The standard forms of the Kaczmarz-Tanabe type methods and their convergence theory

Chuan-gang Kang

School of Mathematical Sciences, Tiangong University, Tianjin, 300387, People’s Republic of China

Abstract

In this paper, we consider the standard form of two kinds of Kaczmarz-Tanabe type methods, one derived from the Kaczmarz method and the other derived from the symmetric Kaczmarz method. As a famous image reconstruction method in computed tomography, the Kaczmarz method has both advantage and disadvantage. The advantage are simple and easy to implement, while the disadvantages are slow convergence speed, and the symmetric Kaczmarz method is the same. For the standard form of this method, once the iterative matrix is generated, it can be used continuously in the subsequent iterations. Moreover, the iterative matrix can be stored in the image reconstructive devices, which makes the Kaczmarz method and the symmetric Kaczmarz method can be used like the simultaneous iterative reconstructive techniques (SIRT). Meanwhile, theoretical analysis shows that the convergence rate of symmetric Kaczmarz method is better than the Kaczmarz method but is slightly worse than that of two iterations Kaczmarz method, which is verified numerically. Numerical experiments also show that the convergence rates of the Kaczmarz method and the symmetric Kaczmarz method are better than the SIRT methods and slightly worse than CGMN method in some cases. However, the Kaczmarz Tanabe type methods have better problem adaptability.

Key words. Kaczmarz method; Symmetric Kaczmarz method; SIRT method; Convergence rate; image reconstruction; Computed tomography

AMS subject classification. 65F10; 65F08; 65N22; 65J20

1 Introduction

In medical imaging tomography (see, i.e., [1–3]), people are often asked to solve the following linear system with equations, i.e.,

\[ Ax = b, \]

(1.1)

where \( A \in \mathbb{R}^{m \times n}, b \in \mathbb{R}^{m} \) are also called projection matrix and measure vector, respectively. (1.1) can be generated by discretizing the radon transform

\[ p = \int_{L} f(x,y)ds, \]

here, \( L(\rho, \theta) = \{(x,y) : x \cos \theta + y \sin \theta = \rho \} \) (see, e.g., [4–6]). We suppose that (1.1) is consistent and \( x^* \) is any true solution. If \( A \) is not full column rank, \( x^\dagger \) is used to denote the minimum norm least-squares solution [7,8] of (1.1).
The Kaczmars method is one of the most popular iterative methods to solve (1.1) in computerized tomography, it was proposed by Polish mathematician Kaczmarsz [9]. Let $A = (a_1, a_2, \ldots, a_m)^T$, then Kaczmarsz’s iteration reads

$$x_k = x_{k-1} + \frac{b_i - (a_i, x_{k-1})}{\|a_i\|^2} a_i, \quad k = 1, 2, \ldots$$

where $i = \text{mod}(k, 2m) + 1$, $(x, y) = x^T y$ and $\|x\|_2 = \sqrt{(x, x)}$ denote the inner product of $x, y$ and the square norm of $x$ in $\mathbb{R}^n$, respectively.

The symmetric Kaczmars method can be described as

$$x_k = x_{k-1} + \frac{b_i - (a_i, x_{k-1})}{\|a_i\|^2} a_i, \quad k = 1, 2, \ldots$$

where

$$i = \begin{cases} \text{mod}(k, 2m - 2), & 1 \leq \text{mod}(k, 2m - 2) \leq m, \\ 2m - \text{mod}(k, 2m - 2), & m < \text{mod}(k, 2m - 2) \leq 2m - 3, \\ 2, & \text{mod}(k, 2m - 2) = 0. \end{cases}$$

Compare with the popular expression of the symmetric Kaczmars method (see, i.e., [10,11]), the iterative scheme (1.3) has good consistence in form with Kaczmarsz’s iteration.

The Kaczmars method has many advantages, such as good convergence, easy to implement and so on, and has been used to solve phase problem [12]. However, the convergence of Kaczmars method sometimes becomes very slow, especially the successive hyperplanes meet at a very small angle. In order to keep the advantages of Kaczmars method and overcome its disadvantages, many scholars consider the subsequence of sequence $\{x_k\}$, where $y_k = x_k$. Kang [13] give the following iterative scheme of Kaczmarsz’s subsequence $\{y_k\}$, i.e.,

$$y_{k+1} = (I - A_S^T MA)y_k + A_S^T Mb, \quad k = 0, 1, 2, \ldots$$

where, $I$ denotes identity matrix of whatever size appropriate to the context, and

$$P_i = I - \frac{a_i a_i^T}{\|a_i\|^2}, \quad i = 1, 2, \ldots, m$$

$$Q_m = I, Q_j = P_m P_{m-1} \cdots P_{j+1}, \quad j = 1, 2, \ldots, m - 1.$$  

$$Q = P_m P_{m-1} \cdots P_1;$$

$$A_S = (Q_1 a_1, Q_2 a_2, \ldots, Q_m a_m)^T;$$

$$M = \text{diag}(1/\|a_1\|_2^2, 1/\|a_2\|_2^2, \ldots, 1/\|a_m\|_2^2).$$

and the subsequential iteration (1.5) was named Kaczmarsz-Tanabe’iteration by Popa [14].

Compared with Kaczmarsz’s iteration, Kaczmarsz-Tanabe’s iteration has good approximate stability (i.e., iterative error is not fluctuate as violently as Kaczmarsz’s iteration (see [13]), which may provide convenience for people to study the regularization theory of Kaczmars method). In fact, (1.5) has some improvement compared with the traditional iterative scheme of Kaczmarsz-Tanabe method, see [14,15], but $A_S$ is the compound of $Q$, and $a_i$, which brings many obstacles for the further research, especially the regularization theory, etc. In this paper, we mainly consider the standard form of (1.5), and the corresponding iterative matrix can be calculated by blocking and parallelization, and the algorithm can be used to generate the symmetric linear system of equations for CGMN method (see [10]).

Assume that $r(Q) = p$, and $\sigma_1, \sigma_2, \ldots, \sigma_p$ are $p$ non-zero singular values of $Q$. Let

$$e_k := y_k - x^* - P_{N(A)} y_0.$$  

Kang gave the following convergence result (refer to [13, Theorem 2.10 & Corollary 2.11]).
Theorem 1.1. [13] For any matrix $A$ with non-zero rows, let $\{y_k, k \geq 0\}$ be the sequence of vectors generated by (1.5), then there holds

$$\|e_{k+1}\|_2 \leq \max_{0<\sigma_i<1} \sigma_i^{k+1} \|e_0\|_2,$$

where $\sigma_i$ is the singular value of $Q$.

We will further consider the Kaczmarz-Tanabe method and hope to get an matrix-vector iteration form similar to the SIRT methods. For ease of reference, we list several typical representations of SIRT methods (see, i.e., [16–24]) and the general iteration reads

$$x_{k+1} = x_k + \lambda_k T A^T M (b - Ax_k),$$

(1.11)

where $\lambda_k$ is the relaxation parameter. When $\lambda_k \equiv 1$, the following methods will be obtained by taking given $T$ and $M$ pairs.

- Landweber: $T = I, M = I$;
- Cimmino: $T = I, M = D = \frac{1}{m} \text{diag}(\frac{1}{\|a_i\|_2})$;
- CAV: $T = I, M = D_S = \text{diag}(\frac{1}{\|a_i\|_S^2}), S = \text{diag(\text{nnz(column j})});$
- DROP: $T = S^{-1}, M = mD;$
- SART: $T = \text{diag(row sums)}^{-1}, M = \text{diag(column sums)}^{-1}.$

The rest of the work is organized as follows. In Section 2, we consider the standard form (i.e., matrix-vector form) of the Kaczmarz-Tanabe method, and introduce some concepts related with the sequential projection. In Section 3, we consider the matrix-vector form of symmetric Kaczmarz method and give the explicit expression of the symmetric Kaczmarz-Tanabe method. Moreover, we also analyze the convergence rate of the symmetric Kaczmarz-Tanabe method and compare it with the Kaczmarz-Tanabe method. In section 4, we give the algorithm flows to calculate $C$ and $\bar{C}$ explicitly. In Section 5, we compare the computational efficiency of the Kaczmarz-Tanabe method, symmetric Kaczmarz-Tanabe method and SIRT methods by numerical experiments.

2 The standard form of Kaczmarz-Tanabe method and its convergence

Compared with (1.2), Kaczmarz-Tanabe iteration (1.5) has made great change in form because it gets rid of the constraint of projection row by row according to the system of equations. Observed from the construction of $A_S$ in (1.5), there still exist many inconveniences to use because each column vector is the product of a matrix $Q_i$ and vector $a_i$.

