arXiv:2602.03239v1 [math.NA] 3 Feb 2026

用于 $AXB = C$ 的确定性与随机 Kaczmarz 方法及其在彩色图像中的应用

复原Wenli Wang[1], Duo Liu[1],
Gangrong Qu[1],Michiel E. Hochstenbach[2*]

[1]北京交通大学数学与统计学院，中国北京，邮编 100044。[2*]荷兰埃因霍温理工大学数学与计算机科学系，埃因霍温，邮编 5600MB。

通讯作者。电子邮箱：m.e.hochstenbach@tue.nl；其他作者：wenli.wang12@gmail.com；20118001@bjtu.edu.cn；grqu@bjtu.edu.cn；

摘要

我们研究求解相容线性矩阵方程的 Kaczmarz 型方法。首先，提出一种采用确定性循环行选择策略的块 Kaczmarz（BK）方法。假设相关系数矩阵具有满列秩或满行秩，我们推导了该 BK 方法一个循环的矩阵公式。此外，我们提出一种贪婪随机块 Kaczmarz（GRBK）方法，并进一步将其扩展为松弛变体（RGRBK）和确定性对应方法（MWRBK）。我们建立了所提方法的收敛性质。数值测试验证了理论结果，并将所提方法应用于彩色图像复原问题。

关键词： 确定性和随机化 Kaczmarz方法、相容线性矩阵方程、收敛、最小范数解、多右端项的 Kaczmarz 方法、彩色图像复原

数学学科分类： 15A06 , 15A24 , 65F10

Dedicated with love to Prof. ?ke Björck, for his beautiful character, friendship, and work on many topics, including some appearing in this paper.

1 引言

给定矩阵 $A \in \mathbb{R}^{m\times p}$、 $B \in \mathbb{R}^{q\times n}$ 和 $C \in \mathbb{R}^{m\times n}$，我们感兴趣的是求解 $X \in \mathbb{R}^{p\times q}$，使其满足相容矩阵方程

$$AXB = C. \tag{1.1}$$

该方程与工程中的众多重要应用密切相关，例如图像处理、稳定性分析、控制理论、回归分析、计算机视觉、图像处理、网络分析以及计算机辅助几何设计；参见例如 [29, 34, 35, 6, 31]。我们将研究的一个具体例子是一个

无噪声彩色图像复原的前向模型，可以写成

$$AXA_{\mathrm{c}}^{T}=C\,; \tag{1.2}$$

例如，参见 [35]。在（1.1）中， A 可以是方阵、超定（$m>p$）或欠定（$m<p$）； B也同样如此。在（1.2）的背景下，系数矩阵 A 通常是一个大型方阵，而 A_{c} 的大小为 3×3。我们将在第 5 节和6节的实验中考察所有这些情形。

求解（1.1）的“直接”方法在维数上的计算复杂度为立方级，当解不唯一或 C 中存在噪声时，可能并不适用。已有多种迭代算法被提出用于求解矩阵方程（1.1） [10, 8, 28,30]。这些方法的一个共同特点是，通常在每次迭代时都需要存储并访问整个系数矩阵。相反，我们研究行作用方法，这类方法在每次迭代中仅访问 A 的一行。行作用和列作用技术作为求解大规模线性系统的高效迭代算法，已受到广泛关注（例如， [17, 14, 1, 27]）

$$Ax=b. \tag{1.3}$$

这些方法根据行（有时也根据列）划分，将系统分解为多个子问题，这可能带来诸多优势，例如减少内存使用 [15]。此外，或许更重要的是，事实证明，即使存在噪声或数据不完整，行作用方法通常也能得到高质量的解（例如，参见 [21]）。对于我们的问题（1.1），由于矩阵B的存在，行作用方法可能更有优势，例如可以避免昂贵的矩阵-矩阵乘积。

经典的 Kaczmarz 方法是一种特殊的行作用方法，已被广泛用于求解线性系统（1.3） [18]。在第 k次迭代中，标准做法是按循环方式选择行索引 i_k 。随后，按照以下方式更新 Kaczmarz 方法中的向量：

$$x^{k+1}=x^{k}+\alpha\,\frac{b_{i_k}-A_{i_k,:}\,x^{k}}{\|A_{i_k,:}\|^{2}}\,A_{i_k,:}^{T},\quad i_k=(k \bmod m)+1. \tag{1.4}$$

这里，$A_{i,:}$ 表示 A的第 i行， b_i 表示 b的第 i个分量， $\|\cdot\|$ 表示 2-范数，而 $\alpha\in(0,\ 2)$是一个松弛参数。通过将当前估计向量 x^k 正交投影到超平面$\{x\mid\langle A_{i_k,:}^{T},\ x\rangle=b_{i_k}\}$上，可得 $\alpha=1$ 的更新公式。在医学成像（计算机断层扫描）领域，Kaczmarz 方法也称为代数重建技术 [16]。

相关工作。 针对 Kaczmarz 方法， 人们提出了许多变体。Strohmer 和 Vershynin [26], 不再采用传统的顺序方式，而是按照与行范数成比例的概率随机选择索引。这由此得到一种随机 Kaczmarz 方法（RK 方法）， 其期望收敛速度被不同领域称为指数收敛或线性收敛。随后，Bai and Wu [2]提出了一种基于加权残差的行选择策略。具体而言，该策略优先选择残差较大的行，与 [26]的方法相比，可能实现更快的收敛。他们将这种方法称为贪婪随机 Kaczmarz（GRK）方法。此外，还可以通过在行选择策略中加入参数 $\theta\in[0,1]$ 对 GRK 方法进行推广（更多详情见第 4.2 节）， 由此得到 [3]提出的松弛贪婪随机 Kaczmarz（RGRK）方法。Du 和 Gao [11] 考虑了极限情形 $\theta=1$，这是一种直接选择加权残差最大的行的确定性方法。他们给出了该方法收敛速度的理论估计，并将这一最初由 McCormick [20], 提出的方法称为最大加权残差 Kaczmarz（MWRK）方法。我们将把这些 GRK、RGRK 和 MWRK 方法推广到问题（1.1）。

现在，我们将重点关注 Kaczmarz 型方法在求解（1.1）方面的扩展。Wu、Liu 和 Zuo [31] 将 [3, 11] 中提出的 RGRK 和 MWRK 方法扩展到求解（1.1）， 并给出了收敛结果。在他们的方法中，原矩阵方程（1.1）被改写为 mn 个子系统 $A_{i,:}XB_{:,j}=C_{i,j}$。

每一步产生的迭代都会加上一个秩一矩阵。这里，$B_{:,j}$ 表示 B 的第 j 列，而 $C_{i,j}$ 是 C 中对应的元素。Niu 和 Zheng [23] 研究了类似的方法，其中矩阵方程（1.1）被分解为一系列形式为 $A_{I,:}XB_{:,J} = C_{I,J}$ 的子块问题；其中，$A_{I,:}$ 和 $B_{:,J}$ 分别表示 A 和 B 中由 I 和 J 索引的行、列子矩阵，而 $C_{I,J}$ 是 C 中对应的子矩阵。与我们的方法差异稍大的是，Du、Ruan 和 Sun[12] 考虑每一步更新 X 的一列。我们的方法和 [31], 中的方法一样，也采用秩一更新，但关键区别在于，[31] 中的方法每一步处理 C 的一个元素和 B 的一列，而我们利用 C 的一行以及整个 B。

与我们的方法更为接近的是，Xing、Bao 和 Li（见 [32] 和算法 1）将（1.1）分解为 m 子系统或行方程 $A_{i,:}XB = C_{i,:}$，并将 [26]中的 RK 方法扩展用于求解该矩阵方程。为了区分他们的方法（该方法使用整个 B）与参考文献 [31] 中的方法（该方法作用于 B的列），他们将其命名为随机块 Kaczmarz（RBK）方法。

我们希望强调，本文所提出方法中“块”一词的使用，是指存在 B，而不是指（同样常见的）对 A 的子块的伪逆进行处理这一含义（参见 [13, 9]）。不过，事实上这很自然，因为从与（1.1）相关的线性系统（$B^T \otimes A$）vec(X) = vec(C)的层面来看，我们处理的是 $B^T \otimes A_{i,:}$，这对应于算子 $B^T \otimes A$的行的一个子集（或小批量）。

关于应用于（众所周知但不同于（1.1）的）Sylvester 矩阵方程的 Kaczmarz 方法的其他推广，参见 [25]。

与其他方法的一些联系。随机梯度下降（SGD）方法（例如参见 [5, 22]）是一种形式如下的迭代优化算法

$$x^{k+1} = x^k - \alpha\, |I_k|^{-1} \sum_{i \in I_k} \nabla f_i(x^k), \tag{1.5}$$

其中，α 是学习率，I_k 是数据的随机子集，$\sum_{i \in I_k} \nabla f_i(x^k)$ 是根据 I_k 计算得到的梯度，$|I_k|$ 表示 I_k 中元素的数量。该方法已被广泛应用于机器学习 [4]。Needell、Srebro 和 Ward [22] 指出，[26] 中提出的随机 Kaczmarz 方法可以视为小批量大小为 1 的随机梯度下降方法（$|I_k| = 1$）。

Landweber 方法是通过最小化目标函数来求解线性系统（1.3）的经典梯度法 $\frac{1}{2}\|Ax - b\|^2$ [19]。这将导出如下迭代方案

$$x^{k+1} = x^k + \alpha\, A^T (b - A\, x^k),$$

其中 $A^T(b - A\,x^k)$ 是目标函数的负梯度，步长α 满足 $0 < \alpha < 2\,\|A\|^{-2}$。用于求解矩阵方程 (1.1) 的基于梯度的迭代算法（GI） [10],如文献所研究，定义如下

$$X^{k+1} = X^k + \alpha\, A^T (C - AX^k B)\, B^T,$$

其中 $A^T\ C - AX^k B\ B^T$ $\frac{1}{2}\|AXB - (\ )$ $C\|_F^2$ $0 < \alpha < 2\,\|A\|^{-2}\,\|B\|^{-2}$ 是目标函数的负梯度，步长满足。Landweber 方法和 GI 方法在每一步都使用整个矩阵。我们所有的方法也都可以由目标函数的梯度推导得到（参见表 1.1）。我们的方法不使用整个 A, 而是采用迭代方案，其中 $|I_k| = 1$ 在 (1.5) 中使用确定性或随机化的行选择策略。

我们的贡献。本研究扩展并分析了用于求解矩阵方程（1.1）的 Kaczmarz 型方法，涵盖确定性和随机 Kaczmarz 方法。我们首先提出并研究了随机 RBK 方法的（确定性）循环行选择变体 [32], ，并将其称为块 Kaczmarz（BK）方法。我们考察了一般的 BK 方法（算法 2），并证明该方法收敛到解 $A^+CB^+ + X^0 - A^+AX^0BB^+$。接下来，我们研究在系数矩阵 B满足特定约束条件时的 BK 方法：满列秩（算法 3）和满行秩（算法 4），其中 B 非奇异是一个特例。这些额外的

约束使得我们能够推导出 BK 方法完整循环的简洁优雅的矩阵公式。（在 BK 方法中，术语 循环和扫描可以互换使用，均指按顺序完整遍历所有行。）表 B 1.1概述了在 B 具有不同假设时的 BK 方法。如果系数矩阵 B 具有满列秩，则使用 QR 分解 $B = QR$，并根据相关的最小二乘函数推导梯度；更多详情请参见第 3 节。

表 1.1 在关于 B的各种假设下，线性矩阵方程及相应 BK 方法表述的概述。

	一般情形	B 满列秩	B 满行秩
矩阵方程	$AXB = C$	$AXQ = \widehat{C}$	$AX = \widetilde{C}$
右端项	C	$\widehat{C} = CR^{-1}$	$\widetilde{C} = CB^T(BB^T)^{-1}$
梯度形式	$A^T(C - AXB)B^T$	$A^T(\widehat{C} - AXQ)Q^T$	$A^T(\widetilde{C} - AX)$
Kaczmarz 步	$A_i^T(C_i - A_iXB)B^T$	$A_i^T(\widehat{C}_i - A_iXQ)Q^T$	$A_i^T(\widetilde{C}_i - A_iX)$
扫描公式	–	✓ （3.11）	✓ （3.14）
参考文献	第 3.1节（算法2）	第 3.2 节（算法 3）	第 3.3 节（算法 4）

接下来，我们以贪婪随机 Kaczmarz 方法 [2]为出发点。该方法专为线性系统（1.3）设计，并从加权残差较大的行中随机选取一行。我们将该方法扩展为贪婪随机块 Kaczmarz（GRBK，算法5）方法，以求解（1.1）。此外，我们提出并研究了松弛版本（RGRBK，算法6）以及最大加权残差块 Kaczmarz（MWRBK，算法7）方法，它们可视为 GRBK 的松弛与确定性变体。所有这些方法都是针对线性系统（1.3）的方法的扩展。RGRBK 包含一个可调参数 $\theta \in [0, 1]$。对于 θ 的两个特殊情形，我们将看到，当 $\theta = \frac{1}{2}$ 时，RGRBK 与 GRBK 一致；当 $\theta = 1$ 时，RGRBK 与 MWRBK 一致；另见表 4.2 和图 4.3。我们证明，对于零初始矩阵，GRBK、RGRBK 和 MWRBK 方法在期望意义下收敛到唯一最小范数解 A^+CB^+。我们还将在第 4 节证明，这三种方法的收敛因子上界均不超过 [32] 中 RBK 方法的收敛因子上界。

与本文相关的求解（1.1）和（1.3）的一些方法概述见表1.2。我们注意到，除 GI 方法外，（1.1）的所有方法在每次迭代中都进行秩一更新。

表 1.2 关于 求解 $Ax = b$ 和相关方法的概览 $AXB = C$。

行作用	$Ax = b$	$AXB = C$	$AXB = C$的备注
确定性 Kaczmarz 方法 [18]		BK（算法 2）	循环行作用
		BK（算法3）	B 满列秩
		BK（算法4）	B 满行秩
	MWRK [11, 20]	MWRK [31], MWRBK（算法 7）	“最大行”、RGRBK 用于 $\theta = 1$
随机化	RK [26]	RBK（算法 1） [32]	
	GRK [2]	GRBK（算法5）	针对 $\theta = \frac{1}{2}$ 的 RGRBK
	RGRK [3]	RGRK [31], RGRBK（算法 6）	涉及 $\theta \in [0, 1]$
所有行	Landweber [19]	GI [10]	梯度

本文其余部分安排如下。第2节介绍必要的记号与预备知识。第3 节介绍 BK 方法及其收敛分析。第4节研究 GRBK 方法及其松弛与确定性变体，并分析其收敛性质。第5 节给出一些数值示例，以展示理论结果，并展现各种方法的性能。第6 节探讨所提出方法在彩色图像复原中的一项特殊应用，最后在第7节作出总结。

2 记号与预备知识

对于矩阵 A，我们用 A^+、A^T、$r(A)$、range(A)、$\|A\|_F$、A_i^T、$:$、$\rho(A)$、$\sigma_{\min}(A)$ 和 tr(A) 分别表示 Moore–Penrose 伪逆、转置、排名、值域、Frobenius 范数、第 i 行的转置、谱半径、最小奇异值以及 A 的迹。矩阵 A 和 B 的内积与 Kronecker积分别记为 $\langle A, B\rangle$ 和 $A \otimes B$。拉伸算子定义为 vec(X) $=(x_1^T, \ldots, x_n^T)^T \in \mathbb{R}^{mn}$，其中矩阵 $X = (x_1, \ldots, x_n) \in \mathbb{R}^{m\times n}$。我们用 0 表示零矩阵或零向量。对于两个正整数 a 和 b，(a mod b) 表示 a 除以 b 所得的余数。对于正整数 m，令 $[m] = \{1, \ldots, m\}$。给定前 k 次迭代时的条件期望记为 $\mathbb{E}_k$，即 $\mathbb{E}_k[\cdot] = \mathbb{E}[\cdot \mid i_0, i_1, \ldots, i_{k-1}]$，其中 $i_m (m = 0, 1, \ldots, k-1)$ 是在第 m 次迭代中选取的行索引。

