

地震波层析成像测试算例

黄子睿，朱延坤，邬嘉恺，秦加俊，曾炳荣，解淑涵，陈久威
同济大学数学科学学院寒假实践

版本：4.0

摘要：地震层析成像是勘探地球物理中的一门新兴技术，能够有效地还原地下介质的速度模型，是探测地下介质速度结构的重要研究手段。本文将结合图像处理技术，利用共轭梯度法和 Kaczmarz 类方法对地震层析成像的具体算例加以处理，比较不同方法的优缺点并进一步优化。

关键词：地震波 层析成像 数值分析 共轭梯度法 随机 Kaczmarz 方法

地震波层析成像是利用地震波相关数据反演地下介质的物理属性，对其逐层分析并进一步绘制出其图像的技术。[1] 20 世纪 70 年代，美国科学家 Chapman 通过实验证明，地球物理勘探涉及的 τ —P 变换与 Radon 变换本质相同，也就是说，医学上的 CT 技术、X 射线等能够应用到地球物理上，这为地震勘探中的反演方法提供了新思路。

地球层析成像首先研究井间速度结构，并用于测井勘探领域。1979 年，地球物理学家 Dines 和 Lytle 首次将层析成像用在论文标题上 (Computerized Geophysical Tomography)，他们通过多次数值模拟，并对检波点接受到的地震波信号进行处理，利用弯曲地震波射线，得到相应结果。1984 年，美国地球物理学家 Anderson 选取天然地震为研究对象，推演出地球的三维速度结构，然人们进一步了解地球密度构造以及地幔流体等地下介质结构。到了 80 年代，地球物理勘探学家协会 (SEG) 设立了地震层析成像的专题，其应用推广开来，研究对象也从天然地震扩展至人工地震，二者发展都相当可观。[2]

在具体的实施过程中，地震波层析成像可以划分为四个板块：建立初始模型 (在研究区域进行网格参数化)，正演数值模拟 (射线追踪、波动方程等)，反演及图形重建，结果评价。其中，反演以及图像重建技术可分为线性处理方法和非线性处理方法，而线性处理方法包括奇异值分解法，最小二乘法和共轭梯度法几种主要方法。

本文介绍的算例模拟中用到的主要计算方法包括：共轭梯度法 (CG)、随机 Kaczmarz 方法 (RK) 和贪婪 Kaczmarz 方法 (GK)。使用 AIR TOOL 工具箱中地震波层析成像测试算例，在图像上加噪声，使用线性方程组的求解方法恢复有噪声和无噪声的算例，并比较不同迭代方法的性能。进一步，使用预处理以及随机采样等技巧对算法进行优化，结合实践数据加以汇总分析。

1 主要计算方法

1.1 共轭梯度法

定义 1.1 (求解线性系统的投影方法) 对于 m 维子空间 \mathcal{K}, \mathcal{L} 和线性系统 $Ax = b, A \in \mathbb{R}^{n \times n}$. 一种求解 $Ax = b$ 近似解的一般投影方法是

$$\text{Find } \tilde{x} \in \mathcal{K}, \quad \text{s.t. } b - A\tilde{x} \perp \mathcal{L}$$

此外, 如果有初始估计 x_0 , 那么考虑

$$\text{Find } \tilde{x} \in x_0 + \mathcal{K}, \quad \text{s.t. } b - A\tilde{x} \perp \mathcal{L}$$

定义 1.2 (Krylov 子空间的定义) 至多 m 维的线性子空间

$$\mathcal{K}_m(A, v) = \text{span}\{v, Av, \dots, A^{m-1}v\}$$

被称为 *Krylov* 子空间

定义 1.3 (共轭梯度法) 考虑 A 对称正定时, 令投影算法中 $\mathcal{K}_m = \mathcal{L}_m = \mathcal{K}_m(A, r_0)$

- **算法 1:**

- 1、初始估计 $x_0, r_0 = b - Ax_0$
- 2、 $\mathcal{K}_m = \mathcal{L}_m = \mathcal{K}_m(A, r_0)$
- 3、求解 $x_m \in x_0 + \mathcal{K}_m, b - Ax_m \perp \mathcal{L}_m$

- **算法 2: 基于 LDL 分解的 CG 算法**

- 1、初始估计 $x_0, r_0 = b - Ax_0, \beta_0 = \|r_0\|_2, q_0 = 0, c_0 = 0$
- 2、For $k = 1, 2, \dots$
- 3、 $q_k = \frac{r_{k-1}}{\beta_{k-1}}, \alpha_k = q_k^T A q_k$
- 4、 $k = 1$ 时,

$$d_k = \alpha_k, \nu_k = \beta_0 / d_1, c_k = q_1$$

$k > 1$ 时,

$$l_{k-1} = \beta_{k-1} / d_{k-1}, d_k = \alpha_k - \beta_{k-1} l_{k-1}, \nu_k = -\beta_{k-1} \nu_{k-1} / d_k, c_k = q_k - l_{k-1} c_{k-1}$$

- 5、 $x_k = x_{k-1} + \nu_k c_k$
- 6、 $r_k = A q_k - \alpha_k q_k - \beta_{k-1} q_{k-1}$
- 7、 $\beta_k = \|r_k\|_2$
- 8、Endfor

- **算法 3: CG 方法的最终版本**

首先, 我们考虑下面一个迭代格式

1、初始估计 $x_0, r_0 = b - Ax_0, p_0 = r_0$

2、For $k = 0, 1, 2, \dots$

3、计算

$$\alpha_k = \arg \min_{\alpha} f(x_k + \alpha p_k), \quad f(x) = \frac{1}{2} x^T A x - b^T x$$

并更新 $x_{k+1} = x_k + \alpha_k p_k$

4、计算

$$\beta_k, \quad \text{s.t.} \quad (b - Ax_{k+1} + \beta p_k)^T A p_k = 0$$

并更新 $p_{k+1} = b - Ax_{k+1} + \beta_k p_k$

最后，我们得到了 CG 的最终版本

1、初始估计 $x_0, r_0 = b - Ax_0, p_0 = r_0$

2、For $k = 0, 1, 2, \dots$

3、 $w_k = A p_k$

4、 $\alpha_k = r_k^T r_k / p_k^T w_k$

5、 $x_{k+1} = x_k + \alpha_k p_k, \quad r_{k+1} = r_k - \alpha_k w_k$

6、 $\beta_k = r_{k+1}^T r_{k+1} / r_k^T r_k$

7、 $p_{k+1} = r_{k+1} + \beta_k p_k$

8、Endfor

设 λ_1 和 λ_n 分别表示 A 的最大特征值和最小特征值， x_* 为精确解，则由共轭梯度法得到的向量序列满足误差估计

$$\|x_k - x_*\|_A \leq 2\|x_0 - x_*\|_A \left[\frac{\sqrt{\lambda_1} - \sqrt{\lambda_n}}{\sqrt{\lambda_1} + \sqrt{\lambda_n}} \right]^k$$

