大规模稀疏矩阵迭代解法研究报告

郭文韬，刘志强

2021 年 1 月 13 号

摘要

关键词：预条件子, 多级, 低秩修正, 舒尔补, 矩阵稀疏化, 图分割

引言

矩阵的求解（$Ax = b$）一般分为直接解法和迭代解法。对于小规模的矩阵，人们一般采用 LU 分解（对于一般稠密矩阵，当然也有选主元等增加算法稳定性的方法)，Choleskey 分解（对于对称正定矩阵）等方法，迭代解法更适用于大规模的稀疏矩阵。常用的迭代解法有 Jacobi 迭代，高斯-赛德尔迭代，逐次超松弛法（SOR 方法，以及用于对称矩阵的 SSOR)，预处理共轭梯度（PCG）等方法。[6]
迭代解法的收敛速度一般难以确保，经常需要人为添加“预条件子”来加快收敛速度。常见的预条件技术可以被分为三大类：

$M^{-1}Ax = M^{-1}b$，$AM^{-1}u = b, x = M^{-1}u$，$M_L^{-1}AM_R^{-1}u = M_L^{-1}b, x = M_R^{-1}u$。使用预条件子可以降低迭代矩阵的 2-条件数来提高迭代的收敛速度和解的准确性，在实际运用中经常可以和 PCG，GMRES 等方法有效结合。

传统的预条件子有 (块)Jacobi 预条件子，高斯-赛德尔预条件子，SOR(SSOR) 预条件子，也有给不完全 Choleskey 分解/不完全 LU 分解的预条件子。在迭代算法中，例如块 Jacobi 迭代适合于并行的实现，但其他的迭代算法未必适合。类似地，我们也希望预条件子的计算也尽可能支持并行。

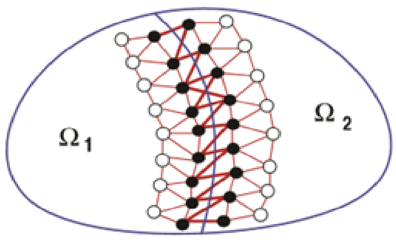
Y. Saad 提到 [5] 一些工业界支持并行的预条件子有 Schwarz，基于舒尔补的，以及多级 ILU 类的预条件子。在本论文，我们主要探讨了基于舒尔补的预条件子和多级低秩修正的 ILU/IC 预条件子。我们也采用了 domain-decomposition 技术 [3] 对矩阵进行划分边界/内部节点然后使用 metis 包做图切割

1 多级低秩修正预条件子方法（MLR）

1.1 图分割

我们知道块 **Jacobi** 预条件子 $M = \mathbf{diag}(A_{11}, \ldots, A_{nn})$ 易于并行，但假如对对角块做完全分解则消耗内存大，假如做不完全分解迭代步数还是会很高，但对于 $A = \begin{bmatrix} A_{11} & A_{12} \\ A_{21} & A_{22} \end{bmatrix}$，我们可以选 $B = \begin{bmatrix} A_{11} & \\ & A_{22} \end{bmatrix}$，然后做一个低秩近似使得 $M^{-1} = B^1 + LRC$

Figure 1: 图分割-对边切割 [3]



我们先对稀疏矩阵 **A** 用 **Metis[2]** 进行图切割（对边进行切割），然后将 A 转换成(1)的形式 [3]

$$\left(\begin{array}{cc|cc} \hat{B}_1 & \hat{F}_1 & & \\ \hat{F}_1^T & C_1 & & -W \\ \hline & & \hat{B}_2 & \hat{F}_2 \\ & -W^T & \hat{F}_2^T & C_2 \end{array}\right) \tag{1}$$

其中 $\hat{B}_i$ 和 C_i 对应区域 Ω_i 的内部/边界节点，$\hat{F}_i$ 对应内部-边界节点的连接。$W \in \mathbb{R}^{m_1 \times m_2}$ 对应区域 1 和区域 2 边界节点的连接

我们分解 W，使得 $W = X_1 X_2$，然后引入 $E = [0; X_1; 0; X_2^T]$

我们此时可以把 A 改写成以下形式 [3]

$$A = B - EE^T,\ B = \begin{pmatrix} B_1 & \\ & B_2 \end{pmatrix},\ B_i = \begin{pmatrix} \hat{B}_i & \hat{F}_i \\ \hat{F}_i^T & C_i + D_i \end{pmatrix} \tag{2}$$

