交替 LU 分裂算法及其在 CFD 中的应用

相 倩 吴颂平 徐 悦

(北京航空航天大学 航空科学与工程学院, 北京 100191) (中国航空研究院 航空数值模拟技术研究应用中心, 北京 100012)

摘 要: 在 CFD(Computational Fluid Dynamics)时间相关算法中,为了保证计算的稳定性,时间步长的取值通常会很小,这将导致计算过程收敛缓慢. 针对这一问题,提出了一种新的迭代算法—交替 LU 分裂(ALUS,Alternating Lower-Upper Splitting)算法,可以有效加速收敛,提高计算效率. ALUS 算法将系数矩阵分裂成上、下三角矩阵,因此仅需要利用追赶法求解两个三角矩阵,计算量较小,容易实现. 给出了 ALUS 算法收敛的定理,并且通过线性问题以及 CFD 圆柱绕流的数值模拟对 ALUS 算法进行了检验. 理论分析和数值实验的结果均表明: ALUS 算法计算量小,大大节省了计算时间,而且该算法是鲁棒的. 因此 ALUS 算法是高效的、稳定的算法,适用于 CFD 数值模拟.

关 键 词: 数值模拟; 大规模线性系统; 正定矩阵; 鲁棒性

中图分类号: V 211.3

文献标识码: A **文 章 编 号**: 1001-5965(2012)07-0953-04

Alternating lower-upper splitting iterative method and its application in CFD

Xiang Qian Wu Songping

(School of Aeronautic Science and Engineering, Beijing University of Aeronautics and Astronautics, Beijing 100191, China)

Xu Yue

(Numerical Simulation Center, Chinese Aeronautical Establishment, Beijing 100012, China)

Abstract: Small time step is used to ensure convergence in the time-dependent method in computational fluid dynamics(CFD). An improved iteration method termed as the alternating lower-upper splitting (ALUS) iterative method was proposed to address the problem, in which the coefficient matrix was split into a lower and an upper triangular matrix. In each inner step, only two triangular matrices were solved by Thom asalgorithm, therefore the ALUS method is simple. Theorems were listed to ensure the ALUS method convergence. A linear equation problem and flow around the cylinder were used to illustrate the characteristic of the ALUS method. Theoretical analysis and numerical results both demonstrate the new method performs well for positive definite matrices. Withless amount of computational work, the CPU time can be greatlydiminished. Thus the new ALUS method is efficient and robust and it is applicable in CFD numerical simulation.

Key words: numerical simulation; large scale linear systems; positive definite matrix; robustness

随着 CFD(Computational Fluid Dynamics)模拟能力的提高,计算中需要求解的方程组的规模也急剧增加. 快速求解大型方程组已经成为 CFD 计算的瓶颈之一[1]. 迭代算法是求解大型方程组的有效方法,其收敛速度依赖于方程组系数矩阵的性质. 针对超声速流动离散方程组系数矩阵往往是本质正定的这一特点,本文提出了一种迭代算法,称为交替 LU 分裂(ALUS,Alternating Lower-Upper Splitting)算法. 该算法计算量小,易于实现,且具有很好的鲁棒性,可大大节省计算时间.

收稿日期: 2011-06-26; **网络出版时间**: 2012-06-29 20:22

网络出版地址: www.cnki.net/kcms/detail/11.2625.V.20120629.2022.015.html

基金项目: 国家自然科学基金资助项目(10904178)

作者简介: 相 倩(1984 -),女,河北新乐人,博士生,axg220377@gmail.com.

1 迭代算法

常用的迭代算法分为两类. 一类是基于系数矩阵分裂的古典迭代方法,例如 Jacobi 方法,Gauss-Seidel 方法,松弛法(SOR,Successive Over Relaxation)等. 其中交替方向算法因其简单容易实现,在实际计算中有着广泛的应用[2]. 近些年逐步发展起来的 HSS(Hermitian and skew-Hermitian Splitting)方法[3]和 PSS(Positive definite and skew-Hermitian Splitting)方法[4]及其扩展算法也可以看作是广义的交替方向算法,具体参见文献[5]. 另一类算法是基于变分极值原理的迭代算法,其中最具代表性的有适用于对称正定系统的共轭梯度法(CG,Conjugate Gradient)以及适用于非对称正定的广义极小残量法(GMRES,Generalized Minimal Residual).