我们首先回顾矩阵方程（1.1）的一些众所周知的性质；例如，参见[24, 12]。矩阵方程（1.1）称为相容的，如果至少存在一个解 $X \in \mathbb{R}^{p\times q}$ 使得该方程成立；在这种情况下，通解为

$$X_{\mathrm{M}} = A^+CB^+ + U - A^+AUBB^+, \tag{2.1}$$

其中 $U \in \mathbb{R}^{p\times q}$ 为任意值。定义

$$X_* := A^+CB^+; \tag{2.2}$$

这是唯一最小范数解，即 $X_* = \arg\min \|X\|_F$。这是成立的$AXB=C$，因为可以验证 $\langle X_*, U - A^+AUBB^+\rangle = 0$，因此 $\|X_{\mathrm{M}}\|_F^2 = \|X_*\|_F^2 + \|U - A^+AUBB^+\|_F^2 \geq \|X_*\|_F^2$。注意，当 A 为满列秩时，$A^+ = (A^TA)^{-1}A^T$；当其为满行秩时，$A^+ = A^T(AA^T)^{-1}$；对 B也有类似结论。众所周知，矩阵方程 (1.1) 可以向量化为形式 ($B^T \otimes A$) vec(X) $=$ vec(C)。对相容矩阵方程 (1.1) 的 $B^T \otimes A$ 进行秩分析表明，只有当 A 为满列秩且 B 为满行秩时，A^+CB^+ 才是其唯一解；在其他情况下，它是唯一最小范数解。

Xing等人提出了一种用于 (1.1) 的随机行方法，见 [32], ，称为 RBK 方法，如算法 1所示。这将作为本文提出的若干新变体的出发点。对 A没有任何限制，唯一的要求是从矩阵方程 (1.1) 中省略零行。我们指出，本文研究的所有方法都使用 A的行，其思路是使 A 的大小至少与 B相同。如果并非如此，我们可以改用转置方程 $B^TX^TA^T = C^T$。

算法 1 RBK 方法 [32]

1 X^0：**输入**：初始猜测：**输出：** 矩阵方程 (1.1) 的近似

2 解 3: **对于** $k = 0, 1, 2, \ldots$ 直至收敛 **执行** 4: 以概率 $\mathbb{P}$选择 $i_k \in \{1, \ldots, m\}$（第 $= i_k$ 行）$= \|A_{i_k,:}\|^2 / \|A\|_F^2$ 5: 更新 $X^{k+1} = X^k + \frac{\alpha}{\|A_{i_k,:}\|^2} A_{i_k,:}^T$ (($C_{i_k,:} - A_{i_k,:} X^k B) B^T$)

6: **End for**

算法 1 的迭代循环概述了更新近似解 X^k 以求解 (1.1) 的过程。迭代会持续进行，直到近似解达到所需精度，详见第 5 节和第 6 节。算法 1 第 5 行的更新操作每次迭代复杂度为 $\mathcal{O}(q(p+n))$，即将列向量 $A_{i_k,:}^T$ 与行向量 $(C_{i_k,:} - A_{i_k,:} X^k B)\ B^T$ 的乘积相加，也就是一次秩一更新。请注意，由于第 5 行的分母中含有因子 $\|A_{i_k,:}\|^2$，该方法在对 A 和 C 的行进行缩放时保持不变（即，当应用于 $DAXB = DC$ 时，对于非奇异对角矩阵 D，结果不变）。已有证明表明，[32, 定理 1]指出，当 (1.1) 相容时，RBK 方法依期望收敛到最小范数解 X_*，如下述结果所示。

定理 2.1。 [32, 定理 1] 设矩阵方程 (1.1)（其中 $A \in \mathbb{R}^{m\times p}$、 $B \in \mathbb{R}^{q\times n}$ 和 $C \in \mathbb{R}^{m\times n}$ ）相容。若 $0 < \alpha < 2\,\|B\|^{-2}$，则由 RBK 从初始矩阵 $X^0 \in \mathbb{R}^{p\times q}$ 出发得到的序列 $\{X^k\}_{k=0}^{\infty}$，其中 $X^0_{:,j} \in \text{range}(A^T)$， $j = 1, \ldots, q$ 且 $(X^0_{i,:})^T \in \text{range}(B)$， $i = 1, \ldots, p$，依期望收敛到唯一最小范数解 X_*。此外，解误差满足

$$\mathbb{E}\left[\|X^k - X_*\|_F^2\right] \le \delta^k \cdot \|X^0 - X_*\|_F^2,$$

其中

$$\delta = \delta(\alpha) = 1 - \frac{2\alpha - \alpha^2\,\|B\|^2}{\|A\|_F^2}\,\sigma_{\min}^2(A)\,\sigma_{\min}^2(B). \tag{2.3}$$

注意， $X^0 = \mathbf{0}$ 的常见取法满足前一定理中的两个值域条件。这个零矩阵是 Kaczmarz 型方法的标准取法，我们也将在第 5 节和第 6节中采用它。此外，我们将比较定理 2.1 中的收敛因子 δ 与第 4节所讨论算法的收敛因子。表 4.1总结了它们的收敛因子。采用（标准和混合）条件数的常用记法 $\kappa(A) = \|A\|/\sigma_{\min}(A)$ 和 $\kappa_{F,2}(A) = \|A\|_F/\sigma_{\min}(A)$，可知当取 $\alpha = \|B\|^{-2}$ 时， δ 最小，此时

$$\delta = \delta(\|B\|^{-2}) = 1 - \kappa_{F,2}(A)^{-2}\,\kappa(B)^{-2}.$$

因此，与用于线性系统的随机 Kaczmarz 方法（1.3） [26]中的因子$1 - \kappa_{F\text{相比},\ 2}(A)^{-2}$，它更接近 1（因而效果较差）。

3 块 Kaczmarz 方法及其收敛性

受 RBK 方法（算法1）的启发，我们现在提出并研究一种用于求解（1.1）的确定性循环变体，称为块 Kaczmarz（BK）方法（这里回顾一下“块”一词的含义，如引言中所述）。在第3.1节中，我们考虑一般的 BK 方法（算法2），该方法对系数矩阵不施加任何限制。在第3.2 节和3.3节中，我们研究系数矩阵 B满足不同约束时的 BK 方法：满列秩（第3.2节，算法3），以及满行秩（第3.3节，算法4）。由此得到略为简化的迭代格式，并给出了完整扫描步骤的矩阵公式。

关于记号的说明：在这些方法中，我们需要区分单行迭代与完整扫描，即完整遍历所有 m行所表示的m 行作用步。扫描在定理 3.2 的证明以及算法 3 和 4中很重要。其他所有方法和结果都使用行作用。稍微扩展一下记号，采用以下约定会比较方便：

记号 X^k 用于表示单行操作的结果，而 X^s （以及 $X^{s,\ i}$）则用于表示通过扫描得到的迭代值。

3.1 一般 BK 方法

在算法 2中，我们给出了适用于任意大小 B 的 BK 方法伪代码。

算法 2 一般 BK 方法

1 X^0：**输入**：初始猜测：**输出：**矩阵方程（1.1）的近似解

2

3: **对于** $k = 0$，1，2，.....直至收敛 **执行**4 $i_k = k \quad m + 1$：

计算（取模）：计算 $X^{k+1} = X^k + \frac{\alpha\ 5}{\|A_{i_k,\ :}\|^2} A^T_{i_k,\ :}$ （ $(C_{i_k,\ :} - A_{i_k,\ :} X^k B)$

B^T）6: **结束循环** ————

算法1 和2 之间唯一的区别是行选择索引 i_k：算法1 根据概率 $\|A_{i_k,\ :}\|^2 / \|A\|_F^2$随机选择一个索引，而

算法2 按顺序扫描所有行。算法2 也可以解释为一种按行梯度下降方案，步长为 α，用于求解一个缩放后的矩阵方程

$$DAXB = DC, \tag{3.1}$$

其中 D 是对角矩阵 diag($\|A_{1,:}\|^{-1}, \ldots, \|A_{m,:}\|^{-1}$)。对于 i（3.1）的第 行，对应的最小二乘问题为

$$\min_X F_i(X) := \min_X \tfrac{1}{2} \|A_{i,:}\|^{-2} \ \|A_{i,:}XB - C_{i,:}\|^2.$$

可以验证，关于 X 的梯度为

$$\nabla F_i(X) = \|A_{i,:}\|^{-2} \ A_{i,:}^T \left(A_{i,:}XB - C_{i,:}\right) B^T.$$

与算法 1一样，算法 2的每次迭代计算成本为二次量级：$\mathcal{O}(q(p+n))$。

算法 2 也可以视为 Ding、Liu 和 Ding [10]提出的 GI 方法的行版本。与 [10],中的方法不同，算法 2 首先将矩阵方程 (1.1) 分解为 m 个形如 $A_{i,:}XB = C_{i,:}$ 的子系统，并在每次迭代中仅使用 A 的一行及 C中对应的一行来更新 X^k 。 [10] 的方法涉及矩阵-矩阵乘积，因此每一步的计算成本相对较高。可以验证，GI 方法的一次更新需要 min($\mathcal{O}(mq(p+n))$ $\mathcal{O}(pn(m+q))$) 次运算。（取最小值是因为可以选择先计算 AX 或 XB。）该方法的计算复杂度相对于大小为 m、n、p、q的量呈立方关系。相比之下，算法 2 的复杂度为二次关系，因此每次迭代的成本低得多。我们希望算法 2 的 m 步比 GI 方法的一步能取得更大进展。第 6 节给出的实验结果确实证实了这一点。

现在我们研究算法2的收敛性。在 [17], Jiang 和 Wang 研究了一种用于求解（1.3）的行作用方法的收敛性。我们将他们的 [17,定理 A。3]推广到（1.1），并证明算法2 收敛到 X_*^0，其定义如下

$$X_*^0 := X_* + X^0 - A^+AX^0BB^+, \tag{3.2}$$

其中 X_* 与（2.2）中的定义相同。以下引理在随后的收敛分析中起重要作用。

引理3.1。由任意初始猜测 X^k 经算法2 得到的迭代矩阵 X^0 满足

$$X^k - A^+AX^kBB^+ = X^0 - A^+AX^0BB^+, \quad k = 0, 1, 2, \ldots .$$

证明。由算法2可得

$$X^k = X^0 + \sum_{j=0}^{k-1} \tfrac{\alpha}{\|A_{i_j,:}\|^2} A_{i_j,:}^T \left(C_{i_j,:} - A_{i_j,:}X^jB\right) B^T. \tag{3.3}$$

随后可得

$$\begin{aligned} A^+AX^kBB^+ &= A^+A\left[X^0 + \sum_{j=0}^{k-1} \tfrac{\alpha}{\|A_{i_j,:}\|^2} A_{i_j,:}^T \left(C_{i_j,:} - A_{i_j,:}X^jB\right) B^T\right] BB^+ \\ &= A^+AX^0BB^+ + \sum_{j=0}^{k-1} \tfrac{\alpha}{\|A_{i_j,:}\|^2} A_{i_j,:}^T \left(C_{i_j,:} - A_{i_j,:}X^jB\right) B^T. \end{aligned} \tag{3.4}$$

令 e_i 为单位矩阵的第 i列。最后一步成立，因为$A^+A\,A_{i_s,:}^T = (A^+A)^T A^T e_{i_s} = A^T e_{i_s} = A_{i_s,:}^T$；且 $B^TBB^+ = B^T(BB^+)^T = B^T(B^T)^+B^T = B^T$。由（3.3）和（3.4），得证。 □

定理 3.2。（[17, 定理 A.3]的推广）设矩阵方程（1.1）中 $A \in \mathbb{R}^{m\times p}$，$B \in \mathbb{R}^{q\times n}$ 和 $C \in \mathbb{R}^{m\times n}$ 相容。若 $0 < \alpha < \frac{2}{\|B\|^2}$，则序列$\{X^k\}_{k=0}^{\infty}$由算法2从任意初始矩阵 $X^0 \in \mathbb{R}^{p\times q}$出发得到，并收敛到解 X_*^0。

证明。由算法 2，可得

$$\begin{aligned}\|X^{k+1}-X_*\|_F^2 &= \|X^{k+1}-X^k+X^k-X_*\|_F^2\\ &= \|X^{k+1}-X^k\|_F^2+2\left\langle X^{k+1}-X^k,\, X^k-X_*\right\rangle+\|X^k-X_*\|_F^2\\ &= \tfrac{\alpha^2}{\|A_{i_k,:}\|^4}\,\|A_{i_k,:}^T\left(C_{i_k,:}-A_{i_k,:}X^kB\right)B^T\|_F^2\\ &\quad+2\,\tfrac{\alpha}{\|A_{i_k,:}\|^2}\left\langle A_{i_k,:}^T\left(C_{i_k,:}-A_{i_k,:}X^kB\right)B^T,\, X^k-X_*\right\rangle\\ &\quad+\|X^k-X_*\|_F^2. \end{aligned}\tag{3.5}$$

注意到（利用秩一矩阵的 Frobenius范数）

$$\begin{aligned}\tfrac{\alpha^2}{\|A_{i_k,:}\|^4}\,\|A_{i_k,:}^T\left(C_{i_k,:}-A_{i_k,:}X^kB\right)B^T\|_F^2 &= \tfrac{\alpha^2}{\|A_{i_k,:}\|^2}\,\|\left(C_{i_k,:}-A_{i_k,:}X^kB\right)B^T\|^2\\ &\le \tfrac{\alpha^2\|B\|^2}{\|A_{i_k,:}\|^2}\,\|C_{i_k,:}-A_{i_k,:}X^kB\|^2.\end{aligned}$$

利用 $AX_*B=C$，可得

$$\begin{aligned}&\left\langle A_{i_k,:}^T\left(C_{i_k,:}-A_{i_k,:}X^kB\right)B^T,\, X^k-X_*\right\rangle\\ &=\operatorname{tr}\left[(C_{i_k,:}-A_{i_k,:}X^kB)^TA_{i_k,:}(X^k-X_*)B\right]\\ &=\operatorname{tr}\left[(C_{i_k,:}-A_{i_k,:}X^kB)^T(A_{i_k,:}X^kB-C_{i_k,:})\right]=-\|C_{i_k,:}-A_{i_k,:}X^kB\|^2.\end{aligned}$$

将前两个数学公式中的结果代入（3.5）可得

$$\begin{aligned}&\|X^{k+1}-X_*\|_F^2-\|X^k-X_*\|_F^2\\ &\le \tfrac{\alpha^2\|B\|^2}{\|A_{i_k,:}\|^2}\,\|C_{i_k,:}-A_{i_k,:}X^kB\|^2-\tfrac{2\alpha}{\|A_{i_k,:}\|^2}\,\|C_{i_k,:}-A_{i_k,:}X^kB\|^2\\ &=-\tfrac{2\alpha-\alpha^2\|B\|^2}{\|A_{i_k,:}\|^2}\,\|C_{i_k,:}-A_{i_k,:}X^kB\|^2.\end{aligned}\tag{3.6}$$