其中，$D_1 = X_1 X_1^T \;\; D_2 = X_2^T X_2^T$

然后根据 **Sherman-Morrison** 公式，我们可以得到以下等式

$$A^{-1} = B^{-1} + B^{-1}E\underbrace{(I - E^T B^{-1} E)}_{X}{}^{-1} E^T B^{-1} \equiv B^{-1} + B^{-1} E X^{-1} E^T B^{-1} \tag{3}$$

我们可以对 $B^{-1}E$ 进行 **rank-k SVD** 分解来低秩近似 [3]

$$B^{-1}E \approx U_k V_k^T \qquad \text{rank-k 低秩近似}$$

最后，从(3)，(rank-k 低秩近似)我们可以写出以下预条件子[1]

$$M^{-1} = B^{-1} + U_k H_k U_k^T,\ H_k = (I - U_k^T E V_k)^{-1} \qquad \text{低秩修正预条件子}$$

1.2 多级方法

我们可以进一步扩展低秩修正预条件子到多级的情况。首先，我们可以将对角块 A_i 写成 $B_i - E_i E_i^T$，$B = \begin{pmatrix} B_{i_1} & \\ & B_{i_2} \end{pmatrix}$。我们可以进一步展开 A_i^{-1} 来得到 M_i^{-1}

$$\begin{aligned} M_i^{-1} \equiv A_i^{-1} &\approx B_i^{-1} + U_i H_i U_i^T \\ &= \begin{pmatrix} A_{i_1}^{-1} & \\ & A_{i_2}^{-1} \end{pmatrix} + U_i H_i U_i^T \\ &\approx \begin{pmatrix} M_{i_1}^{-1} & \\ & M_{i_2}^{-1} \end{pmatrix} + U_i H_i U_i^T \end{aligned} \tag{4}$$

我们最后可以得到以下多级递推式（MLR）

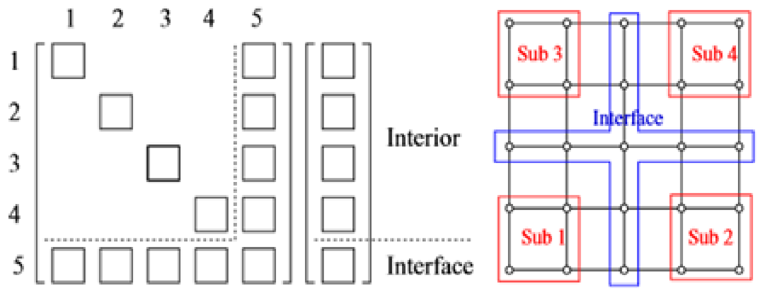
$$M_i^{-1} = \begin{cases} \begin{pmatrix} M_{i_1}^{-1} & \\ & M_{i_2}^{-1} \end{pmatrix} + U_i H_i U_i^T & \text{假如 i 不是叶子节点} \\ L_i^{-T} L_i^{-1} \text{ 或 } L_i^{-T} D_i^{-1} L_i^{-1} & \text{i 是叶子节点，做 IC/ILU 分解} \end{cases}$$

MLR 递归式 [3]

2 基于舒尔补的低秩修正预条件子（SLR）

对于矩阵 A，我们可以图切割（按边进行划分，方法和1的 B_i 的处理类似）

Figure 2: 按边划分矩阵 [4]



[1] 详细推导过程过程请见 [3] 第 2.2 和第 3 节

构造边界方程并求解

$$A\begin{bmatrix}x\\y\end{bmatrix}=\begin{bmatrix}f\\g\end{bmatrix}\quad A=\begin{bmatrix}B&E\\F&C\end{bmatrix}\tag{5}$$

$$\begin{cases}x=B^{-1}(f-Ey)\\(C-FB^{-1}E)y=g-FB^{-1}f\end{cases}\tag{6}$$

构造舒尔补 $S=C-FB^{-1}E$ 需要求解 s 次 B。我们此时需要构造一个 $\hat{S}$ 来近似 S，然后得到预条件子 $M=\begin{pmatrix}I&\\FB^{-1}&I\end{pmatrix}\begin{pmatrix}B&E\\&\hat{S}\end{pmatrix}$ 一种直接的做法是用 C^{-1} 来近似 S^{-1}，此时预条件子求解仅需求解 2 次 B 另一种做法是对 S 做低秩近似 (SLR)

$$S^{-1}\approx C^{-1}+LRC\tag{7}$$

这时我们可以引入矩阵 H 做近似谱分解 [4]

$$H=L^{-1}E^TB^{-1}EL^{-T}=UDU^T,\quad U\text{为酉矩阵},\ D=\mathrm{diag}(\lambda_1,\ldots,\lambda_s)\tag{8}$$

