On fast greedy block Kaczmarz methods for solving large consistent linear systems

A.-Qin Xiao · Jun-Feng Yin · Ning Zheng

Received: 30 October 2022 / Revised: 22 December 2022 / Accepted: 31 January 2023 / Published online: 18 March 2023 © The Author(s) under exclusive licence to Sociedade Brasileira de Matemática Aplicada e Computacional 2023

Abstract
A fast greedy block Kaczmarz method combined with general greedy strategy and average technique are proposed for solving large consistent linear systems. Theoretical analysis of the convergence of the proposed method is given in detail. Numerical experiments show that the proposed methods are efficient and faster than the existing methods.

Keywords Linear systems · Kaczmarz method · Modified greedy strategies · Average block · Convergence property

Mathematics Subject Classification 65F10 · 65F20 · 15A06

1 Introduction

Consider the solution of consistent linear algebraic equations

$$Ax = b,$$

where $A \in \mathbb{R}^{m \times n}$ and $b \in \mathbb{R}^n$, one of the classical and popular iteration methods is the Kaczmarz method (Kaczmarz 1937). Due to its simplicity and efficiency, it was deeply studied and widely used in many practical scientific and engineering applications, for instance, computer tomography (CT) (Kak and Slaney 2001), image reconstruction (Herman and Davidi 2008), machine learning (Needell et al. 2016) and option pricing (Filipović et al. 2019).

Denote $A_i$ by the $i$th row of $A$ and $b_i$ by the $i$th entry of $b$, respectively. Given an initial vector $x_0 \in \mathbb{R}^n$, the classical Kaczmarz method iterates by

Communicated by Xiang Wang.
\[ x_{k+1} = x_k + \frac{b_{i_k} - A_{i_k} x_k}{\| A_{i_k} \|^2_2} (A_{i_k})^T, \quad k = 0, 1, 2, \ldots, \]

where the row index \( i_k \) is cyclically selected from \([m] = \{1, 2, \ldots, m\}\). To accelerate the convergence of the Kaczmarz method, Strohmer and Vershynin (2009) presented a randomized Kaczmarz method by selecting the row index \( i_k \) with probability proportional to \( \| A_{i_k} \|^2_2 \) and proved its linear convergence rate in expectation. Bai and Wu (2018a) constructed a greedy randomized Kaczmarz method to accelerate the convergence performance. For more variants of the randomized and greedy Kaczmarz methods, we refer the reader to Bai and Wu (2018b, 2019) and Yin et al. (2022). Moreover, the Kaczmarz-type methods can be used as preconditioners for Krylov subspace methods, such as GMRES (Saad 2003; Du et al. 2021) method and QGMRES (Jia and Ng 2021) method.

The idea of block Kaczmarz method can date back to the work of Elfving (1980), which used many equations simultaneously at each iteration. The block Kaczmarz method can be described as

\[ x_{k+1} = x_k + A^\dagger_{I_k} (b_{I_k} - A_{I_k} x_k), \quad k = 0, 1, 2, \ldots, \] (1.2)

where \( A^\dagger_{I_k} \) represents the Moore-Penrose general inverse of the chosen submatrix \( A_{I_k} \) and \( I_k \subset [m] \). In Needell and Tropp (2014) the authors presented a randomized block Kaczmarz method, where \( I_k \) is selected uniformly at random from \([m] \). Further, a greedy block Kaczmarz method (Niu and Zheng 2020) was presented, which adaptively chooses the block row indices without predetermining a partition of the rows of \( A \).

However, each iteration of the block Kaczmarz method requires the computation of the pseudoinverse of the selected submatrix corresponding to the residual subvector and this usually incurs a high cost. Necoara (2019) established a unified framework of randomized average block Kaczmarz methods by taking a convex combination of several updateings as a new direction and implemented on the distributed computing units. Miao (2022) proposed an average block variant of the greedy randomized Kaczmarz method (Bai and Wu 2018a) and studied its convergence.