理论上，如果所有计算都没有舍入误差的话，可以证明共轭梯度法最多经过 n 次迭代即可得到精确解 x_* 。但是，由于计算过程中不可避免有舍入误差，所以实际上将共轭梯度法作为一种迭代方法使用。

1.2 随机 Kaczmarz 方法

1.2.1 经典 Kaczmarz 方法

给定相容的线性方程组，波兰数学家 Stefan Kaczmarz 提出 Kaczmarz 方法，这是一种投影的思想，将每一步迭代得到的近似解正交投影到超平面 $a_{i_k} x = b_{i_k}$ 上。因为 a_{i_k} 是超平面 $a_{i_k} x = b_{i_k}$ 的法向量，故近似解 x_{k+1} 考虑如下迭代：

$$x_{k+1} = x_k + \mu a_{i_k},$$

其中

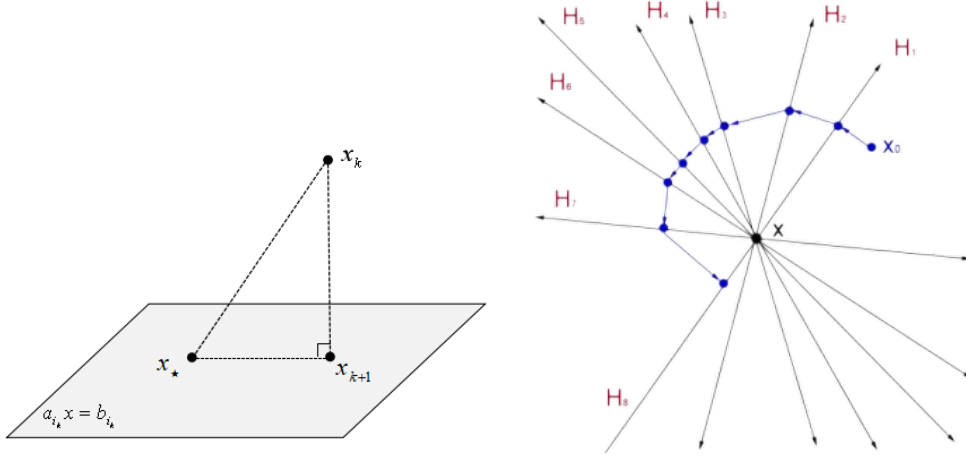
$$\mu = \frac{b_{i_k} - a_{i_k}^T x_k}{\|a_{i_k}\|_2^2},$$

假设初始近似值为 x_0 ，可以得到 Kaczmarz 迭代格式为:

$$x_{k+1} = x_k + \frac{b_{i_k} - a_{i_k}^T x_k}{\|a_{i_k}\|_2^2} a_{i_k}, \quad k = 0, 1, 2, \dots,$$

其中, a_{i_k} 为矩阵 A 的第 i_k 行, b_{i_k} 为向量 b 的第 i_k 个元素, $i_k = (k \bmod m) + 1$.

具体的投影过程为:



1.2.2 随机 Kaczmarz 方法

Strohmer 和 Vershynin[3] 在 2009 年提出, 当采用随机的方法、更确切的说是以概率 $\frac{\|a_{i_k}\|_2^2}{\|A\|_F^2}$ 而非给定的顺序扫描工作行时, 误差期望收敛速度可以达到指数收敛, 并且速率与系统中方程的数量无关。与此同时, 该方法也不需要知道矩阵 A 的所有元素。

定理 1.1 (收敛性) [3][4]

若线性方程组 $Ax = b$ 相容, 其中系数矩阵 $A \in \mathcal{R}^{mn}$ 且 $b \in \mathcal{R}^m$, 初始近似值 $x_0 \in \mathcal{R}(A^T)$ 。令 x_* 是线性方程组的解, 由随机 Kaczmarz 方法生成的迭代序列 $\{x_k\}_{k=0}^\infty$ 在期望意义下收敛到最小范数解 $x_* = A^\dagger b$, 并且满足

$$E \|x_k - x_*\|_2^2 \leq \left(1 - \frac{\sigma_{\min}^2(A)}{\|A\|_F^2}\right) \|x_0 - x_*\|_2^2,$$

$k = 0, 1, 2, \dots$, 其中 $\sigma_{\min}(A)$ 和 $\|A\|_F$ 分别为矩阵 A 的最小非零奇异值和 Frobenius 范数。

• 算法 [3]

- 1、输入 A, b, x_0 .
- 2、输出 $Ax=b$ 的近似解 x .
- 3、for $k = 1, \dots$, 直到满足停机准则 do
- 4、按照概率 $\frac{\|a_{i_k}\|_2^2}{\|A\|_F^2}$ 选取行指标 $i_k \in 1, 2, \dots, m$
- 5、计算

$$x_{k+1} = x_k + \frac{b_{i_k} - a_{i_k}^T x_k}{\|a_{i_k}\|_2^2} a_{i_k}$$

6、end for.

此外, Needell[5] 等的研究表明, 在有噪声的情况, 也即是在方程组右端加上误差向量 γ 后的系统: $Ax \approx b + \gamma$, 随机 Kaczmarz 方法也有良好的表现. 对于不相容线性系统, 随机 Kaczmarz 方法求解效率低下, 甚至不能收敛到系统的最小范数最小二乘解.

1.3 贪婪 Kaczmarz 方法

贪婪 Kaczmarz 方法 [McCormick, 1977] 在每步迭代选择离当前迭代点距离最远的超平面.

• 算法:

矩阵 $A \in \mathbb{R}^{m \times n}$, 右端项 $b \in \mathbb{R}^m$, 初始向量 $x_0 \in \mathbb{R}^n$.

求 $Ax = b$ 的近似解 x 步骤如下:

1、 $k = 0, 1, 2, \dots$, 直到满足停机准则

2、选取 $i_k = \arg \max_{i \in \{1, \dots, m\}} \frac{|b_i - a_i^T x_k|}{\|a_i\|_2}$

3、计算

$$x_{k+1} = x_k + \frac{b_{i_k} - a_{i_k}^T x_k}{\|a_{i_k}\|_2^2} a_{i_k}$$

4、 $k = k+1$

5、返回近似解

定理 1.2 (收敛性) 若线性方程组 $Ax = b$ 相容, 初始近似值 $x_0 \in \mathcal{R}(A^T)$. 则通过贪婪 Kaczmarz 方法生成的迭代序列 $\{x_k\}_{k=0}^\infty$ 收敛到其最小范数解 $x_\star = A^\dagger b$, 且迭代序列 $\{x_k\}_{k=0}^\infty$ 的误差满足:

$$\|x_1 - x_\star\|_2^2 \leq \left(1 - \frac{\sigma_{\min}^2(A)}{\|A\|_F^2}\right) \|x_0 - x_\star\|_2^2,$$