In this section, we will analyze the inherent structure of Kaczmarz-Tanabe’s iteration (1.5) and derive to a concise iterative form similar to the SIRT methods. First, we give the following definitions.

Definition 2.1. We call $Q_i$ a sequential projection matrix on $a_{j+1}, \ldots, a_m$, and denote the sequential projection matrix set with $S_{sp}(a_1, \ldots, a_m)$, i.e.,

$$S_{sp}(a_1, a_2, \ldots, a_m) = \{Q_1, Q_2, \ldots, Q_{m-1}\}.\quad (2.1)$$

Definition 2.2. For any $Q_i \in S_{sp}$, if there exist $\zeta_{i, 1}, \ldots, \zeta_{i, m}$ such that

$$Q_i a_i = \zeta_{i, 1} a_1 + \ldots + \zeta_{i, m} a_m, \quad (2.2)$$

3
then we call $A$ and $S_{sp}$ sequentially compatible. In general, for any $1 \leq i \leq m, i \leq j \leq m$, if there exist $\zeta^{(i,j)}_1, \ldots, \zeta^{(i,j)}_m$ such that

$$Q_j a_i = \zeta^{(i,j)}_1 a_1 + \ldots + \zeta^{(i,j)}_m a_m,$$

then we call $A$ and $S_{sp}$ forward sequential compatible. And call $(\zeta^{(i,j)}_1, \ldots, \zeta^{(i,j)}_m)$ compatible vector of $Q_j a_i$ on $A$.

**Remark 2.3.** From Definition 2.2, we know that, if $A$ and $S_{sp}$ are sequentially compatible, then there hold

$$a_i^T Q^T_1 = \zeta_{i,1} a_1^T + \ldots + \zeta_{i,m} a_m^T,$$

and if $A$ and $S_{sp}$ are forward sequential compatible, then there hold

$$a_i^T Q^T_1 = \zeta^{(i,j)}_1 a_1^T + \ldots + \zeta^{(i,j)}_m a_m^T.$$

Therefore, the definition given through (2.2) and (2.3) is equivalent to the definition through its transpose.

**Remark 2.4.** The definition of forward sequential compatible is actually the constraints for

$$a_1^T Q^T_1, \ldots, a_1^T Q^T_m, \ a_2^T Q^T_1, \ldots, a_2^T Q^T_m, \ldots, \ a_m^T Q^T_m.$$

Moreover, this definition can be extended completely, but we will not do this because it beyond the requirements of this paper.

**Remark 2.5.** Obviously, $Q_m a_m = a_m$, i.e., $\zeta^{(m,m)}_m \equiv 1$. So the definition of forward sequential compatible can be extended to the case of $i = m$, and we do not define it directly because $Q_m$ is an agreed matrix and is not included in $S_{sp}$.

The following theorem shows that $A$ and $S_{sp}$ defined by Kaczmarz’s iteration is forward sequential compatible.

**Theorem 2.6.** Suppose $A$ has non-zero rows, and $S_{sp}$ is defined as (2.1), then $A$ and $S_{sp}$ are forward sequential compatible.

**Proof.** We take the subscript $(i, j)$ of $Q_j a_i$ as an ordered array and prove the conclusion by mathematical induction method.

It is obvious that $a^T_m Q^T_m = a^T_m$, that is, (2.3) holds for $(i, j) = (m, m)$ and $(\zeta^{(m,m)}_1, \ldots, \zeta^{(m,m)}_m) = (0, \ldots, 0, 1)$. In fact, for any $1 \leq i \leq m$, (2.3) obviously holds for $a^T_m Q^T_m$ because $Q_m = I$ is our agreement. Consequently, as the first step of mathematical induction, we prove (2.3) holds for $(i, j) = (m - 1, m - 1)$. Actually,

$$a^T_{m-1} Q^T_{m-1} = a^T_{m-1} P^T_m Q^T_m = a^T_{m-1} = a^T_{m-1} a_m a^T_m \parallel a^T_m \parallel^2.$$ 

So (2.2) holds for $i = m - 1$, where $(\zeta^{(m-1,m-1)}_1, \ldots, \zeta^{(m-1,m-1)}_m) = (0, \ldots, 0, 1, -a^T_{m-1} a_m \parallel a^T_m \parallel^2)$.

Second, we suppose (2.3) holds for any $(i, j)$ satisfied $s < i < m$ and $s \leq t < j < m$, i.e., there exists $(\zeta^{(s,i)}_1, \ldots, \zeta^{(s,i)}_m)$ such that

$$a^T_i Q^T_j = \zeta^{(s,i)}_1 a^T_1 + \ldots + \zeta^{(s,i)}_m a^T_m.$$

Third, we prove (2.3) holds for $(i, j) = (s, t)$. Because of $Q_t = Q_{t+1} P_{t+1}$, then

$$a^T_s Q^T_t = a^T_s P^T_{t+1} Q^T_{t+1} = a^T_s Q^T_{t+1} = a^T_s a^T_{t+1} Q^T_{t+1} \parallel a^T_{t+1} \parallel^2 a^T_{t+1} Q^T_{t+1}.$$
From the hypothesis, there exist \((s_1, s_2, \ldots, s_m)\) and \((t_{s_1}, t_{s_2}, \ldots, t_{s_m})\) such that
\[
a^T_{i}Q_{t+1} = \zeta_i(s_{t+1})a^T_{i1} + \ldots + \zeta_i(s_{t+1})a^T_{im},
\]
\[
a^T_{t+1}Q_{i} = \zeta_t(t_{t+1})a^T_{1} + \ldots + \zeta_t(t_{t+1})a^T_{m}.
\]
Then it follows from (2.4) that
\[
a^T_{t}Q_{t} = \zeta_t(s_{t+1})a^T_{1} + \ldots + \zeta_t(s_{t+1})a^T_{m} - \frac{a^T_{t+1}a^T_{t+1}}{\|a^T_{t+1}\|^2}\zeta_t(t_{t+1})a^T_{m}.
\]
Denote
\[
\zeta(s_{t}, \ldots, \zeta(s_{t+1})) = \zeta(s_{t}) - \frac{a^T_{t+1}a^T_{t+1}}{\|a^T_{t+1}\|^2}\zeta(t_{t+1}), \ldots, \zeta(s_{t+1}) - \frac{a^T_{t+1}a^T_{t+1}}{\|a^T_{t+1}\|^2}\zeta(t_{t+1}).
\]
This proves that (2.3) holds for \((i, j) = (s, t)\).

To sum up the above, the conclusion is proved for all \((i, j)\) with respect to \(1 \leq i \leq m, i \leq j \leq m\). Namely, \(A\) and \(A_{S_{sp}}\) generated by Kaczmarz’s iteration are forward sequential compatible.

From Theorem 2.6, we have the following decomposition corollary of \(A_S\).

**Corollary 2.7.** Under the condition of Theorem 2.6, there exists \(C \in \mathbb{R}^{m \times m}\) such that
\[
A_S = CA,
\] (2.5)

here we call \(C\) the **compatible matrix** of \(A\) and \(S_{sp}\).

**Proof.** According to \(A_S = (Q_1a_1, \ldots, Q_ma_m)^T\) and Theorem 2.6, the corollary can be proved by taking \(C(i, j) = \zeta_{ij}\).

**Remark 2.8.** Corollary 2.7 is valuable for the analysis of the Kaczmarz-Tanabe method, which can lead to the standard form of Kaczmarz-Tanabe’s iteration (i.e., the matrix-vector form). In fact, it follows from (1.5) and Corollary 2.7 that
\[
y_{k+1} = y_k + A^TC^TM(b - Ay_k), \quad k = 0, 1, 2, \ldots.
\] (2.6)

We can hardly see the shadow of Kaczmarz iteration from (2.6), and it more like a member in SIRT methods. The Kaczmarz’s iteration is ever known as algebraic reconstruction technique (ART). However, the appearance of (2.6) confuses the boundaries of ART and SIRT methods and make the Kaczmarz method as easy to use as SIRT methods after obtaining \(C\).

In the above, the matrix \(C\) exists in theory. For the purpose of dealing with its computational problem, the intuitive idea is to find a matrix \(C \in \mathbb{R}^{m \times m}\) to satisfy \(A_S = CA\). For simplicity, we introduce the following notation:
\[
H = (h_{i,j}) := AA^TM,
\]
which yields \(h_{i,j} = a^T_i a_j / \|a_j\|^2\). Additionally, we propose the concept of index set.