In this work, we construct a fast greedy block Kaczmarz method with average technique to avoid computing the pseudoinverse of submatrices of the coefficients matrix, where a modified greedy strategy utilizing the general norm of residual vectors is proposed and well studied, which can choose the working rows based on this greedy criterion dynamically and flexibly. Theoretical analysis of the convergence of the presented method is given in detail. The results of numerical experiments further demonstrate that the presented methods are efficient and faster than the existing approaches.

The outline of this paper is as follows. In Sect. 2, a fast greedy block Kaczmarz method is presented with a modified greedy row selection strategy and the convergence theory of the proposed method is established. Numerical experiments are carried out in Sect. 3 to display the efficiency of the new method. Finally, we draw the conclusions in Sect. 4.

### 2 The fast greedy block Kaczmarz method

In this section, after reviewing the fast deterministic block Kaczmarz method, we present a fast greedy block Kaczmarz method for solving large consistent linear systems by using a modified greedy row selection strategy and the averaging technique.
Let \( \zeta_k \) be a linear combination of unit column vectors \( e_i \in \mathbb{R}^m \) \( (i \in J_k) \) and its coefficients are the corresponding entries of the residual vector, that is,

\[
\zeta_k = \sum_{i \in J_k} (b_i - A_i x_k) e_i, \quad J_k \subset [m], \quad k = 0, 1, 2, \ldots.
\]

Similar to the iteration of average block Kaczmarz methods (Chen and Huang 2022), the stepsize and weight are set to be \( \alpha_k = \frac{\|\zeta_k\|_2}{\|A_{J_k}\|_F^2} \) and \( \omega^k_i = \frac{\|A_i\|_2^2}{\|A_{J_k}\|_F^2} \), and the fast deterministic block Kaczmarz method can iterate as follows

\[
x_{k+1} = x_k + \alpha_k \left( \sum_{i \in J_k} \omega^k_i \frac{b_i - A_i x_k}{\|A_i\|_2^2} (A_i)^T \right)
\]

\[
= x_k + \frac{\|\zeta_k\|_2^2 \|A_{J_k}\|_F^2}{\|A^T \zeta_k\|_2^2} \left( \sum_{i \in J_k} \frac{\|A_i\|_2^2}{\|A_{J_k}\|_F^2} \cdot \frac{b_i - A_i x_k}{\|A_i\|_2^2} (A_i)^T \right)
\]

\[
= x_k + \frac{\zeta_k^T (b - A x_k)}{\|A^T \zeta_k\|_2^2} \left( \sum_{i \in J_k} (b_i - A_i x_k) (A_i)^T \right)
\]

\[
= x_k + \frac{\zeta_k^T (b - A x_k)}{\|A^T \zeta_k\|_2^2} A^T \zeta_k,
\]

where the block indices \( J_k \) is chosen by

\[
J_k = \{ i | |b_i - A_i x_k|^p \geq \gamma_k \|b - A x_k\|_2^p \|A_i\|_2^p \}
\]

with

\[
\gamma_k = \frac{1}{2} \left( \frac{1}{\|b - A x_k\|_2^p} \max_{1 \leq i \leq m} \left\{ \frac{|b_i - A_i x_k|^p}{\|A_i\|_2^p} \right\} + \frac{1}{\|A\|_F^p} \right).
\]

One drawback of the fast deterministic block Kaczmarz method is the greedy strategy for choosing the working block indices. The size of the block \( J_k \) may be small if the Frobenius norm of \( A \) is very small, which may lead to a very slow convergence.