1.1 交替方向算法

设矩阵 $\boldsymbol{A}\in\mathbf{C}^{n\times n}$,考虑线性方程组

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

若 $\boldsymbol{A}$ 可以分解成 $\boldsymbol{A}=\boldsymbol{Q}+\boldsymbol{S}$,则迭代算法

$$\left.\begin{aligned}(\alpha\boldsymbol{I}+\boldsymbol{Q})x^{k+\frac{1}{2}} &= (\alpha\boldsymbol{I}-\boldsymbol{S})x^{k}+b\\ (\alpha\boldsymbol{I}+\boldsymbol{S})x^{k+1} &= (\alpha\boldsymbol{I}-\boldsymbol{Q})x^{k+\frac{1}{2}}+b\end{aligned}\right\} \tag{2}$$

称为交替方向算法. 该算法亦可以写成

$$\boldsymbol{M}x^{k+1} = \boldsymbol{F}x^{k}+b \tag{3}$$

其中 $\boldsymbol{M} = (\alpha\boldsymbol{I}+\boldsymbol{Q})(\alpha\boldsymbol{I}+\boldsymbol{S})/(2\alpha)$

$\boldsymbol{F} = (\alpha\boldsymbol{I}-\boldsymbol{Q})(\alpha\boldsymbol{I}-\boldsymbol{S})/(2\alpha)$

并且 $\boldsymbol{A}=\boldsymbol{M}-\boldsymbol{F}$. 由式(3)可知,矩阵 $\boldsymbol{M}$ 可以看成是一个预处理算子. 一个好的预处理算子 $\boldsymbol{M}$ 应该构造简单易于求逆,并且谱半径 $\rho(\boldsymbol{M}^{-1}\boldsymbol{F})$ 应该尽可能小.

对于交替方向算法式(2),有以下定理成立.

定理 1 对任意的正实数 $\alpha>0$,若 $\alpha\boldsymbol{I}+\boldsymbol{Q}$ 和 $\alpha\boldsymbol{I}+\boldsymbol{S}$ 都是可逆矩阵,迭代过程式(2)收敛的充要条件是:$\rho[(\alpha\boldsymbol{I}-\boldsymbol{Q})(\alpha\boldsymbol{I}+\boldsymbol{Q})^{-1}(\alpha\boldsymbol{I}-\boldsymbol{S})\cdot(\alpha\boldsymbol{I}+\boldsymbol{S})^{-1}]<1$,并且倘若迭代过程式(2)收敛,则必收敛到方程组(1)的解.

1.2 HSS 算法和 PSS 算法

任一正定矩阵矩阵 $\boldsymbol{A}\in\mathbf{C}^{n\times n}$ 均可以分解成 Hermite 矩阵和反 Hermite 矩阵之和,$\boldsymbol{A}=\boldsymbol{H}+\boldsymbol{S}$,其中,$\boldsymbol{H}=\frac{1}{2}(\boldsymbol{A}+\boldsymbol{A}^{\mathrm{H}})$,$\boldsymbol{S}=\frac{1}{2}(\boldsymbol{A}-\boldsymbol{A}^{\mathrm{H}})$.

若 $\boldsymbol{D}$、$\boldsymbol{L}$ 和 $\boldsymbol{U}$ 分别代表矩阵 $\boldsymbol{A}$ 的对角线、严格下三角和严格上三角部分,则 $\boldsymbol{A}$ 还可以分解成下三角矩阵和反 Hermite 矩阵之和,$\boldsymbol{A}=\boldsymbol{P}+\tilde{\boldsymbol{S}}$,其中,$\boldsymbol{P}=\boldsymbol{D}+\boldsymbol{L}+\boldsymbol{U}^{\mathrm{H}}$,$\tilde{\boldsymbol{S}}=\boldsymbol{U}-\boldsymbol{U}^{\mathrm{H}}$.

引理 1 设矩阵 $\boldsymbol{P}\in C^{n\times n}$. 如果 $\boldsymbol{P}+\boldsymbol{P}^{\mathrm{H}}$ 正定,则对任意的实数 $\alpha>0$,当 $\alpha\boldsymbol{I}+\boldsymbol{P}$ 可逆时,$\|(\alpha\boldsymbol{I}-\boldsymbol{P})(\alpha\boldsymbol{I}+\boldsymbol{P})^{-1}\|_2<1$.

引理 2 设矩阵 $\boldsymbol{S}\in C^{n\times n}$ 是反 Hermite 矩阵,则 $\|(\alpha\boldsymbol{I}-\boldsymbol{S})(\alpha\boldsymbol{I}+\boldsymbol{S})^{-1}\|_2=1$,$\forall\alpha>0$.