若 $0<\alpha<\frac{2}{\|B\|^2}$，则 $\left\{\|X^k-X_*\|_F^2\right\}$是一个单调递减序列，并收敛到非负极限 β。经过 $s+1$ 个循环，即 $k=(s+1)m$ 步后，有

$$\|X^{s+1}-X_*\|_F^2\ \le\ \|X^s-X_*\|_F^2-\sum_{i=1}^{m}\tfrac{2\alpha-\alpha^2\|B\|^2}{\|A_{i,:}\|^2}\,\|C_{i,:}-A_{i,:}X^{s,\,i}B\|^2,\tag{3.7}$$

其中 $X^{s,\,i}$ 表示第 s次扫描中的第 i次迭代。由于对 α 的约束，可得

$$0\ \le\ \sum_{i=1}^{m}\tfrac{2\alpha-\alpha^2\|B\|^2}{\|A_{i,:}\|^2}\,\|C_{i,:}-A_{i,:}X^{s,i}B\|^2\ \le\ \|X^s-X_*\|_F^2-\|X^{s+1}-X_*\|_F^2.$$

显然，$\lim\limits_{s\to\infty}\left(\|X^s-X_*\|_F^2-\|X^{s+1}-X_*\|_F^2\right)=\beta-\beta=0$。因此，我们得出结论：

$$\lim_{s\to\infty}\sum_{i=1}^{m}\tfrac{2\alpha-\alpha^2\|B\|^2}{\|A_{i,:}\|^2}\,\|C_{i,:}-A_{i,:}X^{s,\,i}B\|^2=0.$$

这意味着

$$\lim_{s\to\infty}\left(C_{i,:}-A_{i,:}X^{s,\,i}B\right)=\mathbf{0},\quad i=1,\ldots,m.\tag{3.8}$$

误差序列 $\{\|X^s - X_*\|_F\}$ 有界，因此 $\{X^s\}$ 也有界。假设$\widehat{}$子序列 $\{X^{s_p},\ p = 1, 2, \dots\}$ 收敛到极限 X。由算法 2和（3.8），可得

$$\lim_{p\to\infty} \left(X^{s_p,\,i} - X^{s_p,\,i-1}\right) = \mathbf{0}, \quad i = 2, \dots, m.$$

这意味着 $\widehat{X}$ 是每个子序列的极限 $\{X^{s_p,\,i}\}$ 对于 $i = 1,$ ....., m。由（3.8），可得

$$C_{i,:} = \lim_{p\to\infty} A_{i,:}\, X^{s_p m,\,i} B = A_{i,:}\, \widehat{X} B, \quad i = 1, \dots, m,$$

$\widehat{}$这意味着 X 是（1.1）的一个解。

根据（3.6），显然 $\|X^{k+1} - \widehat{X}\|_F \le \|X^k - \widehat{X}\|_F$。因此，$\{\|X^k - \widehat{X}\|_F\}$ 是一个单调递减的有界序列。因此，我们得出 $\lim_{k\to\infty} \|X^k - \widehat{X}\|_F$存在。由于 $\lim_{p\to\infty} \|X^{k_p m} - \widehat{X}\|_F = 0$，可得 $\lim_{k\to\infty} \|X^k - \widehat{X}\|_F = 0$，即 $\lim_{k\to\infty} X^k = \widehat{X}$。

定义 $\mathcal{N} := \{X \in \mathbb{R}^{p\times q} \mid AXB = \mathbf{0}\}$。令 $Z = \widehat{X} - A^+ A\widehat{X}BB^+$。由于 $Z \in \mathcal{N}$且

$$\begin{aligned}
\langle \widehat{X} - Z\,,\, Z\rangle &= \langle A^+A\widehat{X}BB^+\,,\, \widehat{X} - A^+A\widehat{X}BB^+\rangle \\
&= \langle A^+A\widehat{X}BB^+\,,\, \widehat{X}\rangle - \langle A^+A\widehat{X}BB^+\,,\, A^+A\widehat{X}BB^+\rangle \\
&= \mathrm{tr}(\widehat{X}^T A^+A\widehat{X}BB^+) - \mathrm{tr}((BB^+)^T\widehat{X}^T(A^+A)^T A^+A\widehat{X}BB^+) \\
&= \mathrm{tr}(\widehat{X}^T A^+A\widehat{X}BB^+) - \mathrm{tr}(\widehat{X}^T A^+A\widehat{X}BB^+) = 0,
\end{aligned}$$

我们有 $\|\widehat{X}\|_F^2 = \|\widehat{X} - Z\|_F^2 + \|Z\|_F^2 \ge \|\widehat{X} - Z\|_F^2$。因此，$\widehat{X} - Z$ 是最小$\widehat{}$范数解，即 $X - Z = X_*$。根据引理3.1，可得

$$\begin{aligned}
\widehat{X} - A^+A\widehat{X}BB^+ &= \lim_{k\to\infty} X^k - A^+AX^kBB^+ \\
&= \lim_{k\to\infty} X^0 - A^+AX^0BB^+ = X^0 - A^+AX^0BB^+.
\end{aligned}$$

由此可知 $\widehat{X} = X_* + X^0 - A^+AX^0BB^+ = X^0_*$。 □

定义 $g(\alpha) = 2\alpha - \alpha^2\|B\|^2$。不等式(3.6)表明，$g(\alpha)$在$0 < \alpha < 2\|B\|^{-2}$ 上于 $\alpha = \|B\|^{-2}$时取得最大值，这意味着收敛因子的上界达到最小。在第5 节和第6节的实验中，我们因此采用该参数 $\alpha = \|B\|^{-2}$。注意，最优的 α 可能与使上界最小的取值略有不同（另见图5.4）。

我们已在一般情形下研究了 BK 方法（对A 或 B不加限制），现在转而关注 B 具有满列秩或行秩时的影响。

3.2 B 具有满列秩时的 BK 方法

若 B 具有满列秩（这意味着 $q \ge n$），则可将 B 表示为 $B = QR$，其中 $Q \in \mathbb{R}^{q\times n}$ 具有标准正交列，且 $R \in \mathbb{R}^{n\times n}$ 是上三角矩阵。此步骤需要投入 $\mathcal{O}$（qn^2）的计算量；注意 B 在我们的应用中大小可能较小。于是（1.1）变为

$$AXQ = \widehat{C}, \tag{3.9}$$

其中 $\widehat{C} = CR^{-1}$。关于这种初始变换，有几点需要说明。只有当 B 规模不大且不接近秩亏时，这种做法通常才有意义；尤其是在 C 中存在噪声的情况下（本文不考虑这种情况）。在同一

行中，我们也可以考虑 $AX = CB^{+}\,(= C\,(B^TB)^{-1}B^T)$，此时 $Q = I$ 在（3.9）中。不过，对于本节的理论结果，正交 Q 已足够；不需要采用具有 B^{+} 的变换。

对应问题（3.9）的 BK 方法更新公式为

$$X^{k+1} = X^k + \tfrac{\alpha}{\|A_{i_k,:}\|^2}\, A_{i_k,:}^T\,(\widehat{C}_{i_k,:} - A_{i_k,:}\,X^k\,Q)\,Q^T. \tag{3.10}$$

迭代方案（3.10）一个完整循环的伪代码见算法 3。如前所述，我们将算法稍微改写为以扫描表示（使用 s 作为记号），从而推导出命题 3.3中的公式。与算法 2一样，第 5 行的秩一更新每次迭代需要 $\mathcal{O}$（q（$p+n$）次运算。

算法 **第 3 节** 满列 B 的 BK 方法 rank

1: **Input**: initial guess X^0
2: **Output:** approximate solution to matrix equation (3.9)
3: **for** $s = 0, 1, 2, \ldots$ until convergence **do**
4: Set $X^{s,0} = X^s$
5: Compute $X^{s,i} = X^{s,i-1} + \tfrac{\alpha}{\|A_{i,:}\|^2}\, A_{i,:}^T\,((\widehat{C}_{i,:} - A_{i,:}\,X^{s,i-1}\,Q)\,Q^T), \quad i = 1, \ldots, m$
6: Update $X^{s+1} = X^{s,m}$
7: **end for**

矩阵 Q 的正交性使得我们能够推导出算法 3一次扫描的矩阵表达式。将 [14, 命题 4] 推广到问题（3.9），可得到完整扫描的如下公式。

命题 3.3。 算法 3 的一个循环可写为

$$X^{s+1} = X^s + A^T L_\alpha^{-1}\,(\widehat{C} - AX^s\,Q)\,Q^T, \quad L_\alpha = \mathrm{slt}(AA^T) + \alpha^{-1}\,\mathrm{diag}(AA^T), \tag{3.11}$$

其中 $\mathrm{slt}(AA^T)$ 和 $\mathrm{diag}(AA^T)$ 分别表示方阵 AA^T 的严格下三角部分和对角部分。

证明。 设 X_i 表示 $i \times i$ 主子矩阵（取自矩阵 X）。设 $X_{1:i,\,:}$表示前 i行（取自矩阵 X）。假设算法3可以表示为

$$X^{s,i-1} = X^{s,0} + (A_{1:(i-1),:})^T\,((L_\alpha)_{i-1})^{-1}\,\left[(\widehat{C} - AX^{s,0}\,Q)\,Q^T\right]_{1:(i-1),:}.$$

于是我们有

$$\begin{aligned}
X^{s,i} &= X^{s,i-1} + \tfrac{\alpha}{\|A_{i,:}\|^2}\, A_{i,:}^T\,(\widehat{C}_{i,:} - A_{i,:}\,X^{s,i-1}\,Q)\,Q^T \\
&= X^{s,0} + (A_{1:(i-1),:})^T\,((L_\alpha)_{i-1})^{-1}\,\left[(\widehat{C} - AX^{s,0}\,Q)\,Q^T\right]_{1:(i-1),:} \\
&\quad + \tfrac{\alpha}{\|A_{i,:}\|^2}\, A_{i,:}^T\big(\widehat{C}_{i,:} - A_{i,:}\,\big(X^{s,0} + (A_{1:(i-1),:})^T\,((L_\alpha)_{i-1})^{-1} \\
&\qquad \left[(\widehat{C} - AX^{s,0}\,Q)\,Q^T\right]_{1:(i-1),:}\big)\,Q\big)\,Q^T \\
&= X^{s,0} + (A_{1:(i-1),:})^T\,((L_\alpha)_{i-1})^{-1}\,\left[(\widehat{C} - AX^{s,0}\,Q)\,Q^T\right]_{1:(i-1),:} \\
&\quad + \tfrac{\alpha}{\|A_{i,:}\|^2}\, A_{i,:}^T\,(\widehat{C}_{i,:} - A_{i,:}\,X^{s,0}\,Q)\,Q^T \\
&\quad - \tfrac{\alpha}{\|A_{i,:}\|^2}\, A_{i,:}^T\,A_{i,:}\,(A_{1:(i-1),:})^T\,((L_\alpha)_{i-1})^{-1}\,\left[(\widehat{C} - AX^{s,0}\,Q)\right]_{1:(i-1),:}\,Q^T\,Q\,Q^T \\
&= X^{s,0} + \begin{bmatrix}(A_{1:(i-1),:})^T & A_{i,:}^T\end{bmatrix}
\begin{bmatrix}((L_\alpha)_{i-1})^{-1} & \mathbf{0} \\ -\tfrac{\alpha}{\|A_{i,:}\|^2}\,A_{i,:}\,(A_{1:(i-1),:})^T\,((L_\alpha)_{i-1})^{-1} & \tfrac{\alpha}{\|A_{i,:}\|^2}\end{bmatrix} \\
&\qquad \cdot \begin{bmatrix}\left[(\widehat{C} - AX^{s,0}\,Q)\,Q^T\right]_{1:(i-1),:} \\ (\widehat{C}_{i,:} - A_{i,:}\,X^{s,0}\,Q)\,Q^T\end{bmatrix} \\
&= X^{s,0} + \begin{bmatrix}(A_{1:(i-1),:})^T & A_{i,:}^T\end{bmatrix}
\begin{bmatrix}(L_\alpha)_{i-1} & \mathbf{0} \\ A_{i,:}\,(A_{1:(i-1),:})^T & \alpha^{-1}\|A_{i,:}\|^2\end{bmatrix}
\end{aligned}$$

$$\cdot \begin{bmatrix} \left[(\widehat{C} - AX^{s,0}\,Q)\,Q^T\right]_{1:(i-1),:} \\ (\widehat{C}_{i,:} - A_{i,:}\,X^{s,0}\,Q)\,Q^T \end{bmatrix}$$
$$= X^{s,0} + (A_{1:i,:})^T\,((L_\alpha)_i)^{-1}\left[(\widehat{C} - AX^{s,0}\,Q)\,Q^T\right]_{1:i,:}.$$

因此，得到3.11），这是通过数学归纳法证明的。 □

我们回顾一个关于线性迭代过程的一般结果。

引理 3.4. （例如，参见 [33, 定理 4.2.1]）对于迭代方案 $x^{s+1} = M\,x^s + g$，其中 $M \in \mathbb{R}^{n\times n}$ 和 $x^s \in \mathbb{R}^n$，对任意初始猜测 x^0 均收敛的充要条件是 M 的谱半径满足 $\rho(M) < 1$。

应用引理 3.4，可得以下结果。

定理 3.5. 设矩阵方程 (1.1) 中的 $B \in \mathbb{R}^{q\times n}$ 满列秩。若$\alpha \in (0, 2)$，则算法 3 收敛，且 $\rho(I_{pq} - (QQ^T \otimes A^T\,L_\alpha^{-1}\,A)) < 1$，其中该算子是一个从范围$(B)$ $\otimes$ 到其自身的范围(A^T)映射。此外:

(a) 若 (1.1) 相容，则该过程收敛到此矩阵方程的一个解。(b) 若范围(X^0) $\subseteq$范围(A^T) 且范围$((X^0)^T)$ $\subseteq$范围(B)，则该方法收敛到最小范数解 $A^+\widehat{C}\,Q^T\,(= A^+CB^+)$。

证明。由命题 3.3，我们得到

$$\begin{aligned}
\mathrm{vec}(X^{s+1}) &= \mathrm{vec}(X^s) + \mathrm{vec}(A^T\,L_\alpha^{-1}\,(\widehat{C} - A\,X^s\,Q)\,Q^T) \\
&= \mathrm{vec}(X^s) - \mathrm{vec}(A^T\,L_\alpha^{-1}\,A\,X^s\,Q\,Q^T) + \mathrm{vec}(A^T\,L_\alpha^{-1}\,\widehat{C}\,Q^T) \\
&= \mathrm{vec}(X^s) - (Q\,Q^T \otimes A^T\,L_\alpha^{-1}\,A)\,\mathrm{vec}(X^s) + \mathrm{vec}(A^T\,L_\alpha^{-1}\,\widehat{C}\,Q^T) \\
&= [I_{pq} - (Q\,Q^T \otimes A^T\,L_\alpha^{-1}\,A)]\,\mathrm{vec}(X^s) + \mathrm{vec}(A^T\,L_\alpha^{-1}\,\widehat{C}\,Q^T),
\end{aligned}$$

根据定理 3.2，算法 3 在 $0 < \alpha < 2$ 时收敛。由于该方法收敛（引理 3.4），我们得到关于谱半径的断言。注意$I_{pq} - (QQ^T \otimes A^T\,L_\alpha^{-1}\,A)$ 在 $q > n$ 或 A 不满列秩时，其特征值会等于 1；这会导致解与 2.1 中的解相差一个分量 A^+CB^+。然而，该算子在空间范围(B) $\otimes$范围(A^T) 上的限制的谱半径小于 1。考虑到迭代更新位于范围(A^T)，其转置位于范围(B)，这才是 Kaczmarz 方法收敛所相关的内容。