舒尔补 S 满足

$$S=L(I-H)L^T\tag{9}$$

然后我们可以近似 S^{-1}[4]

$$\begin{aligned}S^{-1}&=L^{-T}(I-H)^{-1}L^{-1}\\&=L^{-T}U(I-D)^{-1}U^TL^{-1}\\&\approx L^{-T}U(I-\tilde{D})^{-1}U^TL^{-1},\quad \tilde{D}=\mathrm{diag}(\tilde{\lambda}_1,\ldots,\tilde{\lambda}_s)\\&\approx C^{-1}+L^{-T}U[(I-\tilde{D})^{-1}-I]U^TL^{-1}\end{aligned}\tag{10}$$

R. Li, Y. Xi 和 Y. Saad 在 [4] 中提出了以下几种近似谱分解的做法，两种做法所带来的 $S\tilde{S}^{-1}$ 的谱范数不同

- $$\tilde{\lambda}_i=\begin{cases}\lambda_i & \text{假如}i\le k\\0&\text{其他情况}\end{cases}$$
- $$\tilde{\lambda}_i=\begin{cases}1-(1-\lambda_i)/\epsilon & \text{假如}i\le k\\0&\text{其他情况}\end{cases}$$
- $$\tilde{\lambda}_i=\begin{cases}1 & \text{假如}i\le k\\1-\lambda_i&\text{其他情况}\end{cases}$$

但是我们最后采用的是用 Matlab 中 eigs 命令取前 k 个最大的特征值（我们测试用的 k 是 20）

3 非对称矩阵稀疏化

我们知道矩阵 A 可以分成对角矩阵 D，上对角/下对角矩阵 L, U

$$A = D + L + U \tag{11}$$

我们可以将上三角部分和下三角部分分别看成两个无向图，然后使用无向图稀疏化 [7]。 对于负数边，我们将其权值处理为一个极小的正数。 我们可以得到以下预条件子feGRASS 预条件子

$$M = D + \tilde{L} + \tilde{U} \tag{feGRASS 预条件子}$$

实验与讨论

因为不同的预条件子使用条件不同，我们分开用对称正定矩阵和一般非对称矩阵进行了测试，测试样例可以在SuiteSparse Matrix Collection可以找到 [1]

1 对称正定矩阵

我们用对称正定矩阵作为输入，用 PCG 求解速度/每行填入数量测试了六种预条件子方法 Ichol、feGRASS、Block Jacobi、DD type、MLR、SLR[2]

Table 1: 测试矩阵参数

	size	NNZ (百分比)	2-条件数
Dubcova2	65025 * 65025	1030225 (.024%)	1.04e+04
Dubcova3	146689 * 146689	3636643 (.017%)	1.15e+04
Ecology2	999999 * 999999	4995991 (.0005%)	6.66e+07
Thermal1	82654 * 82654	574458 (.0084%)	4.96e+05
Thermal2	1228045 * 1228045	8580313 (.00057%)	7.48e+06
Apache2	715176 * 715176	4817870 (.00094%)	5.32e+06

[2] Block Jacobi、DD type、MLR、SLR 方法都划分为 8 个子区域，子区域内部使用 Ichol

Figure 3: 对称正定矩阵实验结果

	IChol		feGRASS		BlockJacobi		DDPre		MLR		SLR	
	Fills	Iter(Itime)	Fills	Iter(Itime)	Fills	Iter(Itime)	Fills	Iter(Itime)	Fills	Iter(Itime)	Fills	Iter(Itime)
Dubcova2	10.7	38(0.27)	5.33	164(0.99)	9.37	53(0.39)	9.19	37(0.42)	9.35	58(0.65)	9.19	35(0.39)
Dubcova3	10.7	40(0.63)	7.96	140(2.09)	8.84	53(0.85)	8.94	38(0.89)	8.85	56(1.83)	8.94	35(0.81)
Ecology2	13.9	212(19.1)	6.32	70(5.73)	12.2	373(42.4)	12.2	263(39.2)	12.2	330(46.7)	12.2	229(34.0)
Thermal1	14.6	59(0.61)	4.76	47(0.31)	13.5	120(1.15)	13.2	88(1.46)	13.5	63(1.03)	13.2	50(0.74)
Thermal2	14.9	217(35.1)	6.16	57(5.29)	14.3	337(48.0)	14.3	234(54.0)	14.3	265(58.6)	14.3	212(49.9)
Apache2	13.0	126(8.39)	12.1	79(5.48)	13.4	206(14.0)	13.2	155(17.3)	13.3	170(18.5)	13.2	152(17.2)

