的效果不是太好。

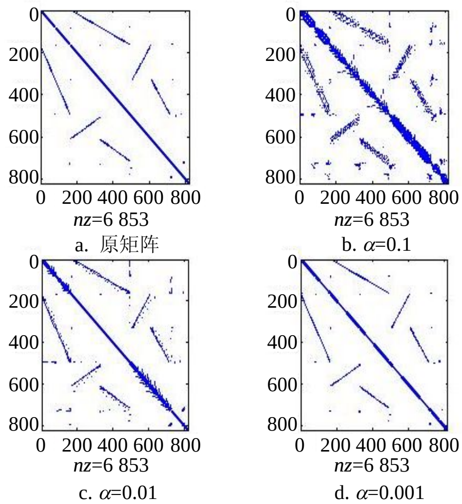

图1　不同$\alpha\alpha$ 取值下选主元对矩阵稀疏模式的影响

43.2　ex32矩阵

ex32矩阵来自于UF Sparse Matrix Collection的FIDAP组[13].]。虽然该矩阵阶数为1 159，，非零元个数为11 047，，但仍为对称不定矩阵. ，如果不用预处理,，SQMR方法不能在400步之内收敛。

表2　ex32矩阵

α	$\tau\tau=10^{-4}$	$\tau\tau=10^{-5}$	$\tau\tau=10^{-6}$

	nz(***L***)	its	*nz*(***L***)	its	*nz*(***L***)	its
0.5	31 907	12	351 983	7	35 815	4
0.1	27 213	12	27 756	6	27 800	5
0.05	27 056	11	27 566	7	27 605	5
0.01	27 062	11	27 243	6	27 255	5

SQMR方法不能在400步之内收敛.

表2记录了SQMR在PMIC(τ)预处理下的收敛情况．。从表2可以看出，，PMIC(τ)是处理对称不定矩阵非常有效的预处理算法，，其中的非零元抛弃参数τ仍然是影响其预处理效率的主要因素.。

4 5 结 论

本文研究了求解对称不定线性系统的一类不定预处理算法．。该算法的预处理子通过RBBK选主元的MIC来构造，，由于选主元原系数矩阵$\boldsymbol{A}$被重排为$\boldsymbol{PAP}^{\mathrm{T}}$． $\boldsymbol{PAP}^{\mathrm{T}}$。给出的数值算例体现了算法的有效性，，不过PMIC算法中的两个参数α和τ的最优值不易从理论上给出，，但是从数值例子也可发现，，合适的参数值是很容易找到的.。

参 考 文 献

[1][1]AASEN J O. On the reduction of a symmetric matrix to tridiagonal form[J]. BIT, 1971, 11: 233-242.

[2][2]BUNCH J R, KAUFMAN L. Some stable methods for calculating inertia and solving symmetric linear systems[J]. Math Comp, 1977, 31: 163-179.

[3][3]BUNCH J R, PARLETT B N. Direct methods for solving symmetric indefinite systems of linear equations[J]. SIAM J Numerical Analysis, 1971, 8: 639-655.

[4][4]ASHCRAFT C, GRIMES R G, LEWIS J G. Accurate symmetric indefinite linear equation solvers[J]. SIAM J Matrix Analysis and Applications, 1998, 20: 513-561.

[5][5]HUANG Ting-zhu, LI Liang. Relaxed forms of BBK algorithm and FBP algorithm for symmetric indefinite linear systems[J]. Computers and Mathematics with Applications, 2008, 55: 801-807.

[6][6]PAIGE C C, SAUNDERS M A. Solution of sparse indefinite systems of linear equations[J]. SIAM J Numerical Analysis, 1975, 12: 617-629.

[7][7]FREUNDAND R W, NACHTIGAL N M. A new Krylov-subspace method for symmetric indefinite linear systems[C]//Proceedings of the 14th IMACS World Congress on Computational and Applied Mathematics. Atlanta: [s.n.], 1994: 1253-1256.

[8][8]CHOW E, SAAD Y. Experimental study of ILU preconditioners for indefinite matrices[J]. J Comp Application Math, 1997, 86: 387-414.

[9][9]SAAD Y. ILUT: a dual threshold incomplete LU factorization[J]. Numerical Linear Algebra with Applications, 1994, 1: 387-402.

[10][10] CHENG S H, HIGHAM N J. A modified Cholesky algorithm based on a symmetric indefinite factorization[J]. SIAM J Matrix Analysis and Applications, 1998, 19: 1097-1110.

[11][11] GOLUB G H, VAN LOAN C F. Matrix computations[M]. Baltimore: Johns Hopkins University Press, 1996.

[12][12] SAAD Y. Iterative methods for sparse linear systems[M]. Boston: PWS Publishing Company, 1996.

[13] DAVIS T A. UF Sparse matrix collection[DB/OL]. [2008-10-12]. http://www.cise.ufl.edu/research/sparse/matrices/.

编 辑 张 俊