**Definition 2.9.** The index set \(I_d(n_1, n_2, v)\) is defined as follows
\[
I_d(n_1, n_2, v) = \{I_d(1), \ldots, I_d(v)\},
\]
where \(n_1, n_2, v\) are positive integers satisfied \(|n_1 - n_2| \geq v \geq 2\). \(I_d(i)\) is integer between \(n_1\) and \(n_2\), and \(I_d(1) = n_1, I_d(v) = n_2\). For any \(i < j\) there satisfy
\[
I_d(i) < I_d(j), \quad n_1 < n_2,
\]
\[
I_d(i) > I_d(j), \quad n_1 > n_2.
\]
Lemma 2.10. Suppose that \( I_d(n_1, n_2, v) \) is actually a set of arrays and the elements in every array are arranged by order, e.g.,

\[
I_d(1, 4, 2) = \{[1, 4]\}, \quad I_d(4, 1, 2) = \{[4, 1]\},
\]
\[
I_d(4, 1, 3) = \{[1, 4, 2], [1, 3, 4]\}, \quad I_d(4, 1, 3) = \{[4, 2, 1], [4, 3, 1]\}.
\]

We must pay attention to the difference of order. In \([1, 4]\), \( I_d(1) = 1, I_d(2) = 4 \); and in \([4, 1]\), \( I_d(1) = 4, I_d(2) = 1 \).

Based on the above definition, we give the expression of \( a_i^T Q_i^T \tilde{x} \) when \( \tilde{x} \in N(A) \).

**Lemma 2.10.** Suppose that \( A \) has non-zero rows, \( Q_i \) is the sequential projection matrix of \( A \) and \( \tilde{x} \in N(A) \). For any \( 1 \leq i \leq m, 1 \leq j \leq m - 1 \), denote

\[
d_{i,j} = \sum_{v=2}^{j-i+1} (-1)^{v-1} \sum_{I_d(i,j,v)} \prod_{s=1}^{v-1} h_{I_d(s), I_d(s+1)}.
\]

Then there holds

\[
a_i^T Q_i^T \tilde{x} = [1, d_{i,i+1}, \ldots, d_{i,m}][a_i^T, a_{i+1}^T, \ldots, a_{m}^T]^T \tilde{x}.
\]

That is, the compatible vector of \( a_i^T Q_i^T \tilde{x} \) on \( A \tilde{x} \) is \([0, \ldots, 0, 1, d_{i,i+1}, \ldots, d_{i,m}]\).

**Proof.** When \( 1 \leq i \leq m \) and \( \tilde{x} \in N(A) \), we have

\[
a_i^T Q_i^T \tilde{x} = (1, -h_{i,v+1})(a_i^T Q_{i+1}^T \tilde{x}, a_{i+1}^T Q_{i+1}^T \tilde{x})^T
\]
\[
= (1, -h_{i,v+1}, -h_{i,v+2} + h_{i,v+1}h_{i,v+2}h_{i,v+3}h_{i,v+4}) (a_{i}^T Q_{i+1}^T \tilde{x}, a_{i+1}^T Q_{i+2}^T \tilde{x}, a_{i+2}^T Q_{i+3}^T \tilde{x})^T
\]
\[
= (1, -h_{i,v+1}, \ldots, \sum_{v=2}^{m-i-1} (-1)^{v-1} \prod_{I_d(i,m,v)} h_{I_d(s), I_d(s+1)}^T) (a_i^T Q_i^T \tilde{x}, a_{i+1}^T Q_{i+1}^T \tilde{x}, \ldots, a_m^T Q_m^T \tilde{x})^T.
\]

Thus (2.8) holds by taking \( d_{i,j} \) according to (2.7).

From the proof of Lemma 2.10, \( d_{i,j} \) is equivalent to the lengthy but intuitive form, i.e.,

\[
\sum_{v=2}^{j-i+1} (-1)^{v-1} \prod_{I_d(i,j,v)} h_{I_d(s), I_d(s+1)} = -h_{i,i+1} + h_{i,i+1}h_{i,i+2} + \ldots + (-1)^{j-i} h_{i,i+1}h_{i,i+2} \ldots h_{j-1,j}.
\]

**Theorem 2.11.** Under the Lemma 2.10, let \( \Omega = (\omega_{i,j})_{m \times m} \) satisfy

\[
\omega_{i,j} = \begin{cases} 
  d_{i,j}, & j > i, \\
  1, & j = i, \\
  0, & j < i. 
\end{cases}
\]

Then there holds

\[
A_S = \Omega A,
\]

where \( A_S \) is defined as (1.9).

**Proof.** For any \( \tilde{x} \in N(A) \), it following from (1.9) that

\[
A_S \tilde{x} = (a_1^T Q_1^T \tilde{x}, a_2^T Q_2^T \tilde{x}, \ldots, a_m^T Q_m^T \tilde{x})^T.
\]

From Lemma 2.10 and (2.10), then we get

\[
A_S \tilde{x} = \Omega A \tilde{x}.
\]

When \( \tilde{x} \in N(A) \), (2.12) obviously holds. Therefore, for any \( \tilde{x} \in \mathbb{R}^n \), there holds \( A_S \tilde{x} = \Omega A \tilde{x} \), which means \( A_S = \Omega A \).
Lemma 2.10 and Theorem 2.11 actually show us a specific form $\Omega$ of matrix $C$, Thus we get
\begin{equation}
y_{k+1} = y_k + A^T \Omega M (b - Ay_k), \quad k = 0, 1, 2, \ldots
\end{equation}
(2.13)

If $A$ is a full row rank matrix, the decomposition of $A_S$ is unique, i.e., $C \equiv \Omega$.

We specifically refer to (2.13) as the *standard form of Kaczmarz-Tanabe’s iteration* and still denote by (2.6) with $C = \Omega$.

Let $E(j, i(-h_{j,i}))$ be a matrix obtained by multiplying the $i$-th row of the identity matrix by $-h_{j,i}$ and adding it to the $j$-th row, i.e., the diagonal elements of $E(j, i(-h_{j,i}))$ are all 1, the $(j, i)$-element is $-h_{j,i}$, and all other elements are 0. Consequently, we have the following theorem.

**Theorem 2.12.** If $\Omega$ is defined as (2.10), then there holds
\begin{equation}
\Omega = H_1 H_2 \cdots H_m,
\end{equation}
(2.14)
where $H_1 = I$ and $H_i = \prod_{j=1}^{i-1} E(j, i(-h_{j,i}))$ for any $1 < i \leq m$.

**Proof.** For any $\tilde{x} \in N(A^T)$, we denote $b = A \tilde{x}$. From (2.11), we have
\begin{equation}
a_j^T \tilde{x} = (0, \ldots, 1, -h_{j,j+1}, \ldots, -h_{j,m} + h_{j,m-1} h_{m-1,j} + \ldots + (-1)^{m-1-j} h_{j,j+1} h_{j,j+2} \cdots h_{m,j})
\end{equation}
\cdot (b_1, \ldots, b_{m-2}, b_{m-1}, b_m)^T.
(2.15)

From (2.11), the coefficient of $b_i(i > j)$ in (2.15) is actually the $(j, i)$-element of $\Omega$, i.e.,
\begin{equation}
\omega_{j,i} = -h_{j,i} + h_{j,i-1} h_{i-1,i} + \ldots + (-1)^{i-j} h_{j,j+1} h_{j,j+2} \cdots h_{m,1-i}.
\end{equation}

Denote $\tilde{H} = H_1 \cdots H_m$, in order to show $\Omega = H_1 \cdots H_m$, we only need to prove $\omega_{j,i} = \tilde{H}_{j,i}$ (where $i > j$), i.e.,
\begin{equation}
\omega_{j,i} = e_j^T \tilde{H} e_i,
\end{equation}
where $e_j$ and $e_i$ are the $j$th and $i$th columns of the identity matrix in $\mathbb{R}^m$ [25, p72], respectively. Owing to $e_j^T H_k = e_j^T$ when $j \geq k$ and $H_i e_i = e_i$ when $i \neq l$, then it follows from the notation of $\tilde{H}$ that when $i > j$,
\begin{equation}
e_j^T \tilde{H} e_i = e_j^T H_{j+1} \cdots H_k e_i
= (e_j^T H_{j+1} H_{j+2} \cdots H_{k+1} H_{k+2} \cdots H_k e_i)
= (0, \ldots, 1, -h_{j,j+1}, 0, \ldots, 0) H_{j+1} \cdots H_k e_i
= (-h_{j,i} + h_{j,i-1} h_{i-1,i} + \ldots + (-1)^{i-j} h_{j,j+1} h_{j,j+2} \cdots h_{m,1-i}, 0, \ldots, 0) e_i
= -h_{j,i} + h_{j,i-1} h_{i-1,i} + \ldots + (-1)^{i-j} h_{j,j+1} h_{j,j+2} \cdots h_{m,1-i}.
\end{equation}

This proves $\omega_{j,i} = \tilde{H}_{j,i}$ for any $1 \leq j \leq n - 1$ and $i > j$. Additionally, there holds $\omega_{j,j} = \tilde{H}_{j,j} = 1$ for any $1 \leq j \leq n$. Consequently, the conclusion is proved.