对于 $k = 1, 2, \dots$,

$$\|x_{k+1} - x_\star\|_2^2 \leq \left(1 - \frac{\sigma_{\min}^2(A)}{\max_{1 \leq j \leq m} \sum_{i=1, i \neq j}^m \|a_i\|_2^2}\right) \|x_k - x_\star\|_2^2.$$

1.4 预处理

1.4.1 左右预处理

考虑线性方程组

$$Ax = b$$

如果 K 是一个非奇异矩阵, 变换后的线性系统

$$K^{-1}Ax = K^{-1}b$$

具有和原方程组相同的解, 矩阵 K 称为预处理子, 如果矩阵 A 是非 Hermitian 的, 则相应的预处理为

$$K_1^{-1}AK_2^{-1}(K_2x) = K_1^{-1}b$$

其中, $K = K_1 K_2$, 上述形式称为左右预处理

$K_1 = I$ 时, 有

$$AK^{-1}(Kx) = b$$

称为右预处理

$K_2 = I$ 时, 有

$$K^{-1}Ax = K^{-1}b$$

称为左预处理

1.4.2 CGLS 方法的预处理

CGNR 和 CGLS 方法基于正规方程

$$A^T Ax = A^T b.$$

求解预处理的正规方程

$$R^{-T} A^T A R^{-1} x = R^{-T} A^T b$$

QR 分解预处理:

$$A = QR, Q \in \mathbb{R}^{m \times n}, R \in \mathbb{R}^{n \times n}$$

计算复杂度: $\mathcal{O}(mn^2)$

1.5 随机嵌入

定理 1.3 (随机嵌入) 令 $E \subset \mathbb{R}^n$ 是一个集合, 令 $\varepsilon \in (0, 1)$ 是一个失真参数. 我们称线性映射 $S: \mathbb{R}^n \rightarrow \mathbb{R}^d$ 是 E 的一个 (ℓ_2) 嵌入, 如果

$$(1 - \varepsilon)\|x\|_2 \leq \|Sx\|_2 \leq (1 + \varepsilon)\|x\|_2, \forall x \in E$$

如果 E 是一个子空间 $\mathcal{V} \subseteq \mathbb{R}^m$, 我们称 S 是一个子空间嵌入

如果上述性质对于任何 k 维子空间 \mathcal{V} 都成立, 则 S 是一个 k 维 (无关) 子空间嵌入

Indyk & Motwani, 1999 年使用随机嵌入提出近似最近邻问题的新算法 [6]

Alon, Matias & Szegedy, 1999 年引入术语 sketching, 并展示了如何使用 sketching 跟踪流数据 [7]
基于随机嵌入的方法大致分为三类。

1.5.1 Sketch-and-solve

Sketch-and-solve 原始的超定问题映射到一个更小的空间中, 使用简化问题的解作为原始问题解的近似。方法快速, 只需要对矩阵 A 进行一次扫描, 但结果可能不准确, 尽管残差具有可比性, 但即使原问题的解是唯一的, 也不一定是 $\hat{x} \approx x_*$

- 算法

- 1、考虑压缩的最小二乘问题

$$\min_{x \in \mathbb{R}^n} \|\mathcal{S}(b - Ax)\|_2$$

- 2、使用 sketched 问题的解 \hat{x} 代替原始问题的解 x_* 。

理论边界:

$$\|A\hat{x} - b\|_2 \leq \|Ax_* - b\|_2, \quad d \sim n \log(n)/\varepsilon^2$$

1.5.2 迭代 sketching

迭代 sketching 通过重复使用 sketch-and-solve 过程来减少残差, 提高精度, 可以产生与传统最小二乘算法一样小的误差, 同时比经典方法更快, 但需要对矩阵 A 进行重复的 sketching 过程, 成本高于 sketch-and-solve

- 算法

- 1、对于每一次迭代 i , 抽样一个新的对于 $\mathcal{R}(A)$ 随机子空间嵌入 \mathcal{S}_i
- 2、求解一系列最小二乘问题

$$\min_{x \in \mathbb{R}^n} \frac{1}{2} \|\mathcal{S}_i A(x - x_{i-1})\|_2^2 + \langle x - x_{i-1}, A^T(b - Ax_{i-1}) \rangle$$

- 3、子问题的解 x_i 用于下一个子问题。

理论边界:

$$\|Ax_j - b\|_2 \leq \|Ax_* - b\|_2, \quad j \sim \log(1/\varepsilon), d \sim n \log(n)/\varepsilon^2$$

1.5.3 Sketch-and-precondition

Sketch-and-precondition 使用随机嵌入来实现输入矩阵的替代, 并使用这个替代来为经典迭代算法构造预处理, 使其在最少的迭代次数内收敛。该方法速度比 sketch-and-solve 和迭代 sketching 快, 但是需要对 A 进行重复应用。

- 算法

- 1、计算 $Y = SA$ 和 $Y = QR, Q \in \mathbb{R}^{m \times n}, R \in \mathbb{R}^{n \times n}$
- 2、求解预处理后的问题

$$\min_{x \in \mathbb{R}^n} \|b - (AR^\dagger)(Rx)\|_2$$

- 3、使用 sketch-and-solve 过程的近似解作为初始估计 $x_0 \in \mathbb{R}^n$

理论边界:

$$\|Ax_j - b\|_2 \leq \|Ax_* - b\|_2, \quad j \sim \log(1/\varepsilon)$$

1.5.4 几个经典的 Sketching 矩阵

随机嵌入矩阵 \mathcal{S} 应当易于构造、并且能提高数据计算效率。此外，根据随机嵌入的定义，矩阵 \mathcal{S} 还应保持集合 E 的几何形状近似不变，也就是满足

$$(1 - \varepsilon)\|x\|_2 \leq \|\mathcal{S}x\|_2 \leq (1 + \varepsilon)\|x\|_2, \forall x \in E$$

• Gaussian 嵌入

Boutsidis, Drineas 和 Magdon-Ismail[8] 在 2014 提出依据正态分布生成的 Gaussian 嵌入矩阵：

$$\mathcal{S} = \frac{1}{\sqrt{d}}R \in \mathbb{R}^{d \times m}, R_{ij} \sim N(0, 1)$$

储存一个 Gaussian 嵌入的成本是 $\mathcal{O}(dm)$ ，将其应用在一个向量上的成本也是 $\mathcal{O}(dm)$ 。

• 子采样随机三角变换 (SRTT)

$$\mathcal{S} = \sqrt{\frac{m}{d}}RFE\Pi \in \mathbb{R}^{d \times m}$$

其中, $R \in \mathbb{R}^{d \times m}$ 是随机约束, $F \in \mathbb{R}^{m \times m}$ 是酉三角变换, $E \in \mathbb{R}^{m \times m}$ 是随机符号翻转, $\Pi \in \mathbb{R}^{m \times m}$ 是随机置换在实数域的情况下, F 通常选择离散余弦变换, 或离散 Hartley 变换; 当 m 是 2 的幂时, 可以考虑 Walsh-Hadamard 变换。存储 SRTT 的成本为 $\mathcal{O}(m \log m)$, 使用快速下采样三角变换的算法可以在 $\mathcal{O}(m \log d)$ 次操作内将其应用于向量。