定理 2 若 $\boldsymbol{H}+\boldsymbol{H}^{\mathrm{H}}$ 正定,则对任意实数 $\alpha>0$,迭代算法

$$\left.\begin{aligned}(\alpha\boldsymbol{I}+\boldsymbol{H})x^{k+\frac{1}{2}} &= (\alpha\boldsymbol{I}-\boldsymbol{S})x^{k}+b\\ (\alpha\boldsymbol{I}+\boldsymbol{S})x^{k+1} &= (\alpha\boldsymbol{I}-\boldsymbol{H})x^{k+\frac{1}{2}}+b\end{aligned}\right\} \tag{4}$$

是收敛的,则称为 HSS 迭代算法.

定理 3 若 $\boldsymbol{P}+\boldsymbol{P}^{\mathrm{H}}$ 正定,则对任意实数 $\alpha>0$,迭代算法

$$\left.\begin{aligned}(\alpha\boldsymbol{I}+\boldsymbol{P})x^{k+\frac{1}{2}} &= (\alpha\boldsymbol{I}-\tilde{\boldsymbol{S}})x^{k}+b\\ (\alpha\boldsymbol{I}+\tilde{\boldsymbol{S}})x^{k+1} &= (\alpha\boldsymbol{I}-\boldsymbol{P})x^{k+\frac{1}{2}}+b\end{aligned}\right\} \tag{5}$$

是收敛的,称其为 PSS 迭代算法.

共轭斜量法是求解对称正定线性方程组的有效方法,消去法是求解下三角线性方程组的有效方法. 文献[6]充分利用反对称矩阵 $\boldsymbol{S}$ 的特点,给出了快速求解带位移的反对称问题 $(\alpha\boldsymbol{I}+\boldsymbol{S})\boldsymbol{y}=\tilde{b}$ 的 SSS(Solving a Shifted Skew-Symmetric Systems)算法. 将这些求解方法结合起来,使得 HSS 迭代算法和 PSS 迭代算法成为有效的广义交替方向迭代算法[7-8]. 尽管如此,HSS 算法和 PSS 算法仍然存在很多问题,例如:当矩阵 $\boldsymbol{A}$ 的阶数为奇数时,反对称矩阵 $\boldsymbol{S}$ 是奇异的,而参数 α 的取值通常比较小,这将导致计算不稳定;无论是 HSS 还是 PSS 方法都需要求解带位移的反对称方程组,但上述 SSS 算法比较复杂计算缓慢. 为此,本文提出了交替 LU 分裂算法.

1.3 ALUS 算法

将对角矩阵 $\boldsymbol{D}$ 分解成 $\boldsymbol{D}_1$,$\boldsymbol{D}_2$ 两部分,则矩阵 $\boldsymbol{A}$ 可分裂成下三角矩阵 $\tilde{\boldsymbol{L}}=\boldsymbol{D}_1+\boldsymbol{L}$ 与上三角矩阵 $\tilde{\boldsymbol{U}}=\boldsymbol{D}_1+\boldsymbol{U}$ 之和,$\boldsymbol{A}=\tilde{\boldsymbol{L}}+\tilde{\boldsymbol{U}}$. 于是由引理 1,可得到以下结论.

定理 4 若 $\tilde{\boldsymbol{L}}+\tilde{\boldsymbol{L}}^{\mathrm{H}}$ 和 $\tilde{\boldsymbol{U}}+\tilde{\boldsymbol{U}}^{\mathrm{H}}$ 均正定,则对任意的实数 $\alpha>0$,迭代算法

$$\left.\begin{aligned}(\alpha\boldsymbol{I}+\tilde{\boldsymbol{L}})x^{k+\frac{1}{2}} &= (\alpha\boldsymbol{I}-\tilde{\boldsymbol{U}})x^{k}+b\\ (\alpha\boldsymbol{I}+\tilde{\boldsymbol{U}})x^{k+1} &= (\alpha\boldsymbol{I}-\tilde{\boldsymbol{L}})x^{k+\frac{1}{2}}+b\end{aligned}\right\} \tag{6}$$

是收敛的,本文称为交替 LU 分裂算法.