Theorem 2.12 actually gives the calculation formula of $\Omega$ that is defined in (2.10). But it is not good idea to calculate $\Omega$ directly according to (2.14) because the calculation speed may be slow. In fact, the matrix $\Omega$ can be calculated in parallel mode when we divide the multiplication of $H_1 H_2 \cdots H_m$ into several small parts, but we should notice that the block operation is executed on matrix $\Omega$ but not on the whole linear system. Consequently, if we perform the block operation on linear systems and solve each linear subsystem with the Kaczmarz-Tanabe method, which indeed can reduce the cost of calculating $\Omega$, will lead to the block Kaczmarz-Tanabe method.

### 3 The standard form of symmetric Kaczmarz-Tanabe method and its convergence

In this section, we mainly consider the standard form of the symmetric Kaczmarz-Tanabe’s iteration and analyze its convergence rate, then compare the convergence rates between Kaczmarz-Tanabe’s iteration and symmetric Kaczmarz-Tanabe’s iteration.
Let \( \{x_k, k > 0 \} \) be the vector sequence determined by (1.3) and (1.4). Denote
\[
\bar{y}_{k+1} = x_{k(2m-2)+m}, \quad y_{k+1} = x_{(k+1)(2m-2)}, \quad k = 0, 1, \ldots,
\]
then from (2.6) there holds
\[
\bar{y}_{k+1} = y_k + A^T C^T M(b - Ag_k),
\]
which is the Kaczmarz-Tanabe’s iteration.

Next, we consider the iterative formula of Kaczmarz-Tanabe method for Kaczmarz’s projection from equation \( m - 1 \) to equation 2 in reverse order. Define
\[
\bar{Q}_i = P_{2i-1} \ldots P_{i-1}, \quad i = 3, \ldots, m - 1,
\]
thus \( \bar{Q}_i \) is the sequential projection matrix on \( (a_{i-1}, \ldots, a_2)^T \), and for \( 3 \leq i \leq m - 1 \),
\[
\bar{Q}_i = \bar{Q}_{i-1} P_{i-1}.
\]
The sequential projection matrix set reads
\[
S_{sp}(a_{m-1}, \ldots, a_2) = \{ \bar{Q}_3, \ldots, \bar{Q}_{m-1} \}.
\]
Additionally, we agree to
\[
\bar{Q}_1 = \bar{Q}_m = 0 \in \mathbb{R}^{m \times m}, \quad \bar{Q}_2 = I,
\]
and denote
\[
\bar{Q} := P_2 \ldots P_{m-1}.
\]
So the symmetric Kaczmarz’s projection can be written as
\[
y_{k+1} = \bar{Q} \bar{y}_{k+1} + \bar{A}_S^T Mb,
\]
where
\[
\bar{A}_S = [\bar{Q}_1 a_1, \bar{Q}_2 a_2, \bar{Q}_3 a_3, \ldots, \bar{Q}_{m-1} a_{m-1}, \bar{Q}_a a_m]^T.
\]
Note that we do not use the expression of ‘symmetric Kaczmarz-Tanabe iteration’ here because it is not the symmetric Kaczmarz-Tanabe iteration at this time and is only the symmetric part of symmetric Kaczmarz’s iteration (i.e., the case of Kaczmarz’s projection (1.3) for \( i = m - 1, \ldots, 2 \)).

Before deriving the standard form of the symmetric Kaczmarz-Tanabe’s iteration, we first give the relationship between \( Q \) and \( \bar{A}_S \) appeared in (3.7).

**Lemma 3.1.** Suppose \( A \) has non-zero rows, and \( \bar{Q} \) and \( \bar{A}_S \) are defined as (3.6) and (3.8), then
\[
\bar{Q} = I - \bar{A}_S^T MA,
\]
where \( M \) is defined in (1.10).

**Proof.** From (3.6) and (3.4), we have
\[
\bar{Q} = P_2 \ldots P_{m-1}
\]
\[
= \bar{Q}_{m-1} - \bar{Q}_{m-1} \frac{a_{m-1} a_{m-1}^T}{\|a_{m-1}\|^2}
\]
\[
= \ldots
\]
\[
= \bar{Q}_2 - \bar{Q}_2 \frac{a_2 a_2^T}{\|a_2\|^2} - \bar{Q}_3 \frac{a_3 a_3^T}{\|a_3\|^2} - \ldots - \bar{Q}_{m-2} \frac{a_{m-2} a_{m-2}^T}{\|a_{m-2}\|^2} - \bar{Q}_{m-1} \frac{a_{m-1} a_{m-1}^T}{\|a_{m-1}\|^2} - \bar{Q}_m \frac{a_m a_m^T}{\|a_m\|^2}.
\]
From (3.5), it follows
\[ Q = I - Q_1 \frac{a_1 a_1^T}{\|a_1\|_2^2} - Q_2 \frac{a_2 a_2^T}{\|a_2\|_2^2} - Q_3 \frac{a_3 a_3^T}{\|a_3\|_2^2} - \cdots - Q_{m-2} \frac{a_{m-2} a_{m-2}^T}{\|a_{m-2}\|_2^2} - Q_{m-1} \frac{a_{m-1} a_{m-1}^T}{\|a_{m-1}\|_2^2} - Q_m \frac{a_m a_m^T}{\|a_m\|_2^2} \]
\[ = I - (Q_1 a_1, Q_2 a_2, \ldots, Q_m a_m) \text{diag}(\frac{1}{\|a_1\|_2^2}, \frac{1}{\|a_2\|_2^2}, \ldots, \frac{1}{\|a_m\|_2^2})(a_1, a_2, \ldots, a_m)^T. \]

This proves (3.9). □

According to Lemma 3.1, we get the equivalent form of (3.7),
\[ y_{k+1} = y_{k+1} + \tilde{A}_S^T M (b - A y_{k+1}). \]

We should notice that (3.10) is not the final form of the symmetric Kaczmarz-Tanabe's iteration because it does not include the Kaczmarz projection process from \( i = 1 \) to \( m \). We next consider the matrix-vector form of (3.10). First, we have the following existence theorem.

**Theorem 3.2.** Suppose \( A \) has non-zero rows, then there exists \( \hat{C} \) such that
\[ \hat{A}_{S} = \hat{C} A. \]

**Proof.** Similar to Theorem 2.6 and Corollary 2.7, the existence of \( \hat{C} \) can be proved, we omit the process here. □

Because \( Q_1 = Q_m = 0 \), this feature causes the first and last rows of \( \hat{C} \) to be 0. Moreover, in view of \( Q = I \), then the first elements of rows from 2 to \( m - 1 \) are 0. These are the major difference from \( C \). We next consider the specific expression of \( \hat{C} \).

**Lemma 3.3.** Suppose \( A \) has non-zero rows and \( Q_i \) is defined as (3.3) and \( \bar{x} \in N(A) \). Let
\[ \tilde{d}_{i,j} = \sum_{v=2}^{i-j+1} (-1)^{v-1} \sum_{I_d(i,j,v)} \prod_{s=1}^{v-1} h_{I_d(s), I_d(s+1)}, \]
then for any \( 2 \leq i \leq m - 1 \) and \( 2 \leq j \leq i - 1 \), there holds
\[ a_i^T Q_i^T \bar{x} = [0, \tilde{d}_{i,2}, \ldots, \tilde{d}_{i,i-1}, 1, \ldots, 0][a_1, \ldots, a_m]^T \bar{x}. \]

**Proof.** Obviously, when \( 2 \leq i \leq m - 1 \) and \( \bar{x} \in N(A) \), there holds from (3.4) that
\[
\begin{align*}
a_i^T Q_i^T \bar{x} &= (-h_{i,i-1,1})(a_{i-1}^T Q_{i-1}^T \bar{x}, a_i^T Q_i^T \bar{x})^T \\
&= (-h_{i,i-1,1})(a_{i-1}^T Q_{i-1}^T \bar{x}, a_i^T Q_i^T \bar{x})^T \\
&= (-h_{i,i-2} + h_i, i-1, i-2, -h_{i,i-1,1})(a_{i-2}^T Q_{i-2}^T \bar{x}, a_{i-1}^T Q_{i-1}^T \bar{x}, a_i^T Q_i^T \bar{x})^T \\
&= \cdots \\
&= (-h_{i,2} + \sum_{k=3}^{i-1} h_{i,k} h_{k,2} + \cdots + (-1)^{i-2} \sum_{k=2}^{i-1} h_{k+1,k}, \ldots, -h_{i,i-1,1})(a_2^T \bar{x}, \ldots, a_i^T \bar{x})^T.
\end{align*}
\]

Because \( \tilde{Q} \equiv I \), then taking \( \tilde{d}_{i,j} \) in (3.14) according to (3.12) yields
\[ a_i^T Q_i^T \bar{x} = (\tilde{d}_{i,2}, \ldots, \tilde{d}_{i,i-1,1})(a_2, \ldots, a_i)^T \bar{x}, \]
this proves (3.13). □

Similar to Theorem 2.11, we have the following theorem.