To further accelerate the fast deterministic block Kaczmarz method, we propose to determine the working rows by a modified greedy strategy that utilizes the general norm of residual vectors. Let \( \eta \in (0, 1] \) and \( p \in [1, +\infty) \), the control index subset is defined as

\[
\tau_k = \{ i | |b_i - A_i x_k|^p \geq \epsilon_k \|A_i\|^p_p \}, \quad k = 0, 1, 2, \ldots,
\]

where

\[
\epsilon_k = \eta \cdot \max_{1 \leq i \leq m} \left\{ \frac{|b_i - A_i x_k|^p}{\|A_i\|_p^p} \right\}.
\]

It follows that

\[
\epsilon_k = \eta \cdot \max_{1 \leq i \leq m} \left\{ \frac{|b_i - A_i x_k|^p}{\|A_i\|_p^p} \right\} \leq \max_{1 \leq i \leq m} \left\{ \frac{|b_i - A_i x_k|^p}{\|A_i\|_p^p} \right\},
\]
which implies that there is at least one index \( j \in [m] \) such that

\[
\frac{|b_j - A_j x_k|^p}{\|A_j\|_p^p} = \max_{1 \leq i \leq m} \left\{ \frac{|b_i - A_i x_k|^p}{\|A_i\|_p^p} \right\},
\]

then \( j \in \tau_k \), i.e. \( \tau_k \) is not empty.

Given an initial vector \( x_0 \), the fast greedy block Kaczmarz method can be described in Algorithm 1.

**Algorithm 1** The fast greedy block Kaczmarz method (FGBK)

**Require:** \( A, b, x_0, l, \eta \in (0, 1) \) and \( p \in [1, +\infty) \)

**Ensure:** \( x_l \)

1: for \( k = 0, 1, 2, \ldots, l - 1 \) do
2: Compute \( \epsilon_k = \eta \cdot \max_{1 \leq i \leq m} \left\{ \frac{|b_i - A_i x_k|^p}{\|A_i\|_p^p} \right\} \).
3: Determine the control index set of positive integers
\[
\tau_k = \left\{ i \mid |b_i - A_i x_k|^p \geq \epsilon_k \|A_i\|_p^p \right\}. \quad (2.2)
\]
4: Compute \( \xi_k = \sum_{i \in \tau_k} (b_i - A_i x_k) e_i \).
5: Set \( x_{k+1} = x_k + \frac{\xi_k^T (b - Ax_k)}{\|A^T \xi_k\|_2^2 A^T \xi_k} \).
6: end for

Moreover, the theoretical analysis for the convergence performance of the fast greedy block Kaczmarz method are established as follows.

**Theorem 2.1** For any initial vector \( x_0 \), the iteration sequence \( \{x_k\}_{k=0}^\infty \) obtained by the fast greedy block Kaczmarz method converges to the least norm solution \( x_* = A^\dagger b \). In addition, the norm of the approximate solution error satisfies

\[
\|x_{k+1} - x_*\|_2^2 \leq \left( 1 - \beta_k(\eta, p) \sigma_{\min}^2(A) \right) \|x_k - x_*\|_2^2, \quad k \geq 0,
\]

where \( \beta_k(\eta, p) = \frac{\eta^2}{\sum_{i \in [m] \setminus \tau_k} \|A_i\|_p^2}, \quad \sum_{i \in \tau_k} \|A_i\|_p^2 \leq \sigma_{\min}(A \tau_k) \cdot \eta \in (0, 1) \) and \( p \in [1, +\infty) \).
Proof From the iterate scheme (2.4), it holds that

\[
x_{k+1} - x_* = x_k - x_* + \frac{\xi_k^T (b - Ax_k)}{\| A^T \xi_k \|_2^2} A^T \xi_k
\]

\[
= x_k - x_* - \frac{\xi_k^T A (x_k - x_*)}{\| A^T \xi_k \|_2^2} A^T \xi_k
\]

\[
= x_k - x_* - \frac{A^T \xi_k \xi_k^T A}{\| A^T \xi_k \|_2^2} (x_k - x_*).
\]

By the Pythagorean theorem, it follows that

\[
\| x_{k+1} - x_* \|_2^2 = \| x_k - x_* \|_2^2 - \| \xi_k \|_2^2 = \sum_{i \in \tau_k} | b_i - A_i x_k |^2.
\]