□

3.3 B 满行秩时的 BK 方法

若 B 满行秩，则矩阵方程 (1.1) 可改写为

$$AX = \widetilde{C}, \tag{3.12}$$

其中 $\widetilde{C} = CB^T\,(BB^T)^{-1}$。这一变换也可以在与第 3.2 节相同的条件下考虑。注意，这是一个多右端项线性系统，我们对该问题采用涉及秩一更新的 Kaczmarz 型方法。于是，BK 方法对应的更新公式（参见算法 2 第 5 行）为

$$X^{k+1} = X^k + \tfrac{\alpha}{\|A_{i_k,:}\|^2}\,A_{i_k,:}^T\,(\widetilde{C}_{i_k,:} - A_{i_k,:}X^k). \tag{3.13}$$

当矩阵 B 的维度相对较小时，也就是说，q 和 n 都相对较小时，每次迭代的计算成本略低于算法 2，~其每次迭代复杂度为 $\mathcal{O}(q(p+n))$。此外，必须预先计算 C，这会产生初始计算成本。在实验 6.2，

我们讨论在模型（1.2）的背景下使用更新公式（3.13）得到的一些数值结果。与算法 3类似，算法 3.13）的迭代公式的一个循环展示于算法 4中。该方法还包括第 5 行内联的秩一更新，每次迭代的计算成本为 $\mathcal{O}(pq)$。

算法 4 当 B 满行秩时的 BK 方法

1: **Input**: initial guess X^0
2: **Output:** approximate solution to matrix equation (3.12)
3: **for** $s = 0, 1, 2, \ldots$ until convergence **do**
4: Set $X^{s,0} = X^s$
5: Compute $X^{s,i} = X^{s,i-1} + \frac{\alpha}{\|A_{i,:}\|^2} A_{i,:}^T (\widetilde{C}_{i,:} - A_{i,:} X^{s,i-1}), \quad i = 1, \ldots, m$
6: Update $X^{s+1} = X^{s,m}$
7: **end for**

算法 4 可视为标准 Kaczmarz 方法（1.4）针对多右端项的一种自然扩展。据我们所知，文献中此前尚未提出或研究过这种方法。与将 $AX = Y$ 视为一组 q 相互独立的线性系统$AX_{:,j} = Y_{:,j}$ 并分别求解每个系统不同，算法 4 在每次迭代中同时更新 X 的所有列。

根据第 3.2 节给出的收敛结果，我们推导出以下推论。方程（3.14）是经典 Kaczmarz 算法针对多右端项情形的直接推广 [14, 命题 4]。条目（a）是对 [14,定理 5]，的扩展，而（b）部分则是多右端项情形所特有的。

定理 3.6. 算法 4 的一个循环可写为

$$X^{s+1} = X^s + A^T L_\alpha^{-1} (\widetilde{C} - AX^s). \tag{3.14}$$

若 $\alpha \in (0, 2)$，则算法4 收敛，且 $\rho(I_p - A^T L_\alpha^{-1} A) < 1$。此外：

(a) 若（1.1）相容，则该过程收敛到此矩阵方程的一个解。(b) 若 $\text{range}(X^0) \subseteq \text{range}(A^T)$ 且 $\text{range}((X^0)^T) \subseteq \text{range}(B)$，则该方法~收敛到最小范数解 A^+C $(= A^+CB^+)$。

证明。将 Q 替换为单位矩阵 I，命题3.3中的（3.14）可直接得到，它对应算法4 的一个循环。

基于（3.14），将 Q 替换为 I，应用定理3.5，可得

$$\text{vec}(X^{s+1}) = [I_{pq} - (I_q \otimes A^T L_\alpha^{-1} A)] \,\text{vec}(X^s) + \text{vec}(A^T L_\alpha^{-1} \widehat{C} Q^T),$$

其中，$I_{pq} - (I_q \otimes A^T L_\alpha^{-1} A)$ 是维数为 $pq \times pq$ 的分块对角矩阵，每个对角块均等于 $I_q - A^T L_\alpha^{-1} A$。因此，我们有

$$\rho\,[I_{pq} - (I_q \otimes A^T L_\alpha^{-1} A)] = \max_{i=1,\ldots,p} |1 - \lambda_i(A^T L_\alpha^{-1} A)| = \rho\,(I_p - A^T L_\alpha^{-1} A).$$

根据定理 3.2，算法 4 在 $0 < \alpha < 2$ 时收敛。因此，由引理 3.4可知 $\rho(I_p - A^T L_\alpha^{-1} A) < 1$。

□

算法 3 的提出基于以下假设：B 具有满列秩，而算法 4假设 B 具有满行秩。当 B 为方阵且非奇异时，这两个条件均成立。我们探讨这一特殊情形下两者之间的联系，并由此得到以下命题。

命题 3.7。算法 3 和 4 在系数矩阵 B 非奇异时等价。

证明。这一等价性源于如下事实：若 B 为方阵，则由 B 的 QR 分解得到的矩阵 Q为方阵，因而正交。此时我们有

$\widehat{C} = CR^{-1}$ 与 $\widetilde{C} = CR^{-1}Q^T$。这意味着

$$\begin{aligned} X^{s,i} &= X^{s,i-1} + \tfrac{\alpha}{\|A_{i,:}\|^2} A_{i,:}^T \left(\widehat{C}_{i,:} - A_{i,:} X^{s,i-1} Q\right) Q^T \\ &= X^{s,i-1} + \tfrac{\alpha}{\|A_{i,:}\|^2} A_{i,:}^T \left(C_{i,:} R^{-1} Q^T - A_{i,:} X^{s,i-1}\right) \\ &= X^{s,i-1} + \tfrac{\alpha}{\|A_{i,:}\|^2} A_{i,:}^T \left(\widetilde{C}_{i,:} - A_{i,:} X^{s,i-1}\right). \qquad \square \end{aligned}$$

在第6节的实验中，我们发现，迭代次数确实完全相同，即使在有限精度算术下也是如此。我们注意到，矩阵方程（3.9）和（3.12）是更一般形式（1.1）的特例。因此，针对矩阵方程（1.1）推导出的理论结果仍适用于方程（3.9）和（3.12）；然而，反过来则未必成立。

现在，我们提供一个示意表，总结循环 Kaczmarz 方法和随机 Kaczmarz 方法的主要性质。与线性系统 $Ax = b$一样，对于（1.1），我们可以得到循环 Kaczmarz 方法和随机 Kaczmarz 方法按步和按扫描计算的上界，以及它们在缩放和行置换下的不变性。

方法	步上界	扫描上界	缩放	排列
循环	—	$\rho(I - A^T L^{-1} A)$	✓	—
随机化	$\delta^{1/2}$	$\delta^{m/2}$	—	✓

循环Kaczmarz每一步都没有非平凡的上界（即小于1的上界），因为在某些步骤中可能不会取得任何进展。需要特别注意的是，谱半径给出的是渐近的因子（因此初始阶段可能不成立），而随机上界是按期望成立的（因此也可能不成立，尤其是在开始阶段；请注意，$\sqrt{\mathbb{E}_k[\|X^k - X_*\|_F^2]} \geq \mathbb{E}_k[\|X^k - X_*\|_F]$ 因此 $\delta^{1/2}$ 是在随机方法中一次行作用步中 $\|X_k - X_*\|_F$ 进展的期望上界）。循环方法和随机方法的扫描界哪个更小取决于具体问题。循环Kaczmarz的收敛速度可能比随机方法（快得多），但也可能反过来，这主要取决于行排序。如果我们首先对 A的行进行随机排列（从而 C），通常会得到既不特别有利也不特别不利的行排序，那么循环Kaczmarz的表现通常会与其随机对应方法非常相似。

4 GRBK方法及其松弛与确定性变体

在本节中，我们将 [2] 中提出的贪心策略扩展至求解矩阵方程（1.1），由此得到 GRBK 方法，该方法将在第4.1节介绍。此外，受 [3] 中的松弛策略以及 [11, 20], 中的最大加权残差策略启发，我们进一步改进 GRBK 方法，得到其松弛与确定性变体，即 RGRBK 和 MWRBK 方法，分别见第4.2节和第4.3节。

4.1 GRBK 方法

暂且将注意力转向具有相关残差r^k 的线性系统（1.3）：$= b - Ax^k$。Bai and Wu [2] 指出，残差范数较大的行对整体误差的影响更大，因此他们提出了 GRK 方法，在求解线性系统（1.3）时优先选择这些行。GRK 方法中的贪心选择策略涉及 r^k 相对于 A 的最大行范数和平均行范数（即 $\max_i \left(\frac{r_i^k}{\|A_{i,:}\|^2} \right)^2$，以及 $\frac{\|r^k\|^2}{\|A\|_F^2}$），其中二者的占比之和为 1，并通过固定因子1 2进行平衡。我们将 [2], 中的贪心策略加以扩展，提出算法5 来求解矩阵方程（1.1）。

GRBK方法的行选择策略基于残差。第 5 行中的两项分别表示 R^k 相对于 A的最大行范数和平均行范数。这些项在算法6 和7以及分析中也发挥着重要作用。

算法 5 GRBK方法

1 X^0：输入：初始猜测 输出：矩阵方程 (1.1) 的近似解 3：

2

对于 $k = 0, 1, 2, \ldots\ldots$ 直到收敛为止，执行 4：计算 $R^0 = C - AX^0B$5：计算

$\gamma_k = \frac{1}{2}\left[\max_{1\le i\le m} \frac{\|R_{i,:}^k\|^2}{\|A_{i,:}\|^2} + \frac{\|R^k\|_F^2}{\|A\|_F^2}\right]$6：确定指标集

$\mathcal{J}_k = \left\{i \mid \|R_{i,:}^k\|^2 \ge \gamma_k \|A_{i,:}\|^2\right\}$7：令 $\widetilde{R}^k$ 包含相应的行：$\widetilde{R}_{i,:}^k = \begin{cases} R_i^k, \\ \end{cases}$

：$i \in \mathcal{J}_k$，如果，否则 8：以概率 $\mathbb{P}$选择 $i_k \in \mathcal{J}_k$（行 $= i_k$）$= \|\widetilde{R}_{i_k,:}^k$

$\mathbf{0}$ $\|^2 / \|\widetilde{R}^k\|_F^2$9：更新 $X^{k+1} = X^k + \frac{\alpha}{\|A_{i_k,:}\|^2} A_{i_k,:}^T (R_{i_k,:}^k B^T)$ 10：计算

$R^{k+1} = R^k - \frac{\alpha}{\|A_{i_k,:}\|^2} (AA_{i_k,:}^T) ((R_{i_k,:}^k B^T) B)$ 11

：**结束循环**

通过第二项，我们可以选择残差范数高于平均值（相对于 A 的行范数）的行。仅使用第一项（如算法7中所做的那样），我们可以选取残差范数最大的行。在算法 5 中，我们取两项的平均值。算法 6 中还考虑了其他权重。算法 5 第 9 行的更新操作涉及秩一更新，其计算复杂度为$\mathcal{O}(q(p+n))$。第 10 行的计算包含类似的更新，计算量为 $\mathcal{O}(qn+mp+mn)$。因此，算法 5 每次迭代的总体计算成本为 $\mathcal{O}(pq+qn+mp+mn)$：关于矩阵规模呈二次关系。

注4.1。 由算法 5第 6 行可知，指标集 $\mathcal{J}_k$ 可以表示为

$$\mathcal{J}_k = \left\{i \;\middle|\; \frac{\|R_{i,:}^k\|^2}{\|A_{i,:}\|^2} \ge \frac{1}{2} \max_{1\le i\le m} \frac{\|R_{i,:}^k\|^2}{\|A_{i,:}\|^2} + \frac{1}{2}\frac{\|R^k\|_F^2}{\|A\|_F^2}\right\}, \tag{4.1}$$

这会筛除残差范数较小的行。指标集 $\mathcal{J}_k$ 在每次迭代中都会更新，并且非空，原因如下。首先，我们观察到

$$\max_{1\le i\le m} \frac{\|R_{i,:}^k\|^2}{\|A_{i,:}\|^2} \ge \sum_{i=1}^m \frac{\|A_{i,:}\|^2}{\|A\|_F^2} \frac{\|R_{i,:}^k\|^2}{\|A_{i,:}\|^2} = \frac{\|R^k\|_F^2}{\|A\|_F^2}.$$

然后，令 $\frac{\|R_j^k\|^2}{\|A_j\|^2} = \frac{\|R_{i,:}^k\|^2}{\|A_{i,:}\|^2}$ 最大值$_{1\le i\le m}$，我们得到

$$\frac{\|R_{j,:}^k\|^2}{\|A_{j,:}\|^2} \ge \frac{1}{2} \max_{1\le i\le m} \frac{\|R_{i,:}^k\|^2}{\|A_{i,:}\|^2} + \frac{1}{2}\frac{\|R^k\|_F^2}{\|A\|_F^2} = \gamma_k.$$

该结果表明 $j \in \mathcal{J}_k$，因此 $\mathcal{J}_k \ne \emptyset$。

现在我们分析算法 5 的收敛性。为此，我们将 [2, 定理 3。1]（该定理针对线性系统（1.3））推广至问题（1.1）。我们首先介绍一些相关量，它们将在接下来的若干结果中发挥重要作用。定义

$$\Omega_k := \left\{i \;\middle|\; \frac{\|R_{i,:}^k\|^2}{\|A_{i,:}\|^2} < \frac{1}{m}\sum_{i\in[m]} \frac{\|R_{i,:}^k\|^2}{\|A_{i,:}\|^2}\right\}. \tag{4.2}$$

这意味着，Ω_k 恰好是残差范数（相对于 A）低于平均值的行指标集合。显然，$\frac{1}{m}\sum_{i\in[m]} \frac{\|R_{i,:}^k\|^2}{\|A_{i,:}\|^2} \le$ 最大值$_{1\le i\le m} \frac{\|R_{i,:}^k\|^2}{\|A_{i,:}\|^2}$。在下文中，我们作一个非常温和的假设：比值$\|R_{i,:}^k\|^2 / \|A_{i,:}\|^2$并非完全相同（这以概率为1成立）。因此，我们可以写作$\frac{1}{m}\sum_{i\in[m]} \frac{\|R_{i,:}^k\|^2}{\|A_{i,:}\|^2} = \varepsilon \cdot$ 最大值$_{1\le i\le m} \frac{\|R_{i,:}^k\|^2}{\|A_{i,:}\|^2}$，其中 $0 < \varepsilon < 1$。接着，令 $\theta \in [0,1]$，并定义

$$\varphi_{k,\theta} := \theta \cdot \frac{1}{\sum_{i\in[m]\setminus\Omega_k} \|A_{i,:}\|^2 + \varepsilon \sum_{i\in\Omega_k} \|A_{i,:}\|^2} + (1-\theta)\cdot\frac{1}{\|A\|_F^2}. \tag{4.3}$$

容易看出，对所有 $\theta \in [0,1]$，都有$\varphi_{k,\theta} \geq \frac{1}{\|A\|_F^2}$。此外，根据我们的温和假设可知，对所有 $\theta \in (0,1]$，都有$\varphi_{k,\theta} > \frac{1}{\|A\|_F^2}$。利用这一点$\varphi_{k,\theta}$，接下来我们定义一个量，它作为收敛因子的上界发挥关键作用：

$$\delta_{k,\theta} := 1 - (2\alpha - \alpha^2 \|B\|^2) \cdot \varphi_{k,\theta} \cdot \sigma_{\min}^2(A)\, \sigma_{\min}^2(B). \tag{4.4}$$