**Theorem 3.4.** Under the Lemma 3.3, let \( \tilde{\Omega} = (\tilde{\omega}_{i,j})_{m \times m} \) satisfy
\[ \tilde{\omega}_{i,j} = \begin{cases} 
\tilde{d}_{i,j}, & 1 < i < m, 2 < j < i - 1, \\
0, & 1 < i < m, j = 1 \lor j > i, \\
1, & 1 < i < m, j = i, \\
0, & i = 1 \lor m, 1 \leq j \leq m.
\end{cases} \]
Then there holds
\[ A_S = \hat{\Omega}A, \quad (3.16) \]
where \( A_S \) is defined as (3.8).

\textbf{Proof.} For any \( \bar{x} \in N(A)^\perp \), it follows from (3.8) that
\[ A_S \bar{x} = (a_1^T Q_1^T \bar{x}, a_2^T Q_2^T \bar{x}, \ldots, a_m^T Q_m^T \bar{x})^T. \quad (3.17) \]
By Lemma 3.3 and (3.15), we get
\[ A_S \bar{x} = \hat{\Omega}A\bar{x}. \quad (3.18) \]
When \( \bar{x} \in N(A) \), from [13, Corollary 2.2], there holds \( A_S \bar{x} = 0 \), thus (3.18) also holds. Then for any \( \bar{x} \in \mathbb{R}^n \), there holds \( A_S \bar{x} = \hat{\Omega}A\bar{x} \), which means \( A_S = \hat{\Omega}A \).

Let
\[ E((j, i)(-h_{j, i}))(s, t) = \begin{cases} E(j, i(-h_{j, i}))(s, t), & (s, t) \neq (1, 1) \land (m, m), \\ 0, & (s, t) = (0, 0) \lor (m, m). \end{cases} \quad (3.19) \]
Similar to Theorem 2.12, we have the following decomposition of \( \hat{\Omega} \).

\textbf{Theorem 3.5.} If \( \hat{\Omega} \) is defined as Theorem 3.4, then there holds
\[ \hat{\Omega} = \hat{H}_{m-1} \hat{H}_{m-2} \cdots \hat{H}_2, \quad (3.20) \]
where \( \hat{H}_i = \prod_{j=i+1}^{m-1} \hat{E}(j, i(-h_{j, i})) \) for any \( 2 \leq i \leq m - 1 \).

\textbf{Proof.} Denote \( \hat{H} = \hat{H}_{m-1} \hat{H}_{m-2} \cdots \hat{H}_2 \), obviously, \( \hat{\Omega} \) and \( \hat{H} \) are unit upper triangular matrices with the same order. Therefore, we only need to prove that the non-zero elements are equal. For any \( \bar{x} \in N(A)^\perp \), from (3.14) and \( Q_2 = I \), we have
\[ a_i^T Q_i^T \bar{x} = (-h_{i, 2} + \sum_{k=3}^{i-1} h_{i, k} h_{k, 2} + \ldots + (-1)^{i-2} \prod_{k=2}^{i-1} h_{k+1, k}, \ldots, \ldots, \ldots, -h_{i, i-1}, 1)(a_2^T \bar{x}, \ldots, a_i^T \bar{x})^T. \quad (3.21) \]
In (3.21), the coefficient of \( a_j^T \bar{x} (2 \leq j < i) \) is actually the \((i, j)\)-element of \( \hat{\Omega} \), i.e.,
\[ \hat{\omega}_{i, j} = -h_{i, j} + \sum_{k=j+1}^{i-1} h_{i, k} h_{k, j} + (-1)^{i-j} \prod_{k=j}^{i-1} h_{k+1, k}. \]
In order to show \( \hat{\Omega} = \hat{H}_{m-1} \hat{H}_{m-2} \cdots \hat{H}_2 \), we only need to prove \( \hat{\omega}_{i, j} = \hat{H}_{i, j} \) (where \( \hat{H}_{i, j} \) denotes the \((i, j)\)-element of \( \hat{H} \)), i.e.,
\[ \hat{\omega}_{i, j} = e_i^T \hat{H} e_j, \]
Owing to \( e_i^T \hat{H} k = e_i^T \) when \( i \leq k \) and \( i = m \), and \( \hat{H} e_j = e_j \) when \( j \neq l \), then it follows when \( 2 \leq i \leq m - 1 \) and \( 2 \leq j < i \),
\[ e_i^T \hat{H} e_j = e_i^T \hat{H}_{i-1} \hat{H}_{i-2} \cdots \hat{H}_j e_j \]
\[ = (e_i^T \hat{H}_{i-1}) \hat{H}_{i-2} \cdots \hat{H}_j e_j \]
\[ = (0, \ldots, h_{i, i-1}, 1, 0, \ldots, 0) \hat{H}_{i-2} \cdots \hat{H}_j e_j \]
\[ = (0, \ldots, 0, -h_{i, i-2} + h_{i, i-1} h_{i-1, i-2}, \ldots, -h_{i, i-1}, 1, 0, \ldots, 0) \hat{H}_{i-3} \cdots \hat{H}_j e_j \]
\[ = -h_{i, j} + \sum_{k=j+1}^{i-1} h_{i, k} h_{k, j} + \ldots + (-1)^{i-j} h_{i, i-1} h_{i-1, i-2} \cdots h_{j+1, j}. \]

10
This proves \( \hat{\omega}_{i,j} = \tilde{H}_{i,j} \) for any \( 2 \leq i \leq m - 1 \) and \( 2 \leq j < i \). Additionally, there holds \( \hat{\omega}_{i,i} = \tilde{H}_{i,i} = 1 \) for any \( 2 \leq i \leq m - 1 \). Consequently, the conclusion is proved.

Compared with Theorem 3.2, \( \tilde{\Omega} \) in Theorem 3.4 gives the specific form of \( \tilde{C} \) and is still denoted with \( \tilde{C} \), thus we get from (3.10) that

\[
y_{k+1} = \tilde{y}_{k+1} + A^T \tilde{C}^T M(b - A\tilde{y}_{k+1}).
\]

(3.22)

Based on (3.2) and (3.22), then we get the following theorem.

**Theorem 3.6.** Suppose \( A \) has non-zero rows, then there exists matrix \( \tilde{C} \in \mathbb{R}^{m \times m} \), such that the symmetric Kaczmarz-Tanabe’s iteration can be written as

\[
y_{k+1} = y_k + A^T \tilde{C}^T M(b - Ay_k).
\]

(3.23)

**Proof.** From (3.2) and (3.22), thus symmetric Kaczmarz-Tanabe’s iteration reads

\[
y_{k+1} = \tilde{y}_{k+1} + A^T \tilde{C}^T M(b - A\tilde{y}_{k+1})
\]

\[
= y_k + A^T (\tilde{C}^T + C^T - \tilde{C}^T MAA^T C^T) M(b - Ay_k).
\]

Denote \( \tilde{C} := \tilde{C}^T + C^T - \tilde{C}^T MAA^T C^T \), then (3.23) is proved.

From (2.13) and (3.23), we know that Kaczmarz-Tanabe’s iteration and symmetric Kaczmarz-Tanabe’s iteration have the same matrix-vector form. And From (3.23), we also have the equivalent form

\[
y_{k+1} = (I - A^T \tilde{C}^T MA)y_k + A^T \tilde{C}^T Mb,
\]

where \( I - A^T \tilde{C}^T MA \) is the iterative matrix of symmetric Kaczmarz-Tanabe method. Considering the principle of the symmetric Kaczmarz’s iteration, thus we have the following corollary.