• 稀疏符号矩阵:

Achlioptas[9] 在 2013 年提出只含有 -1, 0, 1 元素的稀疏符号矩阵, 并证明矩阵并没有降低嵌入的质量。该矩阵为:

$$\mathcal{S} = \sqrt{\frac{m}{\zeta}}[s_1, \dots, s_m] \in \mathbb{R}^{d \times m}$$

其中列 $s_i \in \mathbb{R}^d$ 是 i.i.d. 的随机向量, 对于每一列抽样 ζ 个 i.i.d. 的随机符号, 并将它们置于 ζ 个均匀随机坐标中, 稀疏参数 $2 \leq \zeta \leq d$ 。该方法的储存成本为 $\mathcal{O}(\zeta m \log d)$, 在 $\mathcal{O}(\zeta m)$ 次操作中将其应用于向量。特别的, 当 $\zeta = 1$ 时, 称为 Count sketch 变换 [10]。

定理 1.4 [11][12]

若 $\mathcal{S} \in \mathbb{R}^{d \times m}$ 是一个 count sketch 变换, $d = (n^2 + n) / (\delta \varepsilon^2)$, 其中 $0 < \delta, \varepsilon < 1$, 则在 $1 - \delta$ 的概率下, 有

$$(1 - \varepsilon)\|Ax\|_2 \leq \|\mathcal{S}Ax\|_2 \leq (1 + \varepsilon)\|Ax\|_2 \text{ 对所有 } x \in \mathbb{R}^n,$$

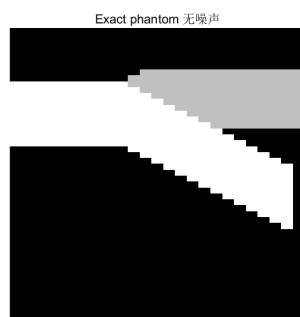
$$(1 - \varepsilon)\sigma_i(A) \leq \sigma_i(\mathcal{S}A) \leq (1 + \varepsilon)\sigma_i(A) \text{ 对所有 } 1 \leq i \leq n$$

此外, 我们还可以抽样使用矩阵的部分信息, 例如均匀抽样 [13]、杠杆植抽样等, 这些方法和矩阵相干性有着密切关系。

2 数值实践

通过数值实验比较在有无噪声的前提下，共轭梯度法（CG）、随机 Kaczmarz 方法（RK）、贪婪随机 Kaczmarz 方法（GK）及三者经过预处理之后的计算效率与图片结果。在本节的所有数值实验中，设置最高迭代步数为 $K=100$ ；噪声分为三个层次：无噪声， $\text{eta}=0.005$ ， $\text{eta}=0.01$ ；这里迭代步数（IT）、计算时间（CPU, 单位：秒）和信噪比（SNR）取 5 次计算结果的中位数。

图 2: 地震波标准成像图



2.1 纯 CGLS 算法、随机 Kaczmarz、贪婪 Kaczmarz 恢复地震波图像

用作未随机嵌入时各方法回复地震波图像的案例。本案例中通过 CGLS 算法、随机 Kaczmarz、贪婪 Kaczmarz 三种主要方法对地震波图像的恢复，比较不同图像恢复的情况，同时测试相应算法的相关性能。

• 代码逻辑：

- 1、通过引用 `seismictomo` 函数 [14]，随机生成地震波，并通过设置一系列参数，如离散点、相对噪声等级、调节有无噪声代码、最大迭代次数、精度等参数，实现对恢复图像的比较与改进。
- 2、纯 CGLS 方法：以 CGLS 方法为基础，在每次迭代中选择一个共轭方向得到相关向量 α ，由此更新解向量。从迭代的方式上，即在循环迭代中，每次计算相对误差，并将其与我们所设计的精度水平进行比较，如果达到满足精度水平的迭代情况，则将相应的迭代值输出，通过 `subplot` 函数实现对其重构结果的图像表示。
- 3、纯 RK 方法：以随机 Kaczmarz 方法为基础，在每一次迭代中，从方程组中随机选择一个方程，然后使用该方程来更新解向量，使其更接近最优解，在算法中即通过最小化方程的残差平方和来更新解向量。从迭代的方式上，即在循环迭代中，每次计算相对误差，并将其与我们所设计的精度水平进行比较，如果达到满足精度水平的迭代情况，则将相应的迭代值输出，通过 `subplot` 函数实现对其重构结果的图像表示。
- 4、纯 GK 方法：与随机 Kaczmarz 方法类似，其以贪婪 Kaczmarz 为基础，在每一次迭代中，都选择与其距离最大的法平面作为投射平面，从而获得相关射影，在算法中即计算具有最

大残差的方程，从而更新解向量。从迭代的方式上，即在循环迭代中，每次计算相对误差，并将其与我们所设计的精度水平进行比较，如果达到满足精度水平的迭代情况，则将相应的迭代值输出，通过 subplot 函数实现对其重构结果的图像表示。

5、以上的算法若超出最大迭代次数仍未满足停机要求，则退出迭代，并在此基础上直接输出不满足精度要求的图像。

表 1: 不同噪声下三种方法的表现

算法	嵌入	无噪声			噪声 $\eta=0.005$			噪声 $\eta=0.01$		
		IT	CPU	SNR	IT	CPU	SNR	IT	CPU	SNR
标准成像	-	-	-	-	-	-	-	-	-	-
CG	-	100	0.0781	28.9191	100	0.0938	18.1896	100	0.0938	11.9732
RK	-	100	0.0469	3.0854	100	0.0938	2.5308	100	0.0469	2.5077
GK	-	100	0.0625	10.3256	100	0.0313	10.1061	100	0.0313	10.4386

图 3: 三种迭代无噪声

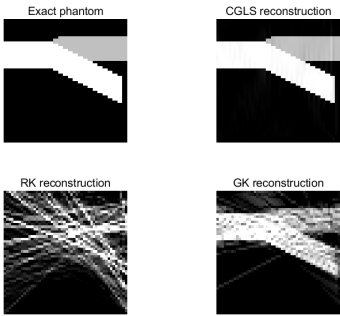


图 4: 三种迭代噪声 $\eta=0.005$

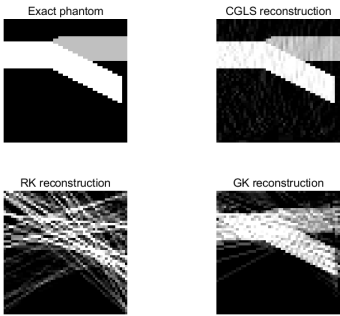


图 5: 三种迭代噪声 $\eta=0.01$

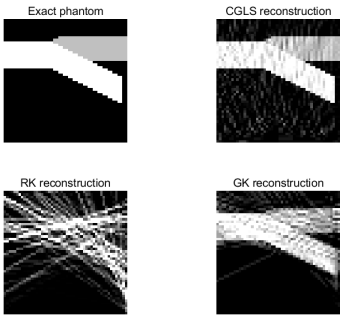
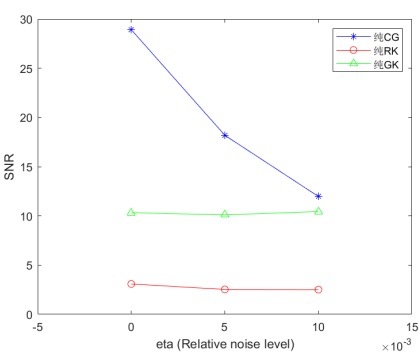