由$\varphi_{k,\theta}$的下界可得

$$\delta_{k,\theta} < 1 - (2\alpha - \alpha^2 \|B\|^2) \cdot \|A\|_F^{-2} \cdot \sigma_{\min}^2(A)\, \sigma_{\min}^2(B) = \delta, \tag{4.5}$$

其中 δ 是算法1 的收敛因子的上界，如（2.3）所示。我们注意到，δ 与$\delta_{k,0}$重合。在下一个定理中，$\varphi_{k,1/2}$ 和$\delta_{k,1/2}$ 是相关的；对于 θ 的其他取值，我们稍后将需要$\varphi_{k,\theta}$ 和$\delta_{k,\theta}$。

定理4.2.（[2,定理3.1]的推广）设带有 $A \in \mathbb{R}^{m\times p}$、$B \in \mathbb{R}^{q\times n}$ 和 $C \in \mathbb{R}^{m\times n}$ 的矩阵方程(1.1)相容。若 $0 < \alpha < \frac{2}{\|B\|^2}$，则由算法 5得到的序列$\{X^k\}_{k=0}^{\infty}$依期望收敛到(1.1)的一个解。若范围$(X^0) \subseteq$范围$(A^T)$且范围$((X^0)^T) \subseteq$范围$(B)$，则算法 5依期望收敛到唯一最小范数解 X_*。此外，解误差满足

$$\mathbb{E}_k\left[\|X^{k+1} - X_*\|_F^2\right] \leq \delta_{k,1/2} \cdot \|X^k - X_*\|_F^2 \leq \delta \cdot \|X^k - X_*\|_F^2. \tag{4.6}$$

因此，我们有

$$\mathbb{E}\left[\|X^{k+1} - X_*\|_F^2\right] \leq \prod_{\ell=0}^{k} \delta_{\ell,1/2} \cdot \|X^0 - X_*\|_F^2 \leq \delta^{k+1} \cdot \|X^0 - X_*\|_F^2. \tag{4.7}$$

证明。令 $\zeta_k = \frac{1}{2}\left[\frac{1}{\|R^k\|_F^2}\right.$最大值$_{1\leq i\leq m} \frac{\|R_i^k\|^2}{\|A_i\|^2} + \left.\frac{1}{\|A\|_F^2}\right]$。然后，根据算法5第 5 行，对于所有 $i \in \mathcal{J}_k$，我们有 $\frac{\|R_i^k\|^2}{\|A_i\|^2} \geq \frac{\|A_i\|^2}{\|A\|^2} \geq \gamma_k = \|R^k\|_F^2\, \zeta_k$。对于 $0 < \alpha < \frac{2}{\|B\|^2}$，我们对（3.6）在给定前 k 次迭代的条件下取期望，得到：

$$\begin{aligned}
&\mathbb{E}_k\left[\|X^{k+1} - X_*\|_F^2\right] \\
&\quad \leq \|X^k - X_*\|_F^2 - (2\alpha - \alpha^2 \|B\|^2) \sum_{i=1}^{m} \frac{\|\widetilde{R}_{i,:}^k\|^2}{\|\widetilde{R}^k\|_F^2} \frac{\|C_{i,:} - A_{i,:} X^k B\|^2}{\|A_{i,:}\|^2} \\
&\quad \leq \|X^k - X_*\|_F^2 - (2\alpha - \alpha^2 \|B\|^2) \sum_{i\in\mathcal{J}_k} \frac{\|\widetilde{R}_{i,:}^k\|^2}{\|\widetilde{R}^k\|_F^2} \zeta_k \|R^k\|_F^2 \\
&\quad = \|X^k - X_*\|_F^2 - (2\alpha - \alpha^2 \|B\|^2)\, \zeta_k\, \|A\,(X^k - X_*)\,B\|_F^2 \\
&\quad = \|X^k - X_*\|_F^2 - (2\alpha - \alpha^2 \|B\|^2)\, \zeta_k\, \|(B^T \otimes A)\,\mathrm{vec}(X^k - X_*)\|^2,
\end{aligned} \tag{4.8}$$

其中 $\mathcal{J}_k$与（4.1）中的相同。此外，我们注意到 vec$(X^0 - X_*) \in$属于 range$(B \otimes A^T)$，并且

$$\mathrm{vec}\left(\tfrac{\alpha}{\|A_{i_k,:}\|^2} A_{i_k,:}^T \left((C_{i_k,:} - A_{i_k,:} X^k B)\, B^T\right)\right) \in \mathrm{range}(B \otimes A^T).$$

因此，由归纳法可得 vec$(X^k - X_*) \in$range$(B \otimes A^T)$。由于对于任意 $x \in$ range(A)，$\|A^T x\|^2 \geq \sigma_{\min}^2(A)\, \|x\|^2$ 均成立（例如参见 [31, 引理 2.1]），故可得

$$\begin{aligned}
\|(B^T \otimes A)\,\mathrm{vec}(X_k - X_*)\| &\geq \sigma_{\min}(B \otimes A^T)\, \|\mathrm{vec}(X_k - X_*)\| \\
&= \sigma_{\min}(A)\, \sigma_{\min}(B)\, \|X_k - X_*\|_F.
\end{aligned} \tag{4.9}$$

因此，不等式（4.8）变为

$$\mathbb{E}_k\left[\|X^{k+1} - X_*\|_F^2\right] \leq \left(1 - (2\alpha - \alpha^2 \|B\|^2)\, \zeta_k\, \sigma_{\min}^2(A)\, \sigma_{\min}^2(B)\right) \cdot \|X^k - X_*\|_F^2. \tag{4.10}$$

此外，我们有

$$\begin{aligned}\zeta_k \|A\|_F^2 &= \frac{1}{2}\frac{\max_i \frac{\|R_{i,:}^k\|^2}{\|A_{i,:}\|^2}}{\frac{\|R^k\|_F^2}{\|A\|_F^2}} + \frac{1}{2} \\ &= \frac{1}{2}\frac{\max_i \frac{\|R_{i,:}^k\|^2}{\|A_{i,:}\|^2}}{\sum_{i\in[m]\setminus\Omega_k}\frac{\|A_{i,:}\|^2}{\|A\|_F^2}\frac{\|R_{i,:}^k\|^2}{\|A_{i,:}\|^2} + \sum_{i\in\Omega_k}\frac{\|A_{i,:}\|^2}{\|A\|_F^2}\frac{\|R_{i,:}^k\|^2}{\|A_{i,:}\|^2}} + \frac{1}{2} \\ &\geq \frac{1}{2}\frac{1}{\sum_{i\in[m]\setminus\Omega_k}\frac{\|A_{i,:}\|^2}{\|A\|_F^2} + \varepsilon\sum_{i\in\Omega_k}\frac{\|A_{i,:}\|^2}{\|A\|_F^2}} + \frac{1}{2}. \end{aligned} \tag{4.11}$$

这意味着 $\zeta_k \geq \varphi_{k,\ 1/2}$，在所有残差行的权重并非都相等这一温和假设下，该值大于 $\|A\|_F^{-2}$。应用全期望定律$\mathbb{E}[\cdot] = \mathbb{E}[\mathbb{E}_k[\cdot]]$（例如参见 [2, 3, 31]），不等式（4.10）变为

$$\mathbb{E}\left[\|X^{k+1} - X_*\|_F^2\right] \leq \delta_{k,\,1/2} \cdot \mathbb{E}\left[\|X^k - X_*\|_F^2\right]. \tag{4.12}$$

对迭代次数进行归纳，并结合 (4.5)，可得估计式 (4.7)。注意，若 range$(X^0) \subseteq$ range(A^T) 且 range$((X^0)^T) \subseteq$ range(B)，则算法 5 中的秩一更新使得对所有 k，我们都保持性质 range$(X^k) \subseteq$ range(A^T) 且 range$((X^k)^T) \subseteq$ range(B)。因此，序列 $\{X^k\}_{k=0}^{\infty}$ 依期望收敛到唯一最小范数解 X_*。□

顺便一提，在 [2]，中，Bai and Wu 发现 $r_{i_{k-1}}^k = 0$，其中 $r_{i_{k-1}}^k$ 是相应残差 r^k 的第 i_{k-1} 个分量。利用这一关系，他们得到

$$\begin{aligned}\frac{\max_i \frac{\|r_i^k\|^2}{\|A_{i,:}\|^2}}{\sum_{i\in[m]}\frac{\|A_{i,:}\|^2}{\|A\|_F^2}\frac{\|r_i^k\|^2}{\|A_{i,:}\|^2}} &= \frac{\max_i \frac{\|r_i^k\|^2}{\|A_{i,:}\|^2}}{\sum_{i\neq i_{k-1}}\frac{\|A_{i,:}\|^2}{\|A\|_F^2}\frac{\|r_i^k\|^2}{\|A_{i,:}\|^2}} \\ &\geq \frac{\|A\|_F^2}{\sum_{i\neq i_{k-1}}\|A_{i,:}\|^2} \geq \frac{\|A\|_F^2}{\max_{1\leq j\leq m}\sum_{i\neq j}\|A_{i,:}\|^2}.\end{aligned}$$

然而，对于我们的矩阵方程 (1.1)，通常不会存在残差为零的行（即不存在索引 i 满足 $R_{i,\,:}^k = \mathbf{0}$）。因此，在推导 (4.11) 时，我们需要采用一种不同的集合 Ω_k（与[2]相比）。对于文献 [2]所讨论的线性系统 (1.3)，也可以利用我们的集合 Ω_k。

由不等式（ 4 12 $\{\mathbb{E}[\|X^k - X_*\|_F^2]\}$ $0 < \alpha <$ 。），$2\|B\|^{-2}$ 3 6 4 12 $g\ \alpha\ =$ 当时，序列递减。此外，与（。）类似，不等式（。）还包含函数（）$2\alpha - \alpha^2\|B\|^2$，该函数在 $\alpha = \|B\|^{-2}$ 处取得最大值。这表明，当参数 α 设置为 $\|B\|^{-2}$时，收敛因子的上界达到最小。

4.2 RGRBK 方法

在上一小节中，GRK 方法对 r^k 相对于 A的最大行范数和平均行范数采用了固定权重 1 2。一个自然的问题是：如果采用不同的比例，会发生什么？为探究这一问题，Bai and Wu [3], 在（1.3）的背景下引入了可变参数 $\theta \in [0,1]$，并提出了 RGRK 方法。

RGRK 方法中的参数 θ 有两种情况特别值得关注：

- $\theta = \frac{1}{2}$：RGRK 方法在 [2] 中与 GRK 方法一致。
- $\theta = 1$: RGRK 方法在 [11, 20],某种确定性方法中等价于 MWRK 方法，该方法选择残差最大的行；另请参阅第 4.3 节。

扩展 RGRK [3]（适用于线性系统（1.3））， 我们现在提出并研究一种 RGRBK 方法，用于求解矩阵方程（1.1），见算法 6（另见表 1.2）。

算法 6 RGRBK 方法

1 $\theta \in [0,1]$：**输入：初始猜测** X^0，**参数**：2 **输出：矩阵方程 (1.1) 的近似解 3：对于** $k=0$，**1，2，...，直到收敛为止 4：计算** $R^0 = C - AX^0B$**5：计算** $\xi_k = \theta \cdot \max_{1\le i\le m} \frac{\|R_i^k\|^2}{\|A_i\|^2} + (1-\theta)\cdot \frac{\|R^k\|_F^2}{\|A\|_F^2}$**6：确定指标集** $\mathcal{H}_k = \{i \mid \|R_{i,:}^k\|^2 \ge \xi_k \|A_{i,:}\|^2\}$**7：令** $\widetilde{R}^k$ **为相应选定的行：** $\widetilde{R}_{i,:}^k = \begin{cases} R_i^k, & i \in \mathcal{H}_k, \\ \mathbf{0}, & \text{否则} \end{cases}$ **8：以概率** $\mathbb{P}$ **选择** $i_k \in \mathcal{H}_k$ （行 $= i_k$） $= \|\widetilde{R}_{i_k,:}^k\|^2 / \|\widetilde{R}^k\|_F^2$**9：更新** $X^{k+1} = X^k + \frac{\alpha}{\|A_{i_k}\|^2} A_{i_k,:}^T (R_{i_k,:}^k B^T)$ **10：计算** $R^{k+1} = R^k - \frac{\alpha}{\|A_{i_k}\|^2} (AA_{i_k,:}^T)((R_{i_k,:}^k B^T)B)$ **11：结束循环**

注意，算法 6 与算法 5具有相同的每次迭代计算复杂度。指标集 $\mathcal{H}_k$ 非空的原因与备注 4.1类似。类似于定理 4.2，我们推导出定理4.3；其证明略去。

定理4.3. 设矩阵方程（1.1）与 $A \in \mathbb{R}^{m\times p}$、 $B \in \mathbb{R}^{q\times n}$ 以及 $C \in \mathbb{R}^{m\times n}$ 相容。若 $0 < \alpha < \frac{2}{\|B\|^2}$，则由算法 6 得到的序列 $\{X^k\}_{k=0}^{\infty}$ 依期望收敛到方程（ 1.1 ）的一个解。若 $\text{range}(X^0) \subseteq \text{range}(A^T)$ 且 $\text{range}((X^0)^T) \subseteq \text{range}(B)$，则算法 6 依期望收敛到唯一最小范数解 X_*。此外，解误差满足

$$\mathbb{E}_k\left[\|X^{k+1} - X_*\|_F^2\right] \le \delta_{k,\theta} \cdot \|X^k - X_*\|_F^2 \le \delta \cdot \|X^k - X_*\|_F^2,$$

因此，我们有

$$\mathbb{E}\left[\|X^{k+1} - X_*\|_F^2\right] \le \prod_{\ell=0}^{k} \delta_{\ell,\theta} \cdot \|X^0 - X_*\|_F^2 \le \delta^{k+1} \cdot \|X^0 - X_*\|_F^2.$$

我们将比较本节所讨论的算法以及 [32] 中提出的RBK算法的收敛因子上界，见第4.3节。

4.3 MWRBK 方法

我们现在将 [11, 20] 中的最大加权残差策略（该策略针对线性系统（1.3））推广到矩阵方程（1.1）（另参见表1.2）。我们在每次迭代中直接选取加权残差最大的行，这也可以通过取 $\theta = 1$从RGRBK 方法得到。由此得到如下MWRBK 方法（算法7），这是一种确定性方法。

算法 7 MWRBK 方法

1 X^0：**输入：初始猜测：输出：矩阵方程（1.1）的近似解 3：对于** 2 $k=0$，**1，2，.....直至收敛，执行 4：计算** $R^0 = C - AX^0B$**5：令** i_k **为使** $\|R_i^k\|^2 / \|A_i\|^2$ **达到最大的下标；若有多个，则选择最小者 6：更新** $X^{k+1} = X^k + \frac{\alpha}{\|A_{i_k}\|^2} A_{i_k,:}^T (R_{i_k,:}^k B^T) B$ **7：计算** $R^{k+1} = R^k - \frac{\alpha}{\|A_{i_k}\|^2} (AA_{i_k,:}^T)((R_{i_k,:}^k B^T)B)$ **8：结束循环**