按照式(3),记

$$\boldsymbol{M}_{\mathrm{ALUS}} = (\alpha\boldsymbol{I} + \tilde{\boldsymbol{L}})(\alpha\boldsymbol{I} + \tilde{\boldsymbol{U}})/(2\alpha)$$

$$\boldsymbol{F}_{\mathrm{ALUS}} = (\alpha\boldsymbol{I} - \tilde{\boldsymbol{L}})(\alpha\boldsymbol{I} - \tilde{\boldsymbol{U}})/(2\alpha)$$

则有

定理 5　若 $\boldsymbol{A}+\boldsymbol{A}^{\mathrm{H}}$ 和 $\boldsymbol{M}_{\mathrm{ALUS}}+\boldsymbol{M}_{\mathrm{ALUS}}^{\mathrm{H}}$ 正定,并且 $\inf\limits_{y\neq 0}\left\{\mathrm{Real}\left(\dfrac{\boldsymbol{y}^{\mathrm{H}}\boldsymbol{F}_{\mathrm{ALUS}}\boldsymbol{y}}{\boldsymbol{y}^{\mathrm{H}}\boldsymbol{A}\boldsymbol{y}}\right)\right\} > -1/2$,则 ALUS 算法收敛. ALUS 算法,不必借助 SSS 算法,只需要直接求解两个三角方程组,计算十分简单. 特别是当矩阵 $\boldsymbol{A}$ 按行、按列严格对角占优,并且 $\boldsymbol{A}$ 的对角线元素全都是正实数,即

$$a_{ii} > 0, \sum_{j\neq i}|a_{ij}| < a_{ii} \qquad \sum_{j\neq i}|a_{ji}| < a_{ii}$$

则可将对角线元素可分解成 $a_{ii}=a_{ii}^{(1)}+a_{ii}^{(2)}$,使得

$$\sum_{j<i}|a_{ij}| < a_{ii}^{(1)} \qquad \sum_{j>i}|a_{ij}| < a_{ii}^{(2)}$$

还可以将对角线元素分解成 $a_{ii}=a_{ii}^{(3)}+a_{ii}^{(4)}$,使得

$$\sum_{j<i}|a_{ji}| < a_{ii}^{(3)} \qquad \sum_{j>i}|a_{ji}| < a_{ii}^{(3)}$$

于是,以 $(a_{ii}^{(1)}+a_{ii}^{(4)})/2$ 为对角线元素构成对角矩阵 $\boldsymbol{D}_1$,以 $(a_{ii}^{(2)}+a_{ii}^{(3)})/2$ 为对角线元素构成对角矩阵 $\boldsymbol{D}_2$,就可以保证矩阵 $\tilde{\boldsymbol{L}}$ 和 $\tilde{\boldsymbol{U}}$ 都是对角占优的,从而 $\tilde{\boldsymbol{L}}+\tilde{\boldsymbol{L}}^{\mathrm{H}}$ 和 $\tilde{\boldsymbol{U}}+\tilde{\boldsymbol{U}}^{\mathrm{H}}$ 都是正定的,此时,交替 LU 迭代算法就是收敛的.

2　数值实验

计算效率以及迭代矩阵的谱半径是衡量迭代算法优劣的重要标准. 通过以下算例对本文的 ALUS 算法进行评估.

2.1　线性问题

在区域 $D=\{0\leqslant x,y\leqslant 1\}$ 上考虑边值问题

$$\left.\begin{aligned} -\Delta u + p\left(\frac{\partial u}{\partial x}+\frac{\partial u}{\partial y}\right) &= f(x,y) \\ u|_{\partial D} &= \mathrm{e}^{2x+y}+2x^2+y+1 \end{aligned}\right\} \qquad (7)$$

其中,$p<0$;$f(x,y)=(3p-5)\mathrm{e}^{2x+y}+4xp+p-4$,该问题的解析解为 $u=\mathrm{e}^{2x+y}+2x^2+y+1$.

取自然数 N,令 $h=1/N$,采用均匀网格 $\Delta x=\Delta y=h$ 进行离散区域 D,考虑式(7)的差分近似

$$-\frac{u_{i+1,j}-2u_{i,j}+u_{i-1,j}}{h^2}-\frac{u_{i,j+1}-2u_{i,j}+u_{i,j-1}}{h^2}+p\frac{u_{i+1,j}-u_{i,j}}{h}+p\frac{u_{i,j+1}-u_{i,j}}{h}=f_{i,j} \qquad (8)$$

用 ALUS 算法求解方程组(8),初始近似取零向量,迭代的收敛标准是残差小于 10^{-6}.