图 6: 三种迭代信噪比随噪声等级变化



从图片的复原效果看，表 1 数据与图 3-5 显示，未做任何处理的 CGLS 方法在迭代相同次数的情况下，对于图片的复原效果要优于两类 Kaczmarz 方法，但从图 6 折线图可以看出，CGLS 方法对于噪声的敏感性很高，易受噪声影响，随着噪声等级的提升图片复原效果急速下降，同时，CGLS 方法相较两类 Kaczmarz 方法，单次迭代需要花更多的时间，所以 CGLS 方法更适合在噪声较小时使用。噪声对两类 Kaczmarz 方法的影响较小，其中贪婪 Kaczmarz 方法在以上数

据实践中要全面优于随机 Kaczmarz 方法，相同迭代次数下，贪婪 Kaczmarz 方法用时更少，复原情况更好，在噪声较高时已然接近 CGLS 方法，所以在噪声较高时，贪婪 Kaczmarz 方法更为契合。

2.2 CGLS 共轭梯度法恢复地震波图像

通过 CGLS 算法本身以及 CGLS 算法结合预处理嵌入优化（CGLS 带 Gaussian 嵌入、CountSketch 变换、SRTT 子采样随机三角变换）共计四种方法，恢复地震波图像，并与原始图像进行比较，同时测试相关性能。

• 代码逻辑：

- 1、通过引用 seismictomo 函数，随机生成地震波，并通过设置一系列参数，如离散点、相对噪声等级、调节有无噪声代码、最大迭代次数、精度等参数，实现对恢复图像的比较与改进。
- 2、纯 CGLS 方法：以 CGLS 方法为基础，在每次迭代中选择一个共轭方向得到相关向量 α ，由此更新解向量。从迭代的方式上，即：在循环迭代中，每次计算残差的相对范数，并将其与我们所设计的精度水平进行比较，如果达到满足精度水平的迭代情况，则将相应的迭代值输出，通过 subplot 函数实现对其重构结果的图像表示。
- 3、纯 CGLS 带 Gaussian 方法：以 CGLS 方法为基础，在以上方法介绍所述的基础上，在循环前选定嵌入矩阵为高斯矩阵 S ，通过 $SA=QR$ 进行 QR 分解，进行相关处理后作为预处理结果，进入循环迭代出 x ，顺便算出所有的性能相关参数。
- 4、带 CountSketch 变换及带 SRTT 变换，原理同 3）。
- 5、以上的算法若超出最大迭代次数仍未满足停机要求，则退出迭代，并在此基础上直接输出不满足精度要求的图像。

其中，CountSketch 变换由于预处理期间的大规模稀疏矩阵问题，编码时无法直接求伪逆，而是需要通过生成相应规模单位矩阵求逆的方式进行计算。由于奇异值分布与特征值相近，实际代码中经测试无法直接左除，相应的步骤在非稀疏矩阵前提下，可进行伪逆运算。

表 2: CGLS 共轭梯度法恢复地震波图像

算法	嵌入	无噪声			噪声 $\eta=0.005$			噪声 $\eta=0.01$		
		IT	CPU	SNR	IT	CPU	SNR	IT	CPU	SNR
CG	-	38	0.0469	24.1739	100	0.2344	18.1896	100	0.1875	11.9732
	Gaussian	7	46.1563	-33.2213	100	476.2031	-43.8059	100	484.0781	-226.9575
	Count Sketch	18	68.5469	-34.8077	100	198.4844	-43.8069	100	193.6094	-49.8275
	SRTT	9	55.4063	-33.9253	100	465.3906	-43.8069	100	495.6094	-49.8176

图 7: 共轭梯度法 (CG) 无噪声

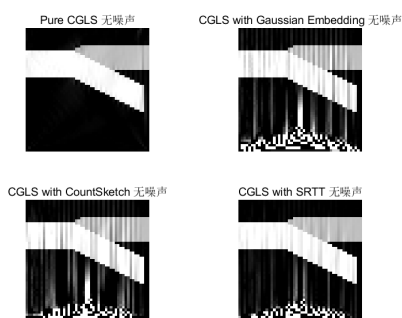
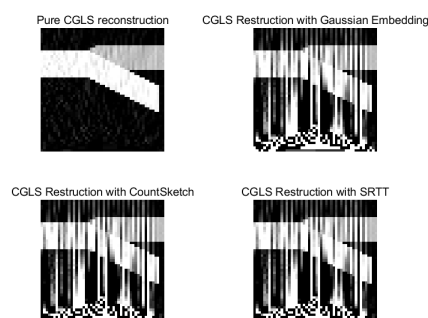
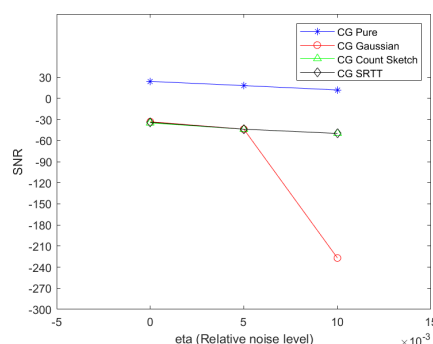
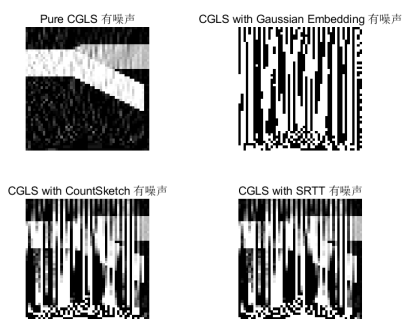
图 8: 共轭梯度法 (CG) 噪声 $\eta=0.005$ 图 9: 共轭梯度法 (CG) 噪声 $\eta=0.01$ 图 10: (CG) 信噪比随噪声等级变化