**Corollary 3.7.** Suppose \( A \) has non-zero rows, then for symmetric Kaczmarz-Tanabe’s iteration, there holds

\[
\tilde{Q}Q = P_2 \ldots P_{m-1} P_m \ldots P_1 = I - A^T \tilde{C}^T MA,
\]

(3.24)

where \( \tilde{C} \) is consistence with that in Theorem 3.6.

Let \( e_k = y_k - P_{N(A)}y_0 - x^\perp \), then it follows from (3.23) that

\[
e_{k+1} = (I - A^T \tilde{C}^T MA)e_k.
\]

(3.25)

For the symmetric Kaczmarz-Tanabe’s iteration, there holds the following set property about \( \{e_k\} \).

**Theorem 3.8.** For any initial vector \( y_0 \in \mathbb{R}^n \), let \( \{y_k, k > 0\} \) be generated by symmetric Kaczmarz-Tanabe’s iteration (3.23), then there holds

\[
e_k \in N(A)^\perp.
\]

**Proof.** We prove the conclusion by mathematical induction method. First, The fact \( e_0 \in N(A)^\perp \) holds because \( y_0 - P_{N(A)}y_0 \) and \( x^\perp \) belong to \( N(A)^\perp \). Second, we suppose for any \( k \geq 0 \) there holds \( e_k \in N(A)^\perp \). Third, we will prove \( e_{k+1} \in N(A)^\perp \). For any \( z \in N(A) \), we have

\[
\langle e_{k+1}, z \rangle = \langle (I - A^T \tilde{C}^T MA)e_k, z \rangle = \langle e_k, z \rangle - \langle \tilde{C}^T MAe_k, Az \rangle = 0,
\]

which proves the conclusion.

**Lemma 3.9.** For any \( 1 \leq i \leq m \), when \( x \in N(A)^\perp \), then there holds

\[
P_i x \in N(A)^\perp.
\]

That is, \( N(A)^\perp \) is the invariant subspace for any \( P_i \).
Proof. For any $1 \leq i \leq m$ and $z \in N(A)$, there holds
\[
(P_i x, z) = \langle (I - \frac{a_i a_i^T}{\|a_i\|^2})x, z \rangle = \langle x, z \rangle - \frac{1}{\|a_i\|^2} (a_i^T x, a_i^T z) = 0.
\]

As can be seen from the proof of Theorem 3.8, Kaczmarz-Tanabe's iteration and symmetric Kaczmarz-Tanabe's iteration have completely uniform set property. Therefore, we can further give an intuitive estimation of $\|P_1 e_{k+1}\|_2$

**Theorem 3.10.** Under the condition of Theorem 3.8, then
\[
\|P_1 e_{k+1}\|_2 \leq \max_{0 < \sigma_i < 1} \sigma_i^2 \|P_1 e_k\|_2, \quad k = 0, 1, \ldots
\]  

where $\sigma_i$ is the singular value of $Q$.

**Proof.** From (3.25), we have
\[
P_1 e_{k+1} = P_1 (I - A^T C^T M A) e_k.
\]

As can be seen from the proof of Theorem 3.8, Kaczmarz-Tanabe's iteration and symmetric Kaczmarz-Tanabe's iteration have completely uniform set property. Therefore, we can further give an intuitive estimation of $\|P_1 e_{k+1}\|_2$

**Theorem 3.11.** Under the condition of Theorem 3.8, for some $k \geq 0$, if $e_{k+1} \in N(P_1)^\perp$, then there holds
\[
\|e_{k+1}\|_2 \leq \max_{0 < \sigma_i < 1} \sigma_i^2 \|P_1^\dagger \|_2 \|e_k\|_2
\]

where $\sigma_i$ is the singular value of $Q$ and $P_1^\dagger$ denotes the pseudo-inverse of $P_1$.

**Proof.** When $e_{k+1} \in N(P_1)^\perp$, we have
\[
P_1^\dagger P_1 e_{k+1} = e_{k+1},
\]

hence
\[
\|e_{k+1}\|_2 \leq \|P_1^\dagger \|_2 \|P_1 e_{k+1}\|_2.
\]

And from (3.26), we get (3.30).

The equality (3.31) depends on $e_{k+1} \in N(P_1)^\perp$, if the latter is not satisfied, then (3.30) may not hold. In the following theorem, we give a general conclusion without constraint condition $e_{k+1} \in N(P_1)^\perp$.

**Theorem 3.12.** Under the condition of Theorem 3.8, then for any $k \geq 0$, at least one of the following statements is true,

(i) $\|e_{k+1}\|_2 < \max_{0 < \sigma_i < 1} \sigma_i \|e_k\|_2$;
(ii) $\|e_{k+2}\|_2 < \max_{0 < \sigma_i < 1} \sigma_i^2 \|e_k\|_2$;

where $\sigma_i$ is the singular value of $Q$. 

12
Because the method is better than that of the Kaczmarz-Tanabe method (because it has been made from 'Remark 3.13. From Theorem 3.12 we can see that the convergence rate of the symmetric Kaczmarz-Tanabe method will perform $2m - 2$ orthogonal projections (i.e., perform projection on the plane determined by equation 1 to m and then vice versa from m - 1 to 2), while the Kaczmarz-Tanabe method only makes m orthogonal projections. Consequently, it is relative fair to compare one symmetric Kaczmarz-Tanabe’s iteration and two Kaczmarz-Tanabe’s iterations. We suppose $y_k$ is the same initial vector for the Kaczmarz-Tanabe’s iteration and the symmetric Kaczmarz-Tanabe’s iteration, then the convergence rate result of two Kaczmarz-Tanabe’s iterations is

$$
\|\tilde{e}_{k+2}\|_2 \leq \max_{0 < \sigma_i < 1} \sigma_i^2 \|\tilde{e}_k\|_2,
$$

which is prior to the results of symmetric Kaczmarz-Tanabe’s iteration (see Theorem 3.12(i)).
Remark 3.14. As can be seen from (2.6) and (3.23), the Kaczmarz-Tanabe method and the symmetric Kaczmarz-Tanabe method have the same iterative formula, but $C$ is different from $\bar{C}$. When $C$ and $\bar{C}$ are known, one iteration of the Kaczmarz-Tanabe method is equivalent to $m$ iterations of the Kaczmarz method, while one iteration of the symmetric Kaczmarz-Tanabe method is equivalent to $2m-2$ iterations of the Kaczmarz method. From this point of view, the calculation efficiency of the symmetric Kaczmarz-Tanabe method is higher than that of the Kaczmarz-Tanabe method.

4 The related algorithms

For the Kaczmarz-Tanabe method, the core work is to generate matrix $C$. Once $C$ is obtained, the Kaczmarz-Tanabe’s iteration is easy to perform. Algorithm 1 shows the process flow of calculating $C$.

Algorithm 1 The calculation of matrix $C$

1: Input
2: $A = (a_1, a_2, \ldots, a_m)^T$
3: $C = I_m$ $\triangleright I_m$ is an identity matrix with order $m$
4: $k \leftarrow m$
5: while $k > 1$ do
6: $i \leftarrow k$
7: while $i > 1$ do
8: $j \leftarrow m$
9: while $j > k - 1$ do
10: if $a_i^T a_k = 0$ then
11: $C(i-1,j) = C(i-1,j)$
12: else
13: $C(i-1,j) = C(i-1,j) + (-a_{i-1}^T a_k / a_i^T a_k)C(k,j)$
14: end if
15: $j \leftarrow j - 1$
16: end while
17: $i \leftarrow i - 1$
18: end while
19: $k \leftarrow k - 1$
20: end while
21: Output $C$

For the symmetric Kaczmarz-Tanabe method, $\hat{C} = \hat{C}^T + C^T - \hat{C}^T M A^T C^T$, where $C$ is the matrix obtained by Algorithm 1. Therefore, we only need to compute $\hat{C}$ in order to perform the symmetric Kaczmarz-Tanabe’s iteration. Algorithm 2 shows the process flow to compute $\hat{C}$.

Algorithm 2 The calculation of matrix $\hat{C}$

1: Input
2: $A = (a_1, a_2, \ldots, a_m)^T$
3: $\hat{C} = I_m$ $\triangleright I_m$ is an identity matrix with order $m$
4: $k \leftarrow m - 1$
5: while $k > 1$ do
6: $i \leftarrow m - 1$
7: while $i > k$ do
8: $j \leftarrow k$
9: while $j > 1$ do

5.1 Tanabe problem

5 Numerical tests

To [26–30] for more details). Blocking technology can be made on the linear system which has been discussed in some articles (please refer to [26–30] for more details).