参数 p 决定了线性方程组的系数矩阵性质. $|p|$ 越小,系数矩阵越接近对称矩阵;$|p|$ 越大,系数矩阵的非对称性越强. 参数 N 决定了系数矩阵的规模. 参数 α 决定了迭代矩阵的谱分布,影响了迭代的收敛速度,记能使迭代步数最少的 α 值为 α^*,通过数值实验,表 1 给出了不同 p 和 N 下 α^* 的取值.

表 1　α^* 的取值

$-p$	$N=8$	$N=16$	$N=32$	$N=64$	$N=128$	$N=256$
0	0.65	0.36	0.16	0.079	0.04	0.019
1	0.65	0.33	0.15	0.079	0.038	0.02
10	2.4	1.3	0.58	0.27	0.15	0.059
10^2	24	11	4.5	1.9	0.65	0.29
10^3	250	122	49	17	6	2.1
10^4	2498	1259	470	180	65	23
10^5	25000	12500	4900	1800	700	200

尽管不能给出 α^* 的精确表达式,但从表 1 中可以看出,当 $|p|\geqslant 10$ 时,α^* 与 $|p|$ 近似成正比,与 N 成反比,可用经验公式表示为 $\alpha^*\approx 2|p|/N$,如图 1 所示.

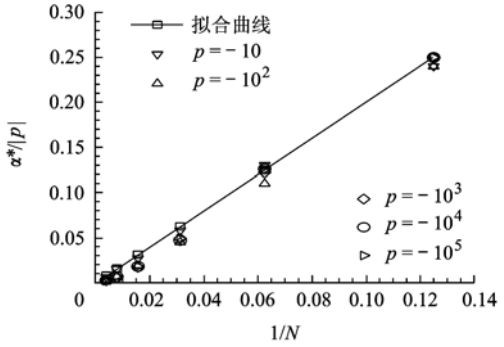


图 1　α^* 与 p、N 关系图

表 2 和表 3 给出了不同 p、N 和 α 下,迭代矩阵的谱半径 $\rho(\boldsymbol{M}_{\mathrm{ALUS}}^{-1}\boldsymbol{F}_{\mathrm{ALUS}})$. 由表 2 和表 3 可以看到,参数 α 可以在很大范围内取值,这表明 ALUS 算法具有很好的鲁棒性. 表 4 给出了 $p=-1$,$N=100$ 时,GMRES 算法,HSS 算法、PSS 算法以及 ALUS 算法收敛所需的 CPU 时间及迭代步数. 由表 4 可知,本文的新算法,计算所需 CPU 时间较小,收敛快是一种高效迭代算法.

表 2　$p=-10$,$N=32$ 时,迭代矩阵谱半径

α	0.06	0.58	1.0	6.0
谱半径	0.9474	0.7360	0.8699	0.9791

表 3　$p=-10^3$,$N=32$ 时,迭代矩阵谱半径

α	0.5	5	49	500
谱半径	0.6595	0.5351	0.0916	0.7795

表 4　不同迭代算法的比较

迭代算法	CPU 时间/s	迭代步数
ALUS(0.049)	2.53	290
HSS(0.612)	15.26	2231
PSS(0.047)	5.18	824
GMRES(10)	6.76	11

2.2 二维超声速圆柱绕流

将 ALUS 算法应用于超音速圆柱绕流的 CFD 计算中. 算例中,来流马赫数$Ma=8$,雷诺数 $Re=1.59\times10^5$.

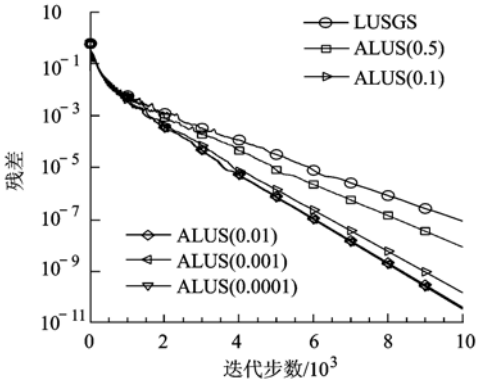
在计算坐标系下,二维 Navier-Stokes 方程的形式可以表示为

$$\frac{\partial Q}{\partial t}+\frac{\partial F}{\partial \xi}+\frac{\partial G}{\partial \eta}=\frac{1}{Re}\left(\frac{\partial F_v}{\partial \xi}+\frac{\partial G_v}{\partial \eta}\right) \tag{9}$$