表 2 数据显示, 通过预处理能有效地降低 CGLS 方法的迭代次数, 却会大幅增加处理时间, 而且在图 7-9 直观显示的图片复原效果上, 仍旧是未做预处理的 CGLS 方法最优。其次, 预处理并没有改变 CGLS 方法容易受噪声影响的情况, 相反, 图 10 显示, 使用 Gaussian 嵌入的 CGLS 方法对于噪声等级的提升反应更为剧烈, 信噪比陡降。在三种嵌入中, 使用 Count sketch 变换的单次迭代时间较少, 且与使用 SRTT 方法有相近的信噪比, 算是三类中的最优。

2.3 贪婪 Kaczmarz 方法恢复地震波图像

在算法的编程时, 以未随机嵌入时贪婪 Kaczmarz 方法恢复地震波图像的案例为基础, 然后在此基础上, 分别加入不同的预处理嵌入方式, 即通过 Gaussian 嵌入、CountSketch 变换以及 SRTT 变换这三种方式与贪婪 Kaczmarz 相结合, 最终比较这三种与未有任何预处理嵌入的纯贪婪 Kaczmarz 方法, 以及地震波图像原图的比较, 得出图像上的恢复结论, 同时测试相关算法性能。

• 代码逻辑:

- 1、通过引用 seismictomo 函数, 随机生成地震波, 并通过设置一系列参数, 如离散点、相对噪声等级、调节有无噪声代码、最大迭代次数、精度等参数, 实现对恢复图像的比较与改进。

2、纯贪婪 Kaczmarz 方法：以贪婪 Kaczmarz 方法为基础，在每一次迭代中，都选择与其距离最大的法平面作为投射平面，从而获得相关射影，在算法中即计算具有最大残差的方程，从而更新解向量。从迭代的格式上，即在循环迭代中，每次计算相对误差，并将其与我们所设计的精度水平进行比较，如果达到满足精度水平的迭代情况，则将相应的迭代值输出，通过 subplot 函数实现对其重构结果的图像表示。

3、贪婪 Kaczmarz 方法带 Gaussian 嵌入：以上述 GK 方法为基础，在循环前选定嵌入矩阵为高斯矩阵 S ，通过 $SA=QR$ 进行 QR 分解，进行相关右处理后生成变量 y ，将其代入循环迭代中， $Ax=b$ 在循环中相应转换为 $Aa*y=b$ ，其中 Aa 即上述提及的右处理后的系数矩阵。从迭代的格式上，即在循环迭代中，每次计算相对误差，并将其与我们所设计的精度水平进行比较，如果达到满足精度水平的迭代情况，则将相应的迭代值输出，最后将确定的 y 值通过相应右处理正交阵 R 的逆运算得到 x ，再通过 subplot 函数实现对其重构结果的图像表示。

4、贪婪 Kaczmarz 方法带 CountSketch 变换及 SRTT 变换，原理同 3)。

5、以上的算法若超出最大迭代次数仍未满足停机要求，则退出迭代，并在此基础上直接输出不满足精度要求的图像。

其中，CountSketch 变换由于预处理期间的大规模稀疏矩阵问题，编码时无法直接求伪逆，而是需要通过生成相应规模单位矩阵求逆的方式进行计算。由于奇异值分布与特征值相近，实际代码中经测试无法直接左除，相应的步骤在非稀疏矩阵前提下，可进行伪逆运算。

表 3: 贪婪 Kaczmarz 方法恢复地震波图像

算法	嵌入	无噪声			噪声 $\eta=0.005$			噪声 $\eta=0.01$		
		IT	CPU	SNR	IT	CPU	SNR	IT	CPU	SNR
GK	-	100	0.0938	10.3256	100	0.0938	10.1061	100	0.0781	10.4386
	Gaussian	100	24.9375	-2.3935	100	25.5781	-2.2563	100	26.6094	-2.6178
	Count Sketch	100	58.2500	NaN	100	59.9219	NaN	100	59.2656	NaN
	SRTT	100	24.4844	-2.4388	100	33.3438	-1.9756	100	30.9531	-2.4091

图 11: 贪婪 Kaczmarz 无噪声

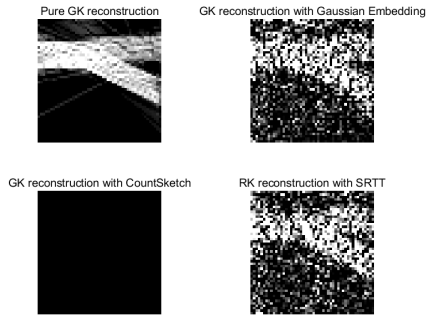
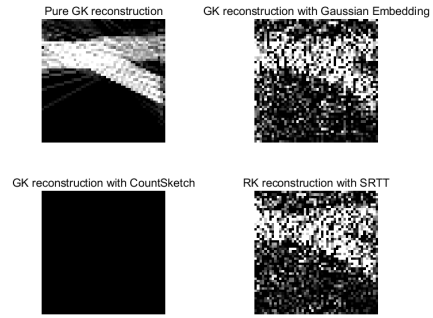
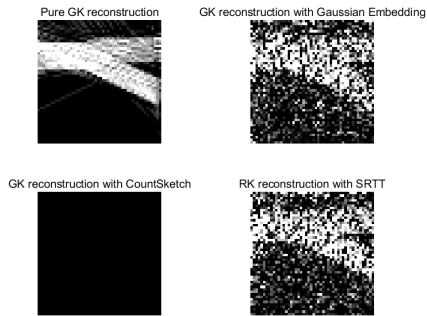
图 12: 贪婪 Kaczmarz 噪声 $\eta=0.005$ 图 13: 贪婪 Kaczmarz 噪声 $\eta=0.01$ 

图 14: (GK) 信噪比随噪声等级变化

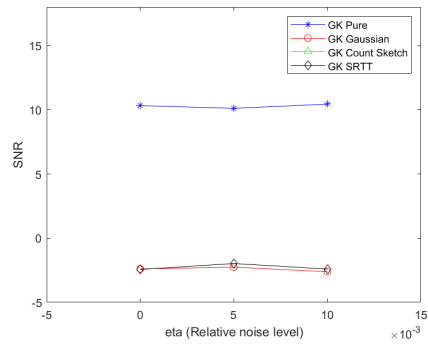


表 3 数据显示, 相同迭代次数下, 通过预处理的贪婪 Kaczmarz 方法会大幅增加处理时间, 而且在图 11-13 直观显示的图片复原效果上, 仍旧是未做预处理的贪婪 Kaczmarz 方法最优。其中, 带 Count sketch 变换的贪婪 Kaczmarz 方法复原图片失败。图 14 显示, 贪婪 Kaczmarz 方法对于噪声的敏感性不高, 不易受噪声影响, 带 Gaussian 嵌入及 SRTT 变换的两种贪婪 Kaczmarz 方法在相同迭代次数下, 复原效果相近, 带 Gaussian 嵌入的在用时上较少, 但优势不明显。