(2.5)

Denote \( E_k \in \mathbb{R}^{m \times |\tau_k|} \) be a matrix whose columns are consisted of all the unit vector \( e_i \) with \( i \in \tau_k \), \( A_{\tau_k} = E_k^T A \), \( \hat{\xi}_k = E_k^T \xi_k \), then

\[
\| \hat{\xi}_k \|_2^2 = \xi_k^T E_k E_k^T \xi_k = \| \xi_k \|_2^2 = \sum_{i \in \tau_k} | b_i - A_i x_k |^2,
\]

(2.6)

and

\[
\| A^T \hat{\xi}_k \|_2^2 = \xi_k^T A A^T \hat{\xi}_k = \hat{\xi}_k^T E_k^T A A^T E_k \hat{\xi}_k = \xi_k^T A_{\tau_k} A_{\tau_k}^T \hat{\xi}_k = \| A_{\tau_k} \hat{\xi}_k \|_2^2.
\]

(2.7)

Therefore,

\[
\| A_{\tau_k}^T \hat{\xi}_k \|_2^2 = \xi_k^T A_{\tau_k} A_{\tau_k}^T \hat{\xi}_k \leq \sigma_{\text{max}}^2 (A_{\tau_k}) \| \hat{\xi}_k \|_2^2,
\]

(2.8)

where \( \sigma_{\text{max}} (A_{\tau_k}) \) represents the largest singular value of selected submatrix \( A_{\tau_k} \). By the definition of \( \xi_k \) in (2.3) and (2.6), it holds that

\[
\xi_k^T (b - Ax_k) = \left( \sum_{i \in \tau_k} (b_i - A_i x_k) e_i^T \right)(b - Ax_k)
\]

\[
= \sum_{i \in \tau_k} \left( (b_i - A_i x_k) e_i^T (b - Ax_k) \right)
\]

\[
= \sum_{i \in \tau_k} | b_i - A_i x_k |^2
\]

\[
= \| \hat{\xi}_k \|_2^2.
\]

(2.9)

Since both \( x_k \) and \( x_k - x_* \) in the column space of \( A^T \), then

\[
\| b - Ax_k \|_2^2 = \| A (x_k - x_*) \|_2^2 \geq \sigma_{\text{min}}^2 (A) \| x_k - x_* \|_2^2.
\]

(2.10)
From Eqs. (2.7)–(2.10) and the definition of $\tau_k$ in (2.2), it follows that

\[
\frac{\xi_k^T (b - A x_k)}{\| A^T \xi_k \|_2^2} = \frac{\left( \sum_{i \in \tau_k} |b_i - A_i x_k|^2 \right) \| \xi_k \|_2^2}{\| A^T \xi_k \|_2^2} \geq \frac{\sum_{i \in \tau_k} |b_i - A_i x_k|^2}{\sigma_{\max}^2 (A_{\tau_k})} \geq \frac{\sum_{i \in \tau_k} (|b_i - A_i x_k|^p)^{\frac{2}{p}}}{\sigma_{\max}^2 (A_{\tau_k})} \geq (\epsilon_k) \cdot \frac{\sum_{i \in \tau_k} \| A_i \|_p^2}{\sigma_{\max}^2 (A_{\tau_k})} \geq (\epsilon_k) \cdot \frac{\sum_{i \in \tau_k} \| A_i \|_p^2}{\sigma_{\max}^2 (A_{\tau_k})}.
\]

(2.11)

In addition, it is seen that

\[
b - A x_k = b - A \left( x_{k-1} + \frac{\xi_{k-1}^T (b_{\tau_k-1} - A_{\tau_k-1} x_{k-1})}{\| A^T \xi_{k-1} \|_2^2} \right) \geq (b - A x_{k-1}) - \frac{A \xi_{k-1}^T (b_{\tau_k-1} - A_{\tau_k-1} x_{k-1})}{\| A^T \xi_{k-1} \|_2^2} \leq 0.
\]