下面给出算法7 收敛速度的理论估计，其推导思路与 [11, 定理 3.1]相同。

定理 4.4。设矩阵方程 (1.1) 满足 $A \in \mathbb{R}^{m\times p}$，$B \in \mathbb{R}^{q\times n}$ 且 $C \in \mathbb{R}^{m\times n}$ 相容。设 $0 < \alpha < \frac{2}{\|B\|^2}$ 和 X^k 分别表示由算法 7 得到的第 k 次迭代，其中 range$(X^0) \subseteq$ range(A^T) 且 range$((X^0)^T) \subseteq$ range(B)。那么

$$\|X^{k+1} - X_*\|_F^2 \le \prod_{\ell=0}^{k} \delta_{\ell,1} \cdot \|X^0 - X_*\|_F^2 \le \delta^{k+1} \cdot \|X^0 - X_*\|_F^2, \tag{4.13}$$

其中 $\delta_{\ell,1}$ 定义于（4.4）中。

证明。对于 $\theta = 1$，该方法是确定性的，因此证明技巧不同于定理 4.2 和 4.3。对于 $0 < \alpha < \frac{2}{\|B\|^2}$，利用 i_k 使算法 7 第 5 行中的量最大化，可得

$$\begin{aligned}
&\|X^{k+1} - X_*\|_F^2 \\
&\le \|X^k - X_*\|_F^2 - \tfrac{2\alpha-\alpha^2\|B\|^2}{\|A_{i_k,:}\|^2} \|R^k_{i_k,:}\|^2 \\
&= \|X^k - X_*\|_F^2 - (2\alpha - \alpha^2\|B\|^2) \frac{\|R^k_{i_k,:}\|^2}{\|A_{i_k,:}\|^2} \frac{\|R^k\|_F^2}{\sum_{i\in[m]\setminus\Omega_k} \frac{\|R^k_{i,:}\|^2}{\|A_{i,:}\|^2}\|A_{i,:}\|^2 + \sum_{i\in\Omega_k} \frac{\|R^k_{i,:}\|^2}{\|A_{i,:}\|^2}\|A_{i,:}\|^2} \\
&\le \|X^k - X_*\|_F^2 - (2\alpha - \alpha^2\|B\|^2) \frac{\|R^k_{i_k,:}\|^2}{\|A_{i_k,:}\|^2} \frac{\|R^k\|_F^2}{\frac{\|R^k_{i_k,:}\|^2}{\|A_{i_k,:}\|^2}\left(\sum_{i\in[m]\setminus\Omega_k}\|A_{i,:}\|^2 + \sum_{i\in\Omega_k}\varepsilon\|A_{i,:}\|^2\right)} \\
&= \|X^k - X_*\|_F^2 - (2\alpha - \alpha^2\|B\|^2) \frac{\|A(X^k - X_*)B\|_F^2}{\sum_{i\in[m]\setminus\Omega_k}\|A_{i,:}\|^2 + \varepsilon\sum_{i\in\Omega_k}\|A_{i,:}\|^2} \\
&\le \left(1 - (2\alpha - \alpha^2\|B\|^2)\cdot\varphi_{k,1}\cdot\sigma_{\min}^2(A)\,\sigma_{\min}^2(B)\right)\cdot\|X^k - X_*\|_F^2.
\end{aligned}$$

在最后一行中，我们使用了（4.9）。对迭代次数进行归纳，得到不等式（4.13）。

□

与第 3.1 节和4.1节中的讨论类似，当 $\alpha = \|B\|^{-2}$时，RGRBK 和 MWRBK 方法的收敛因子上界达到最小。

表 4.1 总结了本节涉及的方法在求解矩阵方程（1.1）时收敛因子的上界。回顾一下，δ等于$\delta_{k,0}$。

表 4.1 求解 $AXB = C$ 时 RBK [32], GRBK、RGRBK 和 MWRBK 方法的收敛因子上界。

方法	收敛因子上界	结果
RBK [32]	$\delta = \delta_{k,0} = 1 - \frac{2\alpha-\alpha^2\|B\|^2}{\|A\|_F^2}\sigma_{\min}^2(A)\,\sigma_{\min}^2(B)$	[32, 定理 1]
GRBK（算法5）	$\delta_{k,1/2} = 1 - (2\alpha-\alpha^2\|B\|^2)\cdot\varphi_{k,1/2}\cdot\sigma_{\min}^2(A)\,\sigma_{\min}^2(B) < \delta$	定理 4.2
RGRBK（算法6）	$\delta_{k,\theta} = 1 - (2\alpha-\alpha^2\|B\|^2)\cdot\varphi_{k,\theta}\cdot\sigma_{\min}^2(A)\,\sigma_{\min}^2(B) < \delta$	定理 4.3
MWRBK（算法 7）	$\delta_{k,1} = 1 - (2\alpha-\alpha^2\|B\|^2)\cdot\varphi_{k,1}\cdot\sigma_{\min}^2(A)\,\sigma_{\min}^2(B) < \delta$	定理 4.4

关于这些上界，我们已经看到，GRBK、RGRBK 和 MWRBK 的收敛因子上界不大于 RBK 的δ。对于线性系统（1.3），[3, 第 2] 节的作者表明，RGRK 的收敛因子上界是参数 θ的单调递减函数。我们通过证明所提出的 RGRBK 方法的收敛因子上界也随 θ 递减，将这一结果推广到（在前文所述的残差行不相等这一较宽松假设下）：

$$\frac{\partial\delta_{k,\theta}}{\partial\theta} = (2\alpha-\alpha^2\|B\|^2)\,\sigma_{\min}^2(A)\,\sigma_{\min}^2(B)\cdot\left(\frac{1}{\|A\|_F^2} - \frac{1}{\sum_{i\in[m]\setminus\Omega_k}\|A_{i,:}\|^2 + \varepsilon\sum_{i\in\Omega_k}\|A_{i,:}\|^2}\right) < 0.$$

这意味着，收敛因子的上界 $\delta_{k,\ \theta}$ 是关于 θ 的递减函数。因此，我们预计（确定性）MWRBK 方法（对应于 $\theta = 1$）可能收敛得最快；以下各节中的实验确实证实了这一点。

表 4.2概述了本节中用于求解方程（1.1）和（1.3）的方法所采用的指标选择准则。RGRBK 方法有一个参数 $\theta \in [0,1]$；当 $\theta = \frac{1}{2}$ 时，它变为 GRBK 方法；当 $\theta = 1$时，它变为 MWRBK 方法。图 4.3 给出了$\delta_{k,\theta}$ 关于参数θ的示意图，并展示了 RGRBK、GRBK 和 MWRBK 方法之间的关系。

表 4.2 本节中用于求解 $Ax = b$ 和 $AXB = C$的方法所采用的指标选择准则。

$Ax = b$	GRK [2]	$\frac{(r^k_{i_k:})^2}{\|A_{i_k:}\|^2} \geq \frac{1}{2}\max_i \frac{(r^k_{i,:})^2}{\|A_{i:}\|^2} + \frac{1}{2}\frac{\|r^k\|_F^2}{\|A\|_F^2}$
	RGRK [3]	$\frac{(r^k_{i_k:})^2}{\|A_{i_k:}\|^2} \geq \theta$ 最大值 $\frac{(r^k_{i,:})^2}{\|A_{i:}\|^2} + (1-\theta)\frac{\|r^k\|_F^2}{\|A\|_F^2}$
	MWRK [11, 20]	$\frac{(r^k_{i_k:})^2}{\|A_{i_k:}\|^2} = \max_i \frac{(r^k_{i,:})^2}{\|A_{i:}\|^2}$
$AXB = C$	GRBK（算法5）	$\frac{\|R^k_{i_k:}\|^2}{\|A_{i_k:}\|^2} \geq \frac{1}{2}\max_i \frac{\|R^k_{i,:}\|^2}{\|A_{i:}\|^2} + \frac{1}{2}\frac{\|R^k\|_F^2}{\|A\|_F^2}$
	RGRBK（算法 6）	$\frac{\|R^k_{i_k:}\|^2}{\|A_{i_k:}\|^2} \geq \theta$ 最大值 $\frac{\|R^k_{i,:}\|^2}{\|A_{i:}\|^2} + (1-\theta)\frac{\|R^k\|_F^2}{\|A\|_F^2}$
	MWRBK（算法7）	$\frac{\|R^k_{i_k:}\|^2}{\|A_{i_k:}\|^2} =$最大值 $\frac{\|R^k_{i,:}\|^2}{\|A_{i:}\|^2}$

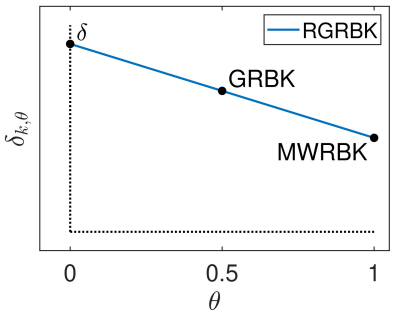


图 4.3 关于 $\delta_{k,\ \theta}$ 相对于 θ 的示意图，以及 RGRBK、GRBK 和 MWRBK 方法之间的关系。

在表4.4中，我们总结了求解矩阵方程（1.1）的各种方法的每次迭代计算复杂度。除 GI [10],外，所有算法都是涉及秩一更新的行作用方法；另见表 1.1。

表 4.4 求解 $AXB = C$ 的各种方法的每次迭代计算复杂度。

方法	复杂度
BK（算法 2），RBK（算法 1 [32]）	$\mathcal{O}(q(p+n))$
BK（算法 3）	$\mathcal{O}(q(p+n))$
BK（算法 4）	$\mathcal{O}(pq)$
GRBK、RGRBK、MWRBK（算法5、6、7）	$\mathcal{O}(pq+qn+mp+mn)$
RGRK [31],MWRK [31]	$\mathcal{O}(pq+qn+mp+mn)$
GI [10]	$\min(\mathcal{O}(mq(p+n))\ \ \mathcal{O}(pn(m+q)))$

5 数值示例

本节提供了一些数值示例，以验证主要理论发现。所有实验均使用 MATLAB R2021b，在配备 Intel(R) Core(TM) i5-12500H CPU @ 2.50GHz 的笔记本电脑上进行。我们从迭代次数（记为'IT'）、CPU 时间（秒，记为'CPU'）和相对解误差（记为'RSE'）这几个方面考察各方法。在每项实验中，随机化方法（GRBK、RGRBK 和 MWRBK）均独立运行 20 次。对于随机化方法，IT 和 CPU 的数值为 20 次试验的平均值。我们还给出了基于 20 次独立实验得到的 CPU 标准差（记为'SD'）和范围（记为'Range'）。根据第3 节和第4节的讨论，我们取参数 $\alpha = \|B\|^{-2}$。此外，如第4.3 节所述，当 $\theta \in (\frac{1}{2}, 1)$ 时，RGRBK 方法的收敛因子上界小于 GRBK 方法；因此，我们重点关注这种情况。

所有实验的初始猜测 X^0 均为零矩阵。这是 Kaczmarz 型方法中的常见选择，因为它往往能给出良好的近似结果。此外，这满足各定理的要求（即值域$(X_0) \subseteq$ 值域(A^T)以及值域$(X_0^T) \subseteq$ 值域(B)），从而使所有方法都收敛到最小范数解 $X_* = A^+CB^+$。被选作测试矩阵的系数矩阵 A 和 B 包括来自佛罗里达大学稀疏矩阵集 [7]的稀疏矩阵，以及由函数randn随机生成的稠密矩阵。这些矩阵也曾用于 [2, 3, 11, 32, 12]。矩阵 X 由randn(p,q)生成。为构造相容系统，矩阵 C 由 $C = AXB$ 生成。我们给出系数矩阵的大小、秩和密度，其中密度定义为

$$\text{density} := \text{number of nonzeros of an } m \times n \text{ matrix} \,/\, (m\,n).$$

现在我们简要讨论可能的停止准则。对所有方法而言，一条实用规则可以基于秩一更新的范数（参见表4.4）：

$$\|X^{k+1} - X^k\| = \alpha \, \|R^k_{i_k,:} B^T\| \,/\, \|A_{i_k,:}\|.$$

另一种自然的停止准则是使用 $\|R^k\|$，尤其是在算法5–7中，此时残差 R^k 已经可以获得。不过，我们不详细比较这些规则，而是着重考察各方法的表现，并考虑解误差：

$$\text{RSE} := \|X^k - X_*\|_F \,/\, \|X_*\|_F.$$

例5.1。本例中，我们分析三种 BK 方法（即算法2、3和4）的收敛行为。我们首先考察算法2的收敛性。对于相容矩阵方程（1.1），系数矩阵可分为九种不同的配置。具体而言，当 A 为行满秩、列满秩或秩亏时，B 也可以是行满秩、列满秩或秩亏。表5.1 列出了用于评估算法2性能的九组相应测试矩阵。图2 给出了算法5.2的收敛曲线。表5.3总结了测试矩阵的维度，并报告了迭代次数和中央处理器的结果。这些结果表明，算法2 似乎适用于各种问题规模，包括超定和欠定系统，且通常呈现线性渐近收敛。每次迭代所需时间似乎与系数矩阵的稀疏性有自然关联。例如，集合1和集合3中的矩阵相对稠密，因此每次迭代的计算时间最长。

接下来，我们研究算法3 和4的收敛行为。我们从表 B 中选取集合 5（算法 3的满列秩 B）和集合 1（算法 4的满行秩 α）。图5.1展示了谱半径随5.4 变化的曲线、不同 α对应的收敛图（相对误差与迭代次数的关系），以及 ρ^k 和相对误差随迭代次数变化的曲线。表5.5给出了图5.4中间一行对应的迭代次数和中央处理器时间。请注意，在图5.4 和表5.5中，"迭代次数"表示扫描次数，而在其他所有表格中

表 5.1 示例中使用的测试矩阵 5.1，其中测试了各种情形： $m > p$ 和 $m < p$； $p > q$ 和 $p < q$； $q > n$ 和 $q < n$。

矩阵	集合 1		集合 2		集合3	
	A	B	A	B	A	B
Name	bibd_11_5	bibd_12_4	bibd_12_4	ash219	bibd_11_5	n3c6-b2
Size	55 × 462	66 × 495	66 × 495	219 × 85	55 × 462	455 × 105
Rank	55	66	66	85	55	91
密度	18.18%	9.09%	9。09%	2.35%	18。18%	2.86%
	集合 4		集合 5		集合 6	
Name	ash219	bibd_12_4	ash219	ash331	ash219	n3c6-b1
Size	219 × 85	66 × 495	219 × 85	331 × 104	219 × 85	105 × 105
Rank	85	66	85	104	85	14
密度	2.35%	9.09%	2.35%	1.92%	2.35%	1.90%
	集合 7		集合 8		集合 9	
Name	n3c6-b1	lp_grow7	n3c6-b2	ash219	n3c6-b2	cis-n4c6-b1
Size	105 × 105	140 × 301	455 × 105	219 × 85	455 × 105	210 × 21
Rank	14	140	91	85	91	20
密度	1.90%	6.20%	2.86%	2.35%	2.86%	9.52%