2.4 随机 Kaczmarz 方法恢复地震波图像

类似于随机 Kaczmarz 方法, 但在实际算法的编写中, 在此基础之上, 分别加入不同的预处理嵌入方式, 即通过 Gaussian 嵌入、CountSketch 变换以及 SRTT 变换这三种方式与随机 Kaczmarz 相结合, 最终比较这三种与未有任何预处理嵌入的纯随机 Kaczmarz 方法, 以及图像原图的比较, 得出图像上的恢复结论, 同时测试相关算法性能。

• 代码逻辑:

- 1、通过引用 seismictomo 函数, 随机生成地震波, 并通过设置一系列参数, 如离散点、相对噪声等级、调节有无噪声代码、最大迭代次数、精度等参数, 实现对恢复图像的比较与改进。
- 2、纯随机 Kaczmarz 方法: 以随机 Kaczmarz 方法为基础, 在每一次迭代中, 从方程组中随

机选择一个方程，然后使用该方程来更新解向量，使其更接近最优解，在算法中即通过最小化方程的残差平方和来更新解向量。从迭代的方式上，即在循环迭代中，每次计算相对误差，并将其与我们所设计的精度水平进行比较，如果达到满足精度水平的迭代情况，则将相应的迭代值输出，通过 subplot 函数实现对其重构结果的图像表示。

3、随机 Kaczmarz 方法带 Gaussian 嵌入：以上述 RK 方法为基础，在循环前选定嵌入矩阵为高斯矩阵 S ，通过 $SA=QR$ 进行 QR 分解，进行相关右处理后生成变量 y ，将其代入循环迭代中， $Ax=b$ 在循环中相应转换为 $Aa*y=b$ ，其中 Aa 即上述提及的右处理后的系数矩阵。从迭代的方式上，即在循环迭代中，每次计算相对误差，并将其与我们所设计的精度水平进行比较，如果达到满足精度水平的迭代情况，则将相应的迭代值输出，最后将确定的 y 值通过相应右处理正交阵 R 的逆运算得到 x ，再通过 subplot 函数实现对其重构结果的图像表示。

4、随机 Kaczmarz 方法带 CountSketch 变换及 SRTT 变换，原理同 3)。

5、以上的算法若超出最大迭代次数仍未满足停机要求，则退出迭代，并在此基础上直接输出不满足精度要求的图像。

其中，CountSketch 变换由于预处理期间的大规模稀疏矩阵问题，编码时无法直接求伪逆，而是需要通过生成相应规模单位矩阵求逆的方式进行计算。由于奇异值分布与特征值相近，实际代码中经测试无法直接左除，相应的步骤在非稀疏矩阵前提下，可进行伪逆运算。

表 4: 随机 Kaczmarz 方法恢复地震波图像

算法	嵌入	无噪声			噪声 $\eta=0.005$			噪声 $\eta=0.01$		
		IT	CPU	SNR	IT	CPU	SNR	IT	CPU	SNR
RK	-	100	0.1094	2.5431	100	0.0781	2.4146	100	0.0938	2.4963
	Gaussian	100	24.2656	-4.1720	100	25.5781	-4.4309	100	24.6250	-4.2960
	Count Sketch	100	58.2500	-5.5675	100	57.5469	-5.6506	100	58.7031	-5.0274
	SRTT	100	29.8594	-4.1120	100	27.5625	-4.2525	100	29.4844	-4.1075