Therefore,

\[
b_{\tau_k-1} - A_{\tau_k-1} x_k = (b_{\tau_k-1} - A_{\tau_k-1} x_{k-1}) - \frac{A \xi_{k-1}^T (b_{\tau_k-1} - A_{\tau_k-1} x_{k-1})}{\| A^T \xi_{k-1} \|_2^2} (b_{\tau_k-1} - A_{\tau_k-1} x_{k-1}) = 0.
\]

It is known that

\[
\| b - A x_k \|_2^2 \leq \sum_{i \in [m] \setminus \tau_k} \frac{|b_i - A_i x_k|^2}{\| A_i \|_p^2} \| A_i \|_p^2 \leq \sum_{i \in [m] \setminus \tau_k} \left( \frac{|b_i - A_i x_k|^p}{\| A_i \|_p^2} \right) \| A_i \|_p^2 \leq \left( \max_{1 \leq i \leq m} \left( \frac{|b_i - A_i x_k|^p}{\| A_i \|_p^2} \right) \right)^{\frac{2}{p}} \sum_{i \in [m] \setminus \tau_k} \| A_i \|_p^2.
\]
It follows that
\[
(\epsilon_k)^{\frac{2}{p}} = \left( \eta \cdot \max_{1 \leq i \leq m} \left\{ \frac{|b_i - A_i x_k|^p}{\|A_i\|_p^p} \right\} \right)^{\frac{2}{p}} \geq \eta \frac{2}{p} \cdot \frac{\|b - A x_k\|_2^2}{\sum_{i \in [m] \setminus \bar{\tau}_{k-1}} \|A_i\|_p^2} \geq \eta \frac{2}{p} \cdot \sigma_{\min}^2(A) \|x_k - x_\ast\|_2^2.
\] (2.12)

From (2.11) and (2.12), it deduces that
\[
\frac{\left| \xi_k^T (b - A x_k) \right|^2}{\|A^T \xi_k\|_2^2} \geq \beta_k(\eta, p) \cdot \sigma_{\min}^2(A) \|x_k - x_\ast\|_2^2,
\] (2.13)
where \( \beta_k(\eta, p) = \sum_{i \in [m] \setminus \bar{\tau}_{k-1}} \|A_i\|_p^2 \cdot \frac{\sigma_{\max}^2(A_{\bar{\tau}_k})}{\sigma_{\max}^2(A_{\bar{\tau}_k})}, \eta \in (0, 1) \) and \( p \in [1, +\infty) \).

Finally, by combining (2.5) and (2.13), it follows that
\[
\|x_{k+1} - x_\ast\|_2^2 \leq \left( 1 - \beta_k(\eta, p) \sigma_{\min}^2(A) \right) \|x_k - x_\ast\|_2^2.
\]

Note that the upper bound of convergence rate of the fast greedy block Kaczmarz method is related to the relaxation parameter \( \eta \), the parameter \( p \), the geometric properties of the coefficient matrix \( A \) and its row submatrices at each iteration. However, the practical convergence speed of the fast greedy block Kaczmarz methods could be faster than the upper bound.

### 3 Numerical experiments

In this section, a number of numerical experiments are presented to illustrate the efficiency of the fast greedy block Kaczmarz (FGBK) method, compared with the greedy block Kaczmarz (GBK) method (Niu and Zheng 2020) and the fast deterministic block Kaczmarz (FDBK) method (Chen and Huang 2022) in aspects of the number of iteration steps (denoted as ‘IT’) and the elapsed computing time in seconds (denoted as ‘CPU’).