经过测试，我们可以发现

- 相比于块-Jacobi 预条件子，低秩修正预条件子确实能降低迭代步数，但 MLR 是否能缩短迭代时间存疑
- SLR...
- 对于相对稠密(非零元百分比较高)的矩阵 (Dubcova2/Dubcova3)，feGRASS 技术迭代步数会显著高于其他预条件子，但对于更加稀疏的矩阵，feGRASS 技术的迭代速度会明显快于其他预条件子
- 相比于其他 5 种预条件子，feGRASS 可以明显减少填充元

2 一般非对称矩阵

对于 Ilu、feGRASS 方法，我们使用一般非对称矩阵作为输入源并使用 GMRES 求解 (restart = 10)

Figure 4: 一般非对称矩阵实验结果

		ILU				GraphSparsification			
Cases	Size	Ptime	Itime	Iters	Fill	Ptime	Itime	Iters	Fill
rajat31	4.7E6(2.0E7)	Zero diagonal				16.8	1.95	2	2.11
freescale1	3.4E6(1.7E7)	0.44	92.4	281	2.99	18.4	141	439	2.27
Asic_680k	6.8E5(2.6E6)	Zero diagonal				0.93	0.87	15	1.32
dc1	1.2E5(7.7E5)	7.52	0.84	102	3.78	2.43	0.17	19	2.06
dc2	1.2E5(7.7E5)	7.48	0.85	103	3.78	2.29	0.18	19	1.97
dc3	1.2E5(7.7E5)	7.43	-	-	3.78	2.38	4.13	568	2.02
trans4	1.2E5(7.5E5)	7.64	0.25	29	3.71	2.45	0.12	10	1.93
trans5	1.2E5(7.5E5)	7.50	2.42	300	3.71	2.60	0.17	18	1.94
transient	1.8E5(9.6E5)	Zero diagonal				1.82	6.03	508	2.12
memchip	2.7E6(1.3E7)	Zero diagonal				9.00	6.82	22	2.15
mc2depi	5.3E5(2.1E6)		-	-			-	-	

经过测试，我们可以发现

- 非对称矩阵稀疏化方法优于 ILU，但是存在着稀疏化以后奇异的问题
- MATLAB ILU 命令可能无法得到分解结果

总结

我们这一次期末 project 主要学习并比较了几种预条件子。针对对称正定矩阵的预条件子，我们学习了低秩修正技术，Schur 补修正技术，然后我们也对其性能做了一个初步的研究比较，但目前我们的实现还比较原始，未能实现完全并行计算。针对非对称矩阵，我们尝试了基于 feGRASS 的稀疏化方法并和 ILU 进行了性能比较，但目前结果显示效果一般

参考文献

[1] T.A. Davis and Y. Hu. The university of florida sparse matrix collection. *ACM Transactions on Mathematical Software*, 38(1), 2011.

[2] G. Karypis. Metis - serial graph partitioning and fill-reducing matrix ordering. https://github.com/KarypisLab/METIS.

[3] R. Li and Y. Saad. Divide and conquer low-rank preconditioners for symmetric matrices. *SIAM Journal on Scientific Computing*, 35(4):A2069–A2095, 2013.

[4] Y.Xi R. Li and Y. Saad. Schur complement-based domain decomposition preconditioners with low-rank corrections. *Numerical Linear Algebra with Applications*, 26:706–729, 2016.

[5] Y. Saad. A short course on: Preconditioned krylov subspace methods. https://www-users.cs.umn.edu/~saad/Calais/PREC.pdf.

[6] Y. Saad. *Iterative Methods for Sparse Linear Systems*. Society for Industrial and Applied Mathematics, 2003.

[7] W. Yu Z. Liu and Z.Feng. fegrass: Fast and effective graph spectral sparsification for scalable power grid analysis. *IEEE Transactions on Computer-Aided Design of Integrated Circuits and Systems*.