图 15: 随机 Kaczmarz 无噪声

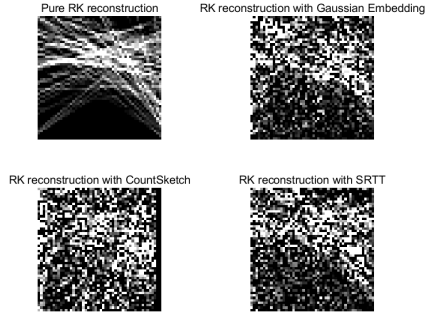


图 16: 随机 Kaczmarz 噪声 $\eta=0.005$

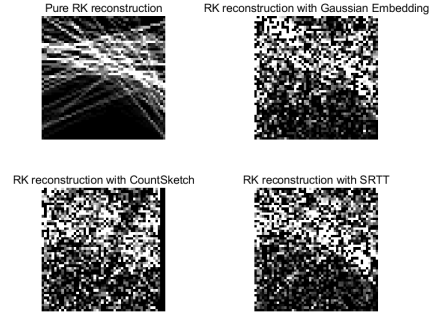


图 17: 随机 Kaczmarz 噪声 $\eta=0.01$

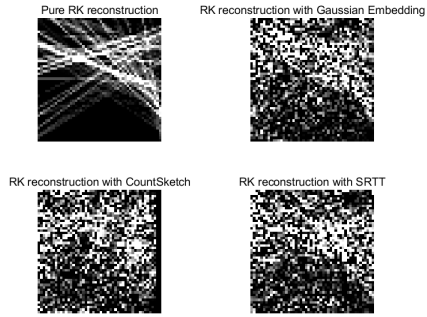


图 18: (RK) 信噪比随噪声等级变化

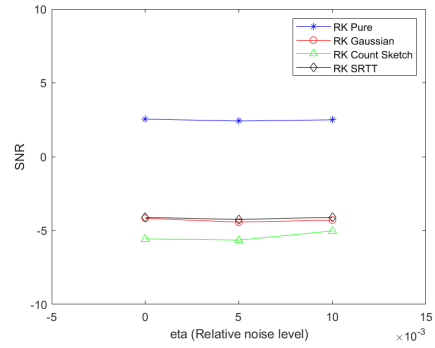


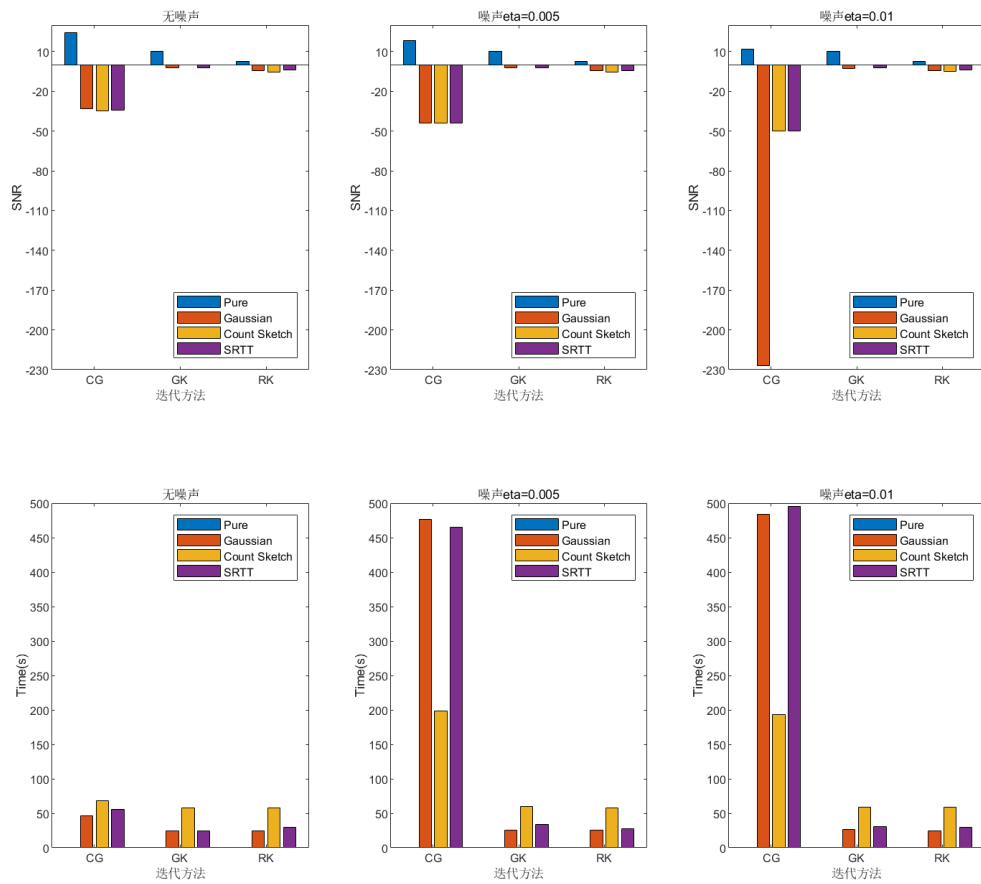
表 4 数据显示, 相同迭代次数下, 通过预处理的随机 Kaczmarz 方法会大幅增加处理时间, 而且在图 15-17 直观显示的图片复原效果上, 仍旧是未做预处理的随机 Kaczmarz 方法最优。图 18 显示, 随机 Kaczmarz 方法对于噪声的敏感性不高, 不易受噪声影响, 带 Gaussian 嵌入及 SRTT 变换的两种随机 Kaczmarz 方法在相同迭代次数下, 用时与复原效果相近, 而带 Count sketch 变换的随机 Kaczmarz 方法则要逊色于其他两种。

2.5 数据汇总

表 5: 三种方法复原地震波图像汇总

算法	嵌入	无噪声			噪声 $\eta=0.005$			噪声 $\eta=0.01$		
		IT	CPU	SNR	IT	CPU	SNR	IT	CPU	SNR
CG	-	38	0.0469	24.1739	100	0.2344	18.1896	100	0.1875	11.9732
	Gaussian	7	46.1563	-33.2213	100	476.2031	-43.8059	100	484.0781	-226.9575
	Count Sketch	18	68.5469	-34.8077	100	198.4844	-43.8069	100	193.6094	-49.8275
	SRTT	9	55.4063	-33.9253	100	465.3906	-43.8069	100	495.6094	-49.8176
GK	-	100	0.0938	10.3256	100	0.0938	10.1061	100	0.0781	10.4386
	Gaussian	100	24.9375	-2.3935	100	25.5781	-2.2563	100	26.6094	-2.6178
	Count Sketch	100	58.2500	NaN	100	59.9219	NaN	100	59.2656	NaN
	SRTT	100	24.4844	-2.4388	100	33.3438	-1.9756	100	30.9531	-2.4091
RK	-	100	0.1094	2.5431	100	0.0781	2.4146	100	0.0938	2.4963
	Gaussian	100	24.2656	-4.1720	100	25.5781	-4.4309	100	24.6250	-4.2960
	Count Sketch	100	58.2500	-5.5675	100	57.5469	-5.6506	100	58.7031	-5.0274
	SRTT	100	29.8594	-4.1120	100	27.5625	-4.2525	100	29.4844	-4.1075

图 19: 各方法 SNR 与 CPU 时间图



3 结论

基于 sketching 的预处理能很好的降低迭代次数,但相同迭代次数情况下需要花更长的时间,且复原效果较未处理的差。在限定迭代次数的情况下,低噪声时,无任何预处理的 CGLS 方法优势明显,用时少,复原效果好。在噪声较高时则选择无任何预处理的贪婪 Kaczmarz 方法,其受噪声影响低,且复原效果优于其他方法。基于 sketching 的预处理未能在限定迭代次数较少时发挥效果。

参考文献

- [1] 李晓玲, 陈磊, 杨春雨, 丁虎, 马佳伟. 地震波成像专利技术综述[J]. China Invention Patent, 2016: 61-64.
- [2] 刘畅, 李振春, 曲英铭, 等. 地震层析成像方法综述[J]. 物探与化探, 2020, 44(2): 227-234.
- [3] STROHMER T, VERSHYNIN R. A randomized kaczmarz algorithm with exponential convergence[J]. Journal of Fourier Analysis and Applications, 2009, 15(2): 262-278.
- [4] BAI Z Z, WU W T. On greedy randomized kaczmarz method for solving large sparse linear systems[J]. SIAM Journal on Scientific Computing, 2018, 40(1): A592-A606.
- [5] NEEDELL D. Randomized kaczmarz solver for noisy linear systems[J]. BIT Numerical Mathematics, 2010, 50: 395-403.
- [6] INDYK P, MOTWANI R. Approximate nearest neighbors: towards removing the curse of dimensionality[C]//Proceedings of the thirtieth annual ACM symposium on Theory of computing. 1998: 604-613.
- [7] ALON N, GIBBONS P B, MATIAS Y, et al. Tracking join and self-join sizes in limited storage[C]//Proceedings of the eighteenth ACM SIGMOD-SIGACT-SIGART symposium on Principles of database systems. 1999: 10-20.
- [8] BOUTSIDIS C, DRINEAS P, MAGDON-ISMAIL M. Near-optimal column-based matrix reconstruction[J]. SIAM Journal on Computing, 2014, 43(2): 687-717.
- [9] ACHLIOPTAS D. Database-friendly random projections: Johnson-lindenstrauss with binary coins[J]. Journal of Computer and System Sciences, 2003, 66(4): 671-687.
- [10] CHARIKAR M, CHEN K, FARACH-COLTON M. Finding frequent items in data streams[C]//Automata, Languages and Programming. Springer Berlin Heidelberg, 2002: 693-703.
- [11] MENG X, MAHONEY M W. Low-distortion subspace embeddings in input-sparsity time and applications to robust linear regression[C]//Proceedings of the Forty-Fifth Annual ACM Symposium on Theory of Computing. Association for Computing Machinery, 2013: 91-100.
- [12] WOODRUFF D P. Sketching as a tool for numerical linear algebra[J]. Foundations and Trends® in Theoretical Computer Science, 2014, 10(1-2): 1-157.
- [13] KATRUTSA A, OSELEDITS I V. Preconditioning kaczmarz method by sketching[J]. CoRR, 2019, abs/1903.01806.
- [14] HANSEN P C, JØRGENSEN J S. Air tools ii: algebraic iterative reconstruction methods, improved implementation[J]. Numerical Algorithms, 2018, 79(1): 107-137.