In the numerical experiment, the solution vector \( x \) is firstly constructed and \( b = A x \) so that the linear system is consistent. All the iterations are started from the initial vector \( x_0 = 0 \), and terminated when the relative solution error (denoted as ‘RSE’)

\[
\text{RSE} = \frac{\|x_k - x_\ast\|_2^2}{\|x_0 - x_\ast\|_2^2} < 10^{-6},
\]

or \( \text{IT} \) exceeds a maximal number, e.g., 100,000. For the greedy block Kaczmarz method, the control index set is determined by

\[
\bar{\tau}_k = \left\{ i \left| \left| b_i - A_i x_k \right|^2 \geq \delta_k \max_{1 \leq i \leq m} \left\{ \frac{|b_i - A_i x_k|^2}{\|A_i\|_2^2} \right\} \|A_i\|_2^2 \right\}
\]

with parameter \( \delta_k \) is

\[
\delta_k = \frac{1}{2} + \frac{1}{2} \frac{\|b - A x_k\|_2^2}{\|A\|_F^2} \left( \max_{1 \leq i \leq m} \left\{ \frac{|b_i - A_i x_k|^2}{\|A_i\|_2^2} \right\} \right)^{-1}
\]
Table 1 Numerical results for overdetermined random matrices

| Method  | $m \times n$   | 10,000 $\times$ 5000 | 12,000 $\times$ 5000 | 14,000 $\times$ 5000 | 16,000 $\times$ 5000 | 18,000 $\times$ 5000 |
|---------|----------------|-----------------------|----------------------|----------------------|----------------------|----------------------|
| GBK     | IT            | 471                   | 294                  | 229                  | 182                  | 144                  |
|         | CPU           | 19.2449               | 13.8489              | 12.0386              | 10.7724              | 9.6516               |
|         | RSE           | $9.96 \times 10^{-7}$  | $9.64 \times 10^{-7}$ | $9.71 \times 10^{-7}$ | $9.47 \times 10^{-7}$ | $9.59 \times 10^{-7}$ |
| FDBK    | IT            | 494                   | 300                  | 235                  | 189                  | 153                  |
|         | CPU           | 24.3282               | 17.9247              | 16.2996              | 15.2890              | 13.3954              |
|         | RSE           | $9.94 \times 10^{-7}$  | $9.94 \times 10^{-7}$ | $9.98 \times 10^{-7}$ | $9.51 \times 10^{-7}$ | $9.67 \times 10^{-7}$ |
| FGBK ($p = 1$) | $\eta_{exp}$ | 0.05                 | 0.05                 | 0.05                 | 0.05                 | 0.05                 |
|         | IT            | 72                    | 46                   | 36                   | 29                   | 25                   |
|         | CPU           | 8.1323                | 6.8570               | 6.7586               | 6.8286               | 7.1922               |
|         | RSE           | $9.14 \times 10^{-7}$  | $9.91 \times 10^{-7}$ | $9.75 \times 10^{-7}$ | $9.82 \times 10^{-7}$ | $7.02 \times 10^{-7}$ |
| FGBK ($p = 2$) | $\eta_{exp}$ | 0.05                 | 0.05                 | 0.05                 | 0.05                 | 0.05                 |
|         | IT            | 71                    | 47                   | 37                   | 30                   | 25                   |
|         | CPU           | 5.0230                | 4.4264               | 4.3369               | 4.2468               | 4.1622               |
|         | RSE           | $9.51 \times 10^{-7}$  | $9.13 \times 10^{-7}$ | $7.55 \times 10^{-7}$ | $7.92 \times 10^{-7}$ | $7.93 \times 10^{-7}$ |
| FGBK ($p = 3$) | $\eta_{exp}$ | 0.05                 | 0.05                 | 0.05                 | 0.05                 | 0.05                 |
|         | IT            | 78                    | 50                   | 39                   | 32                   | 27                   |
|         | CPU           | 6.2096                | 5.8312               | 6.0091               | 6.3810               | 6.6951               |
|         | RSE           | $8.86 \times 10^{-7}$  | $9.15 \times 10^{-7}$ | $8.68 \times 10^{-7}$ | $7.90 \times 10^{-7}$ | $9.35 \times 10^{-7}$ |

while the relaxation parameter $\eta_{exp}$ in the fast greedy block Kaczmarz method is experimentally selected by minimizing the numbers of total iterations.