5.1 Tanabe problem

Considering the following linear system with equations

\[
\begin{bmatrix}
1.0 & 3.0 & 2.0 & -1.0 \\
1.0 & 2.0 & -1.0 & -2.0 \\
1.0 & -1.0 & 2.0 & 3.0 \\
2.0 & 1.0 & 1.0 & 1.0 \\
5.0 & 5.0 & 4.0 & 1.0 \\
4.0 & -1.0 & 5.0 & 7.0
\end{bmatrix}
\begin{bmatrix}
x_1 \\
x_2 \\
x_3 \\
x_4
\end{bmatrix}
= \begin{bmatrix}
5.0 \\
0.0 \\
5.0 \\
15.0 \\
15.0
\end{bmatrix},
\]

(5.1)

Linear system (5.1) is consistent and over-determined, the general solution is

\[
(x_1, x_2, x_3, x_4)^T = k(-2/3, 1, -2/3, 1)^T + (5/3, 0, 5/3, 0)^T,
\]

(5.2)

where \( k \in \mathbb{C} \) is any constant and \( \mathbb{C} \) is complex field. In numerical experiments, \( x^* = (1, 1, 1, 1)^T \) is taken as the test solution. We compare the convergence rates of the Kaczmarz-Tanabe and symmetric Kaczmarz-Tanabe methods on the one hand, and compare that of the Kaczmarz-Tanabe type methods and SIRT methods on the other hand.

Numerical results are shown in Figures 5.1~5.2, where Figure 5.1 shows the error curves of \( \|y_k - x^*\|_2 \), \( \|y_k - x^*\|_2 \), and \( \|y_k - x^* - P_N(A)x_0\|_2 \) when \( x_0 = (7, 6, 10, 6)^T \), and Figure 5.2 shows the corresponding results when \( x_0 = (0, 0, 0, 0)^T \). In Figures 5.1(a)(c)(c) and 5.2(a)(c), we compare the errors of Kaczmarz-Tanabe method, symmetric Kaczmarz-Tanabe method and two iterations Kaczmarz-Tanabe method (marked with 'Kaczmarz-Tanabe(2)' in these figures).

In Figures 5.1(b)(d)(f) and 5.2(b)(d), we compare the errors of Kaczmarz-Tanabe method, symmetric Kaczmarz-Tanabe method, Cimmino method, DROP method, SART method, CAV method and CGMN method when \( x_0 = (7, 6, 10, 6)^T \) and \( x_0 = (0, 0, 0, 0)^T \) respectively. Since the calculation amount of the Kaczmarz-Tanabe method and symmetric Kaczmarz-Tanabe method is roughly the same as that of SIRT methods when

```plaintext
10: if \( a_k^T a_k = 0 \) then
11: \( \hat{C}(i,j) = \hat{C}(i,j) \)
12: else
13: \( \hat{C}(i,j) = \hat{C}(i,j) + (-a_k^T a_k/\bar{a}_k^T a_k)\hat{C}(k,j) \)
14: end if
15: \( j \leftarrow j - 1 \)
16: end while
17: \( i \leftarrow i - 1 \)
18: end while
19: \( k \leftarrow k - 1 \)
20: end while
21: \( \hat{C}(1,1) = 0, \hat{C}(m,m) = 0 \)
22: Output \( \hat{C} \)
```

For Kaczmarz-Tanabe’s iteration and symmetric Kaczmarz-Tanabe’s iteration, the matrices \( C \) and \( \hat{C} \) are invariant in the subsequent iterations which is benefit for computation, e.g., in medical imaging equipment, people can calculate and store the matrices \( C \) and \( \hat{C} \) or related matrices in imaging device in advance. \( C \) and \( \hat{C} \) can be calculated by block mode or parallel block mode, which will greatly reduce the cost to compute them. Blocking technology can be made on the linear system which has been discussed in some articles (please refer to [26–30] for more details).
$C$ and $\bar{C}$ are determined, therefore we deal with these method with the same way, that is, comparing one Kaczmarz-Tanabe’s iteration with one symmetric Kaczmarz-Tanabe’s iteration, as well as other methods.

In Figure 5.1, (a), (b) are the same as (e), (f) respectively, although they look different. Denote

$$\xi = \left(\frac{-2}{3}, 1, \frac{-2}{3}, 1\right)^T,$$

we know from (5.2) that $N(A) = \text{span}\{\xi\}$. Thus

$$P_{N(A)}x_0 = P_{N(A)}x^* = \frac{\xi^T x_0}{\|\xi\|^2} \xi = \frac{3}{13} \left(\frac{-2}{3}, 1, \frac{-2}{3}, 1\right)^T,$$

$$x^* = x^\dagger + P_{N(A)}x_0,$$

which means that

$$\|y_k - x^*\|_2 = \|y_k - x^\dagger - P_{N(A)}x_0\|_2.$$ 

Therefore, the convergence of the error curves in Figure 5.1 (a), (b), (e) and (f) are consistent with the theoretical results, and this is also why the curves in Figure 5.1 (c) and (d) do not tend to $x$-axis.

In addition, Figure 5.1 (a), (c) and (e) also show that symmetric Kaczmarz-Tanabe’s iteration is better than Kaczmarz-Tanabe’s iteration, and slightly worse than two iteration Kaczmarz-Tanabe’s iteration. Meanwhile, Figure 5.1 (b), (d) and (f) show that the convergence speed of the Kaczmarz-Tanabe and symmetric Kaczmarz-Tanabe methods is faster than the SIRT methods.

Figure 5.2 shows the efficiency of these methods when $x_0 = (0, 0, 0, 0)^T$. Figure 5.2 (a) is slightly different from Figure 5.1 (a), and Figure 5.2 (c) is consistent with Figure 5.1 (e). It seems from Figure 5.2 (b) that the SART method is better than others, the reason is that the SART’s iteration converges to $x^*$ rather than $x^\dagger$ when $x_0 = (0, 0, 0, 0)^T$, which can be seen from Figure 5.2 (d).

We also note that the CGMN method is sensitive to iteration step, and converges quickly at the beginning, and then the results become worse. Suppose the linear system to be solved by CGMN method is $Bx = c$, this
Figure 5.1: The comparison of errors $\|y_k - x^*\|_2$, $\|y_k - x^\dagger\|_2$ and $\|y_k - x^\dagger - P_{N(A)}x_0\|_2$ when $x_0 = (7, 6, 10, 6)^T$, where (a),(c) and (e) are comparisons among the Kaczmarz-Tanabe method, the symmetric Kaczmarz-Tanabe method and two iterations Kaczmarz-Tanabe method for solving Tanabe problem, and (b), (d) and (f) are comparisons among the Kaczmarz-Tanabe method, the symmetric Kaczmarz-Tanabe method and SIRT type methods for solving Tanabe problem. (see (1.11) for the iterative schemes).

Figure 5.2: The comparison of errors $\|y_k - x^*\|_2$, $\|y_k - x^\dagger\|_2$ when $x_0 = (0, 0, 0, 0)^T$. where (a) and (c) are comparisons among the Kaczmarz-Tanabe method, the symmetric Kaczmarz-Tanabe method and two iterations Kaczmarz-Tanabe method for solving Tanabe problem, and (b) and (d) are comparisons among the Kaczmarz-Tanabe method, the symmetric Kaczmarz-Tanabe method and SIRT type methods for solving Tanabe problem.
phenomenon may be related to the positive semi-definite of $B$, in other words, the descending direction $d$ of conjugate gradient (CG) method becomes eigenvector of 0 eigenvalue of $B$ or $Bd \approx 0$.

5.2 Headphantom problem

In computerized tomography, the distribution of some physical parameter(such as absorption intensities) at the cross-section of the object need to be reconstructed from the projection data such as medical diagnosis—the distribution of the absorption intensities of tissue slice need to be reconstructed from X-ray data. The computerized tomography system attributes to a linear system $Ax = b$, where $A$ is a projected system, $b$ is scanning data, $x$ is unknown intensity image of an object. In the general case, the system is overdetermined.

The linear system is generated from the subroutine ‘parallel’ in ARTool package [16], and there are 36 projective angles at equal intervals in $[0, 2\pi]$ and 75 equi-spaced parallel rays per angle. The headphantom is discretized into $50 \times 50$ pixels, and the dimension of $A$ is $2700 \times 2500$.