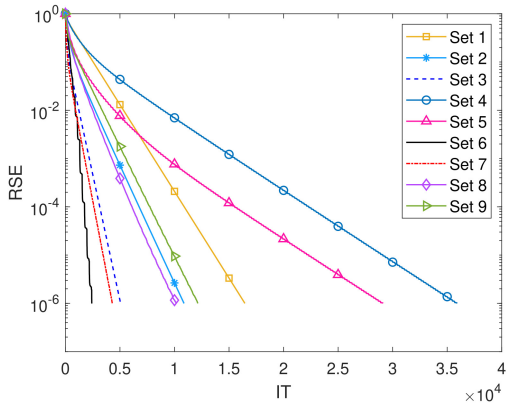


图 5.2 算法的收敛曲线 2 ，展示了表中所列测试矩阵的相对误差与迭代次数的关系5.1。

表 5.3 迭代次数、中央处理器和 CPU/IT 的结果，对应于图5.2。

Set	1	2	3	4	5
IT	16444	10855	5029	35893	27139
CPU	15.98	2.46	5.30	4.41	2.15
CPU/IT	$9.7 \cdot 10^{-4}$	$2.3 \cdot 10^{-4}$	$1.1 \cdot 10^{-3}$	$1.2 \cdot 10^{-4}$	$7.9 \cdot 10^{-5}$
Set	6	7	8	9	
IT	2427	4162	10363	12384	
CPU	0.08	0.53	0.60	0.66	
CPU/IT	$3.4 \cdot 10^{-5}$	$1.3 \cdot 10^{-4}$	$5.8 \cdot 10^{-5}$	$5.3 \cdot 10^{-5}$	

和图中，“迭代次数”指迭代步数。这些结果证实， ρ较小时，方法的表现更好。从谱半径曲线可以看出，当 $\rho < 1$ $\alpha \in (0, 2)$时，这与定理3.5 和3.6一致。

例 5.2. 在本例中，我们比较 RBK [32],GRBK、RGRBK 和 MWRBK 方法的收敛行为。对系数矩阵 A 和 B 的三种配置进行了测试：(1) 满列秩 A 与满行秩 B （表5.6中的集合 1 和 2），(2) 满行秩 A 与满列秩 B （表 5.6中的集合 3 和 4），以及 (3) 秩亏 A 和 B （表 5.6中的集合 5 和 6）。

表5.7 使用表5.6中的测试系数矩阵，给出了迭代次数、中央处理器时间、标准差、范围和中央处理器时间/迭代次数的比较结果。如表5.7所示，尽管 RBK 方法的平均单次迭代时间最短，但 GRBK、RGRBK 和 MWRBK 所需的迭代次数更少，总计算时间也更短。在

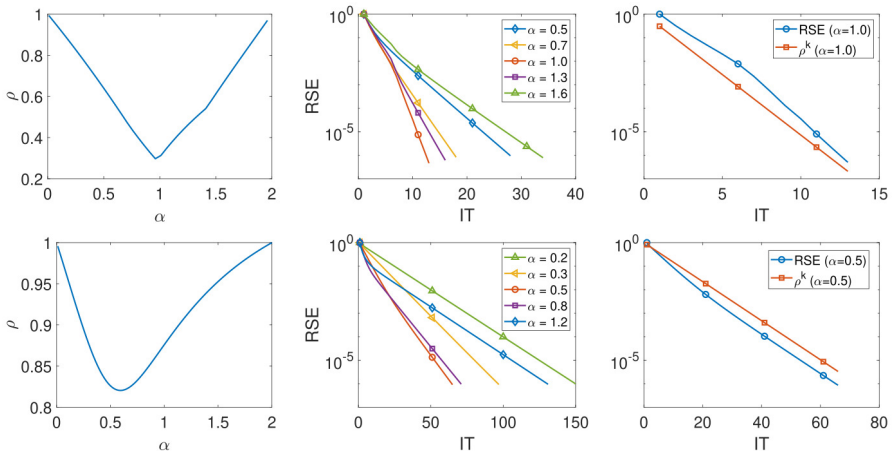


图 5.4 对于集合 5（上排）和集合 1（下排）：不同 α对应的谱半径、收敛情况（相对误差与迭代次数的关系），以及 ρ^k 与相对误差和迭代次数关系图，分别对应算法 3 （上排）和 4 （下排）。注意，在这些情况下， *upper bound* 在 ρ 上的最小值出现在 $\alpha = 1$ 处，但实际的极小值点 α 可能与该值不同。仅在此图中，迭代次数指扫描次数（而非行作用步数）。

表 5.5 图 5.4中间列所示的迭代次数和中央处理器时间结果。

算法 3			算法 4		
α	迭代次数	中央处理器	α	IT	CPU
0.5	27	0.38	0.2	150	0.50
0.7	17	0.23	0.3	96	0.30
1.0	12	0.16	0.5	65	0.21
1.3	15	0.20	0.8	71	0.23
1.6	33	0.43	1.2	131	0.41

表 5.6 示例 5.2中使用的系数矩阵。

矩阵	集合 1		集合 2	
	A	B	A	B
Name	ash608	bibd_15_3_	randn(455, 20)	randn(80, 320)
Size	608 × 188	105 × 455	455 × 20	80 × 320
Rank	188	105	20	80
密度	1.06%	2.86%	100%	100%
	集合3		集合 4	
Name	lp_afiro	ash219	randn(35, 60)	randn(80, 20)
Size	27 × 51	219 × 85	35 × 60	80 × 20
Rank	27	85	35	20
密度	7.41%	2.35%	100%	100%
	集合 5		集合 6	
Name	flower_4_1_	n3c6-b2	A=randn(265, 25); A=[A, A]	B=randn(10, 345); B=[B; B]
Size	121 × 129	455 × 105	265 × 50	20 × 345
Rank	108	91	25	10
密度	2.47%	2.86%	100%	100%

此外，随机算法的标准差较小，其中 GRBK 和 RGRBK 的标准差通常略低于 RBK；这是自然的，因为它们只从较少的行中进行选择。

6 彩色图像复原的一个应用

许多成像系统，例如磁共振成像、计算机断层扫描等，都可以用线性方程组（1.3）来建模 [16, 17]。类似地，矩阵方程（1.1在某些彩色图像复原问题中也发挥着重要作用 [35]。

表 5.7 表 5.6 中测试系数矩阵的 RBK、GRBK、RGRBK 和 MWRBK 方法的数值结果 5.6（其中 m、 p、 q、 n 为相应取值）。

方法	集合 1 (608, 188, 105, 455)					集合 2 (455, 20, 80, 320)				
	IT	CPU 标准差		范围	CPU/IT	IT	CPU 标准差		范围	CPU/IT
RBK	33891	29.9	2.17	[25.5, 33.3]	$8.8 \cdot 10^{-4}$	2589	1.50	0.05	[1.38, 1.57]	$5.8 \cdot 10^{-4}$
GRBK	12736	21.9	0.04	[21.9, 22.1]	$1.7 \cdot 10^{-3}$	1417	1.33	0.01	[1.30, 1.35]	$9.4 \cdot 10^{-4}$
RGRBK	12715	21.7	0.04	[21.6, 21.7]	$1.7 \cdot 10^{-3}$	1399	1.28	0.02	[1.27, 1.34]	$9.2 \cdot 10^{-4}$
MWRBK	12711	17.1	–	–	$1.3 \cdot 10^{-3}$	1398	1.18	–	–	$8.4 \cdot 10^{-4}$
	集合3 (27, 51, 219, 850)					集合 4 (35, 60, 80, 20)				
RBK	30897	1.87	0.34	[1.62, 2.65]	$6.1 \cdot 10^{-5}$	32702	0.46	0.05	[0.40, 0.60]	$1.4 \cdot 10^{-5}$
GRBK	13229	0.89	0.02	[0.86, 0.96]	$6.7 \cdot 10^{-5}$	23519	0.46	0.02	[0.43, 0.49]	$2.0 \cdot 10^{-5}$
RGRBK	13219	0.88	0.02	[0.85, 0.93]	$6.6 \cdot 10^{-5}$	23512	0.46	0.03	[0.42, 0.53]	$2.0 \cdot 10^{-5}$
MWRBK	13213	0.79	–	–	$6.0 \cdot 10^{-5}$	23512	0.37	–	–	$1.6 \cdot 10^{-5}$
	集合 5 (121, 129, 455, 105)					集合 6 (265, 50, 20, 345)				
RBK	32859	8.03	0.14	[7.87, 8.42]	$2.4 \cdot 10^{-4}$	1043	0.16	0.01	[0.14, 0.18]	$1.5 \cdot 10^{-4}$
GRBK	11071	3.64	0.04	[3.58, 3.76]	$3.3 \cdot 10^{-4}$	388	0.14	0.01	[0.12, 0.16]	$3.6 \cdot 10^{-4}$
RGRBK	11029	3.42	0.03	[3.37, 3.48]	$3.1 \cdot 10^{-4}$	363	0.13	0.01	[0.13, 0.15]	$3.7 \cdot 10^{-4}$
MWRBK	11012	2.85	–	–	$2.6 \cdot 10^{-4}$	355	0.11	–	–	$3.1 \cdot 10^{-4}$

我们首先介绍彩色图像复原的前向模型；另见 [35]。期望的原始彩色图像 $\mathcal{X}$（观测彩色图像 $\mathcal{C}$）可表示为一个大小为 $m \times n \times 3$ 的三维数组，其中红色（R）、绿色（G）和蓝色（B）通道分别对应三个矩阵 $X_{\mathrm{r}} := \mathcal{X}(:, :, 1)$、 $X_{\mathrm{g}} := \mathcal{X}(:, :, 2)$、$X_{\mathrm{b}} := \mathcal{X}(:, :, 3)$（观测图像也同理，分别为 $C_{\mathrm{r}} := \mathcal{C}(:, :, 1)$、 $C_{\mathrm{g}} := \mathcal{C}(:, :, 2)$、 $C_{\mathrm{b}} := \mathcal{C}(:, :, 3)$）。令$X = (\mathrm{vec}(X_{\mathrm{r}}), \mathrm{vec}(X_{\mathrm{g}}), \mathrm{vec}(X_{\mathrm{b}}))$ 和 $C = (\mathrm{vec}(C_{\mathrm{r}}), \mathrm{vec}(C_{\mathrm{g}}), \mathrm{vec}(C_{\mathrm{b}}))$，前向模型可表示为 $C = AXA_{\mathrm{c}}^T + E$，其中 $A \in \mathbb{R}^{mn \times mn}$ 表示通道内模糊， $A_{\mathrm{c}} \in \mathbb{R}^{3 \times 3}$ 表示通道间模糊，$E \in \mathbb{R}^{mn \times 3}$ 表示加性噪声。理想情况下，如果模糊图像的生成过程中没有噪声，前向模型就变为（1.2）。本文仅考虑这种无噪声情形。

我们选取大小为 $92 \times 92 \times 3$ 的人脸 、大小为 $96 \times 96 \times 3$ 的鸟 以及大小为 $125 \times 120 \times 3$ 的mandril 作为测试图像，如图6.1顶行所示，以考察以下迭代方法的性能：GI [10], RBK [32], 以及我们提出的 BK、GRBK、RGRBK 和 MWRBK 方法。为获得模糊图像，使用 Matlab 函数fspecial 生成大小为 5、标准差为 6 的旋转对称高斯低通滤波器，以模拟高斯滤波器造成的模糊图像。通道间模糊矩阵 A_{c} 刻画不同颜色通道（即 RGB）之间可能因模糊而产生的关联。在典型系统中，一个通道的主要贡献来自其自身，这意味着 $(A_{\mathrm{c}})_{i,i} > (A_{\mathrm{c}})_{i,j}$对于 $i \neq j$。为确保总强度得以保留，约束 A_{c} 的每一行之和为 1，即 $\sum_j (A_{\mathrm{c}})_{i,j} = 1$。在我们的示例中，我们选取如下（相当典型的）通道间模糊矩阵：

$$A_{\mathrm{c}} = \begin{bmatrix} 0.90 & 0.05 & 0.05 \\ 0.00 & 0.90 & 0.10 \\ 0.05 & 0.10 & 0.85 \end{bmatrix}. \tag{6.1}$$

图 6.1 的最下排显示了模糊图像，以及它们对应的峰值信噪比（PSNR）值，分别为 19.31、17.52 和 18.86。



图 6.1 测试图像（上排）和模糊图像（下排）。

表 6.2 概述了本节实验的规模。注意，对于此应用， $m = p$ （即， A 为方阵），并且 $q = n$ （B 为方阵），而 $p \gg q$（X 为高而窄的矩阵）。我们指出，在这一具体实验中，矩阵 $B = A_c^T$是一个较小的 3×3 非奇异矩阵，因此直接求逆是自然的，并且能够得到自然的解，如方法 4所示。我们在实验 6.2中测试了这种方法，但同时要强调的是，我们已经开发出能够求解（1.1）的迭代次数 A 和 B 任意大小的方法。

表 6.2 第 6 节实验规模概览，以及解中的自由度（dofs） X （等于 pq）。

实验	$m = p$	$q = n$	dofs
face	8464	3	25392
bird	9216	3	27648
mandril	15000	3	45000

实验 6.1。 在此实验中，我们比较 GI [10],BK（算法 2）、RBK [32] （算法 1）、GRBK（算法 5）、RGRBK（算法 6）和 MWRBK（算法 7）方法恢复的结果。除 GI 以外，所有方法每次迭代处理一行。GI 方法每次迭代使用所有行，这可能会带来更高的计算开销，本实验也证实了这一点。

在图 6.3 中，我们绘制了这些方法的峰值信噪比与耗时的关系。在此实验中，随机化方法只运行一次。为了达到相近的峰值信噪比，GI、BK 和 RBK 方法所需的计算时间比 GRBK、RGRBK 和 MWRBK 方法更长，其中 GI 方法最耗时。正如第 4 节所述，增大θ 从 0.5（GRBK）经由区间 (0.5，1) 中的一个中间值（RGRBK）增至 1（MWRBK），会提升性能。此外，随机化 RBK 仍比确定性 BK 更快，这说明选取范数较大的行是有益的。如果我们预先缩放（1.1），使所有行的范数相等，那么预计 BK 和 RBK 的表现会相似。

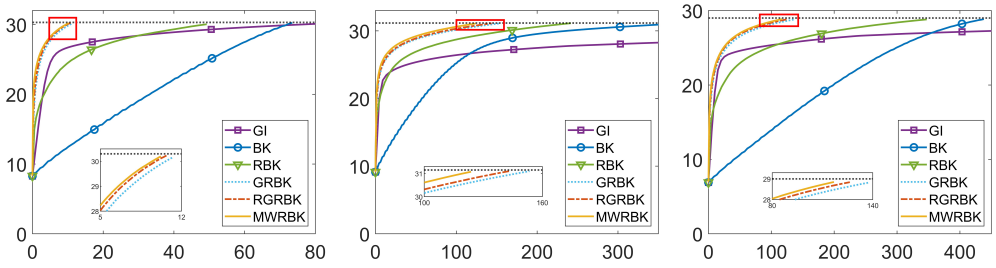


图 6.3 使用不同方法复原的人脸（左）、鸟（中）和 mandril（右）图像的峰值信噪比与中央处理器之间的关系。

实验 6.2。 在本实验中，我们比较通过 GI [10], BK（算法 2）、RBK [32], GRBK、RGRBK 和 MWRBK 方法获得的恢复结果。为进行公平比较并准确反映各种方法的典型表现，我们在停止准则中使用精确解，并在 RSE $\leq 8 \cdot 10^{-2}$时终止迭代。不同算法获得的复原图像的峰值信噪比值相同：30.23（人脸）、31.08（鸟）和 28.85（mandril），分别对应于上述三种图像。表6.4 给出了数值结果，图 6.5 展示了通过 MWRBK 方法获得的复原图像。我们观察到，GI 方法每次迭代和总计算时间最长，这与其立方计算复杂度相一致，而其他算法的计算复杂度为二次。尽管 GI 所需的迭代次数因情况而异，但总体而言，它似乎是一种竞争力较弱的方法。