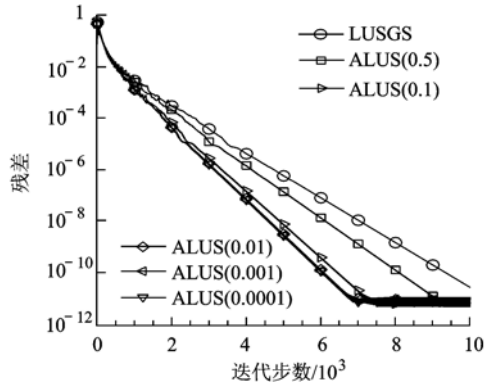
式中的粘性通量采用中心差分,无粘通量采用 Harten-Yee 的 TVD(Total Variation Diminishing)格式,时间推进则采用了新的 ALUS 算法. 计算网格为结构网格,大小为 100×80. 迭代的收敛标准是残差小于 10^{-8}.

表 5 给出了不同参数 α 下,ALUS 算法的迭代步数以及 CPU 计算时间,并与经典的 LU-SGS (Lower-Upper Symmeffic Gauss-seidel)算法进行对比. 可以看出,本文的 ALUS 算法较经典的 LU-SGS 算法收敛的更快,计算时间甚至可以节约 20% 以上;而且 α 在很广的范围内,迭代都是收敛的.

为了进一步分析 LU 分裂算法的计算效率,图 2 给出了 CFL 值分别为 1.3 以及 2.4 时,ALUS 算法以及 LU-SGS 算法的收敛史.



a CFL 数为 1.3



b CFL 数为 2.4

图 2 不同 CFL 数下的收敛史

表 5 ALUS 算法与 LU-SGS 算法的比较

迭代算法	迭代步数	CPU 时间/s
LU-SGS	15977	530.01
ALUS(0.1)	10246	420.33
ALUS(0.01)	9568	392.81
ALUS(0.001)	9500	389.67
ALUS(0.0001)	9492	389.33

数值实验的结果表明,本文给出的 ALUS 算法是一种效率高、鲁棒性好的迭代算法,可用于 CFD 的计算.

3 结 论

对于求解大型方程组的交替方向算法,提出了新的 ALUS 算法. 该算法在每步迭代中只需要求解两个三角方程组,计算量很小. 与其他同类迭代算法相比,收敛得更快,因此是一种高效的迭代算法. 数值实验还表明,算法中的参数 α 在很广的范围内取值,迭代均能收敛,可见新算法鲁棒性很好. 二维超声速圆柱绕流的算例则表明,ALUS 算法可用于 CFD 计算.

参考文献 (References)

[1] 李良. 大型线性方程组求解技术及在计算电磁学中的应用研究[D]. 成都:电子科技大学数学科学学院,2009
Li Liang. Study of solutions to large linear systems with applications in computational electromagnetic [D]. Chengdu: School of Mathematical, University of Electronic Science and Technology, Jr of China, 2009 (in Chinese)

[2] Peaceman D W, Rachford H H, Jr. The numerical solution of parabolic and elliptic differential equations [J]. J Soc Indust and Appl Math, 1955, 3(1): 28 – 41

[3] Bai Z Z, Golub G H, Nq M K. Hermitian andskew-Hermitian splitting methods for non-Hermitian positive definite linear systems[J]. J Matrix Anal and Appl, 2003, 24(3): 603 – 626

[4] Bai Z Z, Golub G H, Lu L Z, et al. Block triangular and skew-Hermitian splitting methods forpositive-definite linear systems [J]. J Sci Comput, 2006, 26(3): 844 – 863

[5] 蒋尔雄. 矩阵计算[M]. 北京:科学出版社, 2008: 104 – 120
Jiang Erxiong. Matrix computations[M]. Beijing: Science Press, 2008: 104 – 120 (in Chinesec)

[6] Jiang Erxiong. Algorithm for solving a shifted skew-symmetric linear system [J]. Front Math China, 2007, 2(2): 227 – 242

[7] Benzi M, Guo X P. A dimensional split preconditioner for Stokes and linearized Navier-Stokes equations [J]. Applied Numerical Mathematics, 2011, 61(1): 66 – 76

[8] Bai Z Z, Benzi M, Chen F. On preconditioned MHSS iteration methods forcomplex symmetric linear systems [J]. Numer Algorithms, 2011, 56(2): 297 – 317

(编 辑:李 晶)