The initial value is taken as $x_0 = 0 \in \mathbb{R}^{2500}$, and numerical results are shown in Figure 5.3, where (a) and (c) are results of the Kaczmarz-Tanabe method, symmetric Kaczmarz-Tanabe method and two iterations Kaczmarz-Tanabe method for solving the Headphantom problem, (b) and (d) are results of the Kaczmarz-Tanabe method, symmetric Kaczmarz-Tanabe method, SIRT type methods and CGMN method for solving the problem. For this problem, the CGMN method seems to be better than other methods and there is no phenomenon in the Tanabe problem.

From Figure 5.3, we can see that the Kaczmarz-Tanabe and symmetric Kaczmarz-Tanabe methods are significantly better than the SIRT methods, and slight worse than CGMN method. Numerical images of these methods are shown in Figure 5.4. From the visual effect, the Kaczmarz-Tanabe method, symmetric Kaczmarz-Tanabe method and CGMN method are close and better than the SIRT type methods.

![Figure 5.3: The comparison of errors $\|y_k - x^*\|_2$, $\|y_k - x^1\|_2$, where (a) and (c) are comparisons among the Kaczmarz-Tanabe method, the symmetric Kaczmarz-Tanabe method and two iterations Kaczmarz-Tanabe method for solving Headphantom problem, and (b) and (d) are comparisons among the Kaczmarz-Tanabe method, the symmetric Kaczmarz-Tanabe method and SIRT type methods for solving Headphantom problem.](image-url)
6 Conclusion

The Kaczmarz-Tanabe method is the further research of the Kaczmarz method. Due to the row to row iterative characteristic of the Kaczmarz method, Kaczmarz’s iteration generally converges slowly and has volatility for perturbed linear systems. The Kaczmarz-Tanabe method overcomes the volatility of Kaczmarz’s method and can smoothly approach the ‘pseudo-inverse solution’ when solving the perturbation problem, which lays a foundation for us to further study the minimum norm least squares solution.

In addition, as a comparison, we also consider the more popular symmetric Kaczmarz-Tanabe method and derive its standard form. We should pay attention to the symmetric Kaczmarz-Tanabe method because one iteration of the symmetric Kaczmarz-Tanabe method can obtain the effect of two iterations of the Kaczmarz-Tanabe method. Kaczmarz-Tanabe’s iteration and symmetric Kaczmarz-Tanabe’s iteration have the same iterative formula, if $C$ and $\bar{C}$ are known, then the symmetric Kaczmarz-Tanabe method has obvious advantages over the Kaczmarz-Tanabe method in computational efficiency.

Numerical tests also show that the Kaczmarz-Tanabe type methods, i.e., the Kaczmarz-Tanabe method and the symmetric Kaczmarz-Tanabe method in this paper, are better than the SIRT methods. Although Kaczmarz-Tanabe type methods can not achieve the convergence effect of the CGMN method in some cases, They have advantages in problem applicability, i.e., they converge stably to the minimum norm least-squares solution for all compatible linear systems when the initial guess $x_0 \in R(A^T)$. In particular, after obtaining $C$
and $\tilde{C}$, Kaczmarz-Tanabe’s iteration and symmetric Kaczmarz-Tanabe’s iteration can be implemented as easily as the SIRT methods. In practical applications, such as medical image reconstruction and so on, $C$ and $\tilde{C}$ can be calculated in advance and stored in the device, which enables us implement these iterative methods quickly and get a better solution.

References

[1] R. S. Ledley and W. R. Ayers. Computerized medical imaging and graphics evolves from computerized tomography. *Computerized Medical Imaging & Graphics the Official Journal of the Computerized Medical Imaging Society*, 12(1):v–xviii, 1988.

[2] G. T. Herman. *Fundamentals of computerized tomography*. Academic Press, 2010.

[3] F. Natterer. *The Mathematics of Computerized Tomography*. Society for Industrial and Applied Mathematics, 2001.

[4] J. Radon. Uber die bestimmung von funktionen durch ihre integralwerte langs gewisser mannigfaltigkeiten. *Ber. Verh. Sächs. Akad. Wiss. Leipzig*, 69:262–267, 1917.

[5] Gabor T. Herman. Image reconstruction from projections. *Real-Time Imaging*, 1:3–8, 1995.

[6] R. Gordon, R. Bender, and G. T. Herman. Algebraic reconstruction techniques (ART) for three dimensional electron microscopy and X-ray photography. *Journal of Theoretical Biology*, 29:471–481, 1970.

[7] H. W. Engl, M. Hanke, and A. Neubauer. *Regularization of inverse problems*. Kluwer Academic, 1996.

[8] Fang Wang, Weiguo Li, Wendi Bao, and Zhonglu Lv. Gauss-Seidel method with oblique direction. *Results in Applied Mathematics*, 12:100180, 2021.

[9] S. Kaczmarz. Angenäherte auflösung von systemen linearer gleichungen. *Bulletin de Academie Polonaise des Sciences et Lettres*, 35:355–357, 1937.

[10] Å. Björck and T. Elfving. Accelerate projection methods for computing pseudoinverse solutions of systems of linear equations. *BIT Numerical Mathematics*, 19(2):145–163, 1979.

[11] W. Huang. The convergence of the multigrid method using the symmetric Kaczmarz iteration as its smoothing method. *Acta Mathematicae Applicatae Sinica*, 16:100–106, 1993.

[12] K. Wei. Solving systems of phaseless equations via Kaczmarz methods: a proof of concept study. *Inverse Problems*, 31:125008, 2015.

[13] C. G. Kang. Convergence rates of the Kaczmarz-Tanabe method for linear system. *Journal of Computational & Applied Mathematics*, 394:113577, 2021.

[14] C. Popa. Convergence rates for Kaczmarz-type algorithms. *Numerical Algorithms*, 79:1–17, 2018.

[15] K. Tanabe. Projection method for solving a singular system of linear equations and its applications. *Numerische Mathematik*, 17:203–214, 1971.

[16] P. C. Hansen and J. S. Jorgensen. AIR Tools II: algebraic iterative reconstruction method, improved implementation. *Numerical Algorithms*, 79:107–137, 2018.

[17] L. Landweber. An iteration formula for Fredholm integral equations of the first kind. *American Journal of Mathematics*, 73:615–624, 1951.
[18] G. Cimmino. Calcolo approssimato per le soluzioni dei sistemi di equazioni lineari. *La Ricerca Scientifica (Roma)*, pages 326–333, 1938.

[19] Y. Censor, T. Elfving, G. T. Herman, and T. Nikazad. On diagonally relaxed orthogonal projection methods. *SIAM Journal on Scientific Computing*, 30(1):473–504, 2007.

[20] Y. Censor, G. Dan, and R. Gordon. Component averaging: an efficient iterative parallel algorithm for large and sparse unstructured problems. *Parallel Computing*, 27:777–808, 2001.

[21] A. van der Sluis and H. A. van der Vorst. SIRT- and CG-type methods for the iterative solution of sparse linear least-squares problems. *Linear Algebra & Its Applications*, 130:257–303, 1990.

[22] M. Jiang and G. Wang. Convergence of the simultaneous algebraic reconstruction technique (SART). *IEEE Transactions on Image Processing A Publication of the IEEE Signal Processing Society*, 12(8):957–61, 2003.

[23] A. H. Andersen and A. C. Kak. Simultaneous algebraic reconstruction technique (SART): A superior implementation of the ART algorithm. *Ultrasonic Imaging*, 6(1):81–94, 1984.

[24] X. Wan, F. Zhang, Q. Chu, K. Zhang, S. Fei, B. Yuan, and Z. Liu. Three-dimensional reconstruction using an adaptive simultaneous algebraic reconstruction technique in electron tomography. *Journal of Structural Biology*, 175(3):277–287, 2011.

[25] C. Lay David. *Linear algebra and its applications*. Pearson Education, Inc, -4th edition, 2012.

[26] D. Needell and J. A. Tropp. Paved with good intentions: analysis of a randomized block Kaczmarz method. *Linear Algebra & Its Applications*, 441:199–221, 2014.

[27] A. Ma, D. Needell, and A. Ramdas. Convergence properties of the randomized extended Gauss-Seidel and Kaczmarz methods. *SIAM Journal on Matrix Analysis & Applications*, 36(4):1590–1604, 2015.

[28] T. Elfving. Block-iterative methods for consistent and inconsistent linear equations. *Numerische Mathematik*, 35(1):1–12, 1980.

[29] Y. Censor and T. Elfving. Block-iterative algorithms with diagonally scaled oblique projections for the linear feasibility problem. *SIAM Journal on Matrix Analysis & Applications*, 24:40–58, 2002.

[30] D. Needell, R. Zhao, and A. Zouzias. Randomized block Kaczmarz method with projection for solving least squares. *Linear Algebra & Its Applications*, 484:322–343, 2015.