尽管 GRBK、RGRBK 和 MWRBK 的每次迭代 CPU 时间（CPU/IT）高于 BK 和 RBK，但它们所需的迭代次数显著更少，因此总计算时间短于 BK 和 RBK。这些结果表明，它们的总体收敛速度快于 BK 和 RBK。

随着问题规模增大，随机化算法（RBK、GRBK 和 RGRBK）的标准差略有增加。例如，在人脸实验中，随机化方法的标准差均小于 2；而在狒狒实验中，标准差略有上升，但仍低于 7。

此外，在未报告的实验中，我们比较了三种 BK 方法（算法 2、3 和 4）的表现。对于这个 3×3 矩阵 A_c，无论 A_c 与单位矩阵的接近程度如何，这些 BK 方法的迭代次数和运行时间通常都非常相似。命题 3.7 表明，鉴于 A_c 是非奇异的，算法 3 和4 等价；我们的结果也确实证实，迭代次数相同。

表 6.4 各种方法复原图像的峰值信噪比、迭代次数、中央处理器时间、标准差、范围和 CPU/IT；除 GI 外，所有方法每次迭代均采用秩一更新。

方法	face					
	PSNR	IT	CPU	SD	范围	CPU/IT
GI	30.23	37689	113.8	–	–	$3.0 \cdot 10^{-3}$
BK	30.23	281401	105.2	–	–	$3.7 \cdot 10^{-4}$
RBK	30.23	139442	67.0	1.73	[64.1, 71.3]	$4.8 \cdot 10^{-4}$
GRBK	30.23	11534	19.0	0.58	[18.1, 20.0]	$1.7 \cdot 10^{-3}$
RGRBK	30.23	11191	17.8	0.70	[16.7, 18.9]	$1.6 \cdot 10^{-3}$
MWRBK	30.23	12045	17.4	–	–	$1.5 \cdot 10^{-3}$
	bird					
GI	31.08	602495	1919.3	–	–	$3.2 \cdot 10^{-3}$
BK	31.08	821648	330.0	–	–	$4.0 \cdot 10^{-4}$
RBK	31.08	390479	207.8	4.11	[200.5, 217.2]	$5.3 \cdot 10^{-4}$
GRBK	31.08	97871	167.8	3.09	[163.1, 173.2]	$1.7 \cdot 10^{-3}$
RGRBK	31.08	96407	158.4	2.10	[154.6, 162.1]	$1.6 \cdot 10^{-3}$
MWRBK	31.08	97475	147.2	–	–	$1.5 \cdot 10^{-3}$
	mandril					
GI	28.85	254358	1224.6	–	–	$4.8 \cdot 10^{-3}$
BK	28.85	595118	474.2	–	–	$8.0 \cdot 10^{-4}$
RBK	28.85	375388	314.1	6.41	[299.1, 328.7]	$8.4 \cdot 10^{-4}$
GRBK	28.85	59750	153.8	2.29	[150.6, 158.3]	$2.6 \cdot 10^{-3}$
RGRBK	28.85	58563	145.1	2.52	[141.0, 149.4]	$2.5 \cdot 10^{-3}$
MWRBK	28.85	58522	123.8	–	–	$2.1 \cdot 10^{-3}$

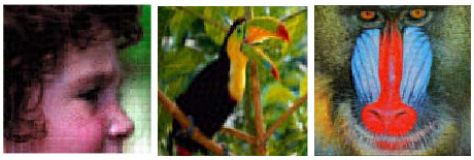


图 6.5 使用 MWRBK 方法对实验6.2获得的复原图像。

根据表6.2 和6.4中的结果，我们观察到，三幅测试图像的自由度与每次迭代计算成本（CPU/IT）成正比。例如，在 RBK 方法中，CPU/IT 从人脸 的$8.4 \cdot 10^{-4}$ 增加到 mandril 的 $4.8 \cdot 10^{-4}$，比例约为 1.75。mandril 的自由度约为人脸的 1.77 倍。这一相容的趋势表明，问题规模会影响每次迭代的计算成本。示例鸟所需的迭代次数最多，这可能是因为图中有许多边缘。迭代次数与通道内模糊矩阵 A 的条件数之间没有明显关系，因为鸟 对应的矩阵在三个示例中条件数最小。

最后，谈谈这些方法在较大图像上的缩放情况。我们采用人脸的较大 140×140 和 280×280 版本；其自由度分别为 $140^2 \cdot 3 = 58800$ 和 $280^2 \cdot 3 = 235200$，相比原始 92×92 图像，未知数约多 2.3 倍和 9.3 倍。CPU/IT 随自由度严格线性增长。较大图像所需的迭代次数有所增加，这可能是因为图像包含更多细节。对于 140×140 人脸，迭代次数约为 19000，而非 12000。总之，CPU/IT 与自由度成正比，而迭代次数似乎取决于图像的复杂程度。

7 结论

我们研究了确定性和随机 Kaczmarz 方法，以求解一致矩阵方程 (1.1)；该方程常见于许多应用中。除算法 3 和算法 4 要求 B 具有满列秩或满行秩外，我们的方法适用于任意大小的 A 和 B （即超定和欠定情形）。 [10] 中的梯度型 GI 方法在每次迭代中都使用满秩更新。 [31] 中的方法每一步都使用 C 的一个元素和 B 的一列进行秩一加法更新。我们提出的所有方法还会在每次迭代中使用 C 的一行和整个矩阵 B进行秩一更新。如表 4.4 所总结， [31] 中的方法和本文提出的方法关于矩阵维数 m、 n、 p、 q的计算成本均为二次，而 [10] 中的 GI 方法的复杂度为三次；另请参见第 6 节。

我们首先在第 3.1 节研究了一种（确定性、循环式）BK 方法（算法 2）。这是将标准 Kaczmarz 方法直接扩展到矩阵方程 (1.1)，也是对 [32] 的 RBK 方法进行确定性（循环式）改造的版本。我们证明了，当 $0 < \alpha < 2\,\|B\|^{-2}$ 时，该方法收敛到解 $X_*^0 (= A^+CB^+ + X^0 - A^+AX^0BB^+)$。我们还考虑了该 BK 方法的若干变体，分别假设 B 满列秩（第 3.2 节，算法 3）或满行秩（第 3.3 节，算法 4）。这些附加约束使我们能够推导出 BK 方法一次扫描（循环）的矩阵表达式。请注意，算法 2 对 B 没有任何要求，但后两种方法要求 B 满行秩或满列秩（非奇异是其中的特殊情况），且规模适中。此外，如命题 3.7 所示，当系数矩阵 B 非奇异时，后两种方法是等价的。在实验 6.2 中，我们确认了这三种 BK 方法在 B 为 3×3 时表现出相似的收敛行为。

接下来，我们在第4节提出并研究了 GRBK 方法（算法5）及其松弛与确定性变体：RGRBK（算法6）和 MWRBK（算法7），它们根据残差范数选择行。根据第6节的实验，计算该残差范数所需的额外开销是值得的；此外，残差还可作为合适的停止准则。我们已证明，当 A^+CB^+ 时，$0 < \alpha < 2\,\|B\|^{-2}$它们收敛到唯一最小范数解。根据第3 节和第4节的收敛分析，所提出方法的收敛因子上界在$\alpha = \|B\|^{-2}$时达到最小值。如第4.3节所示，GRBK、RGRBK 和 MWRBK 方法的收敛因子上界不大于 RBK 方法的上界。因此，可以合理预期它们达到收敛所需的迭代次数可能更少。第 5 节和第6 节的实验表明，事实确实如此。当$\theta = \frac{1}{2}$时，RGRBK 方法退化为 GRBK 方法；当$\theta = 1$时，则退化为 MWRBK 方法，如图4.3 和表4.2所示。其收敛因子上界随 θ 增大而减小。因此，在 $\theta = 1$时，收敛因子上界达到最小值，此时 RGRBK 与 MWRBK 相同。

最后，基于理论分析和实验结果，所提出方法的总体性能可总结如下。BK 方法渐近收敛（见定理 3.2）；其收敛速度可能对 A 中行的排序较为敏感。它可能（远）快于其随机化对应方法（RBK），但也可能更慢。若对行采用随机初始排序，BK 方法的表现通常与随机方法 RBK 相似，且没有 RBK 的方差这一缺点。我们注意到，尽管其收敛往往慢于其他方法（见第6节），BK 对于通过与随机化（RBK）及贪心方法的比较来理解行选择的影响仍有价值

（GRBK、RGRBK、MWRBK）对应方法。贪心扩展（GRBK、RGRBK 和 MWRBK）的收敛始终快于 RBK，这在第 5 节和第 6 节的数值测试中也得到了观察。尽管贪心方法每次迭代比 BK 方法和 RBK 方法更慢，但它们使用的迭代次数更少，运行时间也更短。随机算法（RBK、GRBK 和 RGRBK）在实验中的标准差较小。总体而言，理论和实验均表明，（确定性）MWRBK 方法（算法 7）在迭代次数和计算时间方面似乎是所考虑方法中最具吸引力的。最后，我们指出，本文考虑的是不含噪声的矩阵方程。Kaczmarz 型方法对于含噪声的右端项 C 也可能特别适用；这将留待后续工作。

致谢。 我们非常感谢三位专家审稿人提出的宝贵建议。本研究得到了中央高校基本科研业务费（编号：2024YJS111）和国家留学基金管理委员会资助（编号：202407090119）的支持。

利益冲突。 不适用。

参考文献

[1]Ansorge, R.: Cimmino方法与Kaczmarz方法在奇异及正则方程组求解中的联系. Computing, 33, 367–375 (1984).[2] Bai, Z.Z., Wu, W.T.: 求解大型稀疏线性方程组的贪婪随机 Kaczmarz 方法. SIAM J. Sci. Comput. 40(1), A592–A606 (2018).[3] Bai, Z.Z., Wu, W.T.: 求解大型稀疏线性方程组的松弛贪心随机Kaczmarz方法. Appl. Math. Lett. 83, 21–26 (2018).[4]Bottou, L.: 基于随机梯度下降的大规模机器学习. Proceedings of COMPSTAT, 177–186 (2010).[5]Chen, X.M.: Kaczmarz算法、行作用法与统计学习算法. Frames Harmon. Anal. 706, 115–127 (2018).[6] Costa, O.L.V., Fragoso, M.D.: 具有马尔可夫跳变参数的离散时间线性系统的稳定性结果. J. Math. Anal. Appl. 179(1), 154–178 (1993).[7] Davis, T.A., Hu, Y.F.: 佛罗里达大学稀疏矩阵集. ACM Trans. Math. Software, 38(1), 1–25 (2011).[8] Deng, Y.B., Bai, Z.Z., Gao, Y.H.: 两个相容矩阵方程的Hermitian最小范数解的迭代正交方向法. Numer. Linear Algebra Appl. 13, 801–823 (2006).[9] Derezi´nski, M., Needell, D., Rebrova, E., Yang J.: 具有超Krylov收敛的随机Kaczmarz方法. SIAM J. Matrix Anal. Appl. 46(4) (2025) https://doi.org/10.1137/25M1727187.[10] Ding, F.,Liu, P.X.,Ding, J.: 利用递阶辨识原理求广义Sylvester矩阵方程的迭代解. Appl. Math. Comput. 197, 41–50 (2008).[11] Du, K., Gao, H.: 最大加权残差Kaczmarz算法收敛速度的一个新理论估计. Numer. Math. Theor. Meth. Appl., 12(2): 627–639 (2019).[12] Du, K., Ruan, C.C., Sun, X.H.: 矩阵最小二乘问题的随机块坐标下降算法的收敛性. Appl. Math. Lett. 124, 107689 (2022).[13]Elfving, T.: 相容与不相容线性方程的分块迭代法. Numer. Math. 35(1), 1–12 (1980).[14] Elfving, T., Nikazad, T.: 一类分块迭代法的性质. Inverse Probl. 25, 115011 (2009).[15] Ferreira, I., Acebr´on, J., Monteiro, J.: 一类迭代行作用方法综述：Kaczmarz 方法. Numer. Math. https://doi.org/10.1007/s11075- 024-01945-2.

[16] Herman, G.T., Meyer, L.B.: 代数重建技术可以在计算上实现高效. IEEE Trans. Med. Imaging, 12(3), 600–609 (1993).[17]Jiang, M., Wang, G.: 图像重建迭代算法的收敛性研究. IEEE Trans. Image Process. 22(5), 569–579 (2003).[18] Kaczmarz, S.: Angen¨aherte Aufl¨osung von Systemen linearer Gleichungen（线性方程组的近似解）. Bull. Int. Acad. Polon. Sci. Let. A, 35, 355–357 (1937).[19] Landweber, L.: 第一类Fredholm积分方程的迭代公式. Amer. J. Math. 73(3), 615–624 (1951).[20] McCormick, S.F.: 希尔伯特空间中求解线性方程与最小二乘问题的Kaczmarz方法及行正交化方法. Indiana Univ. Math. J. 26, 1137–1150 (1977).[21] Needell, D., 含噪线性系统的随机Kaczmarz求解器. BIT Numer. Math. 50, 395–403 (2010).[22] Needell, D., Srebro, N., Ward, R.: 随机梯度下降、加权抽样与随机Kaczmarz算法. Math. Program., Ser. A, 155, 549–573 (2016).[23] Niu, Y.Q., Zheng, B.: 求解矩阵方程 $AXB = C$的随机块Kaczmarz算法. J. Comput. Appl. Math. 460, 116421 (2025).[24] Penrose, R.: 矩阵的广义逆. Math. Proc. Cambridge Philos. Soc. 51, 406–413 (1955).[25] Shafiei, S.G., Hajarian, M.: 发展用于求解Sylvester矩阵方程的Kaczmarz方法. J. Franklin Inst. 359(16), 8991–9005 (2022).[26] Strohmer,T., Vershynin,R.: 具有指数收敛性的线性系统随机Kaczmarz算法. J. Fourier Anal. Appl. 15, 262–278 (2009).[27] Su, Y.S., Han, D.R., Zeng, Y., Xie, J.X.: 求解线性系统的贪婪多步惯性随机Kaczmarz方法. Calcolo, 61:68 (2024).[28] Tian, Z.L., Tian, M.Y., Liu, Z.Y., Xu, T.Y.: 矩阵方程 $AXB = C$的Jacobi与Gauss–Seidel型迭代方法. Appl. Math. Comput. 292, 63–75 (2017).[29] Waegeman, W., Dembczy´nski, K., H¨ullermeier, E.: 多目标预测：问题与方法的统一观点. Data Min. Knowl. Discov. 33(2), 293–324 (2019).[30] Wang, X., Li, Y., Dai, L.: 线性矩阵方程 $AXB = C$的Hermitian与斜Hermitian分裂迭代方法. Comput. Math. Appl. 65(4), 657–664 (2013).[31] Wu, N.C., Liu, C.Z., Zuo, Q.: 基于松弛贪婪选择求解矩阵方程 $AXB = C$的Kaczmarz方法. J. Comput. Appl. Math. 413, 114374 (2022).[32] Xing, L.L., Bao, W.D., Li, W.G.: 求解矩阵方程的随机块Kaczmarz算法的收敛性. Mathematics, 11, 4554 (2023).[33] Xu, S.F, Gao, L., Zhang, P.W.: 数值线性代数. 北京大学出版社, 2000.[34] Zeng, X.L., Liang, S., Hong, Y.G., Chen, J.: 线性矩阵方程的分布式计算：优化视角. IEEE Trans. Automat. Contr. 65(5) 1858–1873 (2019).[35] Zhang, J.J., Nagy, J.G.: 一种有效的彩色图像复原交替方向乘子法. Appl. Numer. Math. 164, 43–56 (2021).