In the first example, we test some synthetic matrices $A$ with $m > n$, the matrix $A$ is generated by using the MATLAB function ‘randn’, where the size of the matrices is chosen to be 10,000 $\times$ 5000, 12,000 $\times$ 5000, 14,000 $\times$ 5000, 16,000 $\times$ 5000 and 18,000 $\times$ 5000, respectively.

In Table 1, the number of iterations and CPU time of the greedy block Kaczmarz, the fast deterministic block Kaczmarz and fast greedy block Kaczmarz methods with $p = 1, 2$ and 3 are reported, respectively.

From Table 1, it can be observed that GBK, FDBK, FGBK ($p = 1$), FGBK ($p = 2$) and FGBK ($p = 3$) methods converge successfully and the FGBK-type methods outperform the other two methods in aspects of both the iteration steps and CPU time. Moreover, the fast greedy block Kaczmarz method with $p = 1$ requires the least number of iterations for all matrices except the size of the matrix is chosen to be 10,000 $\times$ 5000. The fast greedy block Kaczmarz method with $p = 2$ has the least computing time for all matrices. It shows that the efficiency of the modified greedy row selection strategy and indicates that a small value of $p$ may further accelerate the convergence.

In the second example, we test some synthetic matrices $A$ with $m < n$, the matrix $A$ is generated by using the MATLAB function ‘randn’, where the size of the matrices is chosen to be 5000 $\times$ 10,000, 5000 $\times$ 12,000, 5000 $\times$ 14,000, 5000 $\times$ 16,000 and 5000 $\times$ 18,000, respectively. In Table 2, the number of iterations and CPU time of the greedy block Kaczmarz, the fast deterministic block Kaczmarz and fast greedy block Kaczmarz methods with $p = 1, 2$ and 3 are reported respectively.
Table 2  Numerical results for underdetermined random matrices

| Method      | $m \times n$   | 5000 $\times$ 10,000 | 5000 $\times$ 12,000 | 5000 $\times$ 14,000 | 5000 $\times$ 16,000 | 5000 $\times$ 18,000 |
|-------------|-----------------|-----------------------|-----------------------|-----------------------|-----------------------|-----------------------|
| GBK         | IT              | 543                   | 349                   | 251                   | 208                   | 169                   |
|             | CPU             | 40.3710               | 36.0279               | 33.5738               | 35.6595               | 35.6040               |
|             | RSE             | $9.97 \times 10^{-7}$ | $9.92 \times 10^{-7}$ | $9.89 \times 10^{-7}$ | $9.82 \times 10^{-7}$ | $9.76 \times 10^{-7}$ |
| FDBK        | IT              | 559                   | 356                   | 256                   | 209                   | 170                   |
|             | CPU             | 27.8735               | 21.4744               | 17.9981               | 16.5299               | 15.4575               |
|             | RSE             | $9.91 \times 10^{-7}$ | $9.85 \times 10^{-7}$ | $9.71 \times 10^{-7}$ | $9.54 \times 10^{-7}$ | $9.97 \times 10^{-7}$ |
| FGBK ($p = 1$) | $\eta_{exp}$   | 0.10                  | 0.05                  | 0.05                  | 0.05                  | 0.05                  |
|             | IT              | 73                    | 47                    | 35                    | 29                    | 24                    |
|             | CPU             | 4.5057                | 3.6111                | 3.1248                | 2.9545                | 2.7597                |
|             | RSE             | $9.53 \times 10^{-7}$ | $9.52 \times 10^{-7}$ | $9.29 \times 10^{-7}$ | $9.70 \times 10^{-7}$ | $8.80 \times 10^{-7}$ |
| FGBK ($p = 2$) | $\eta_{exp}$   | 0.05                  | 0.05                  | 0.05                  | 0.05                  | 0.05                  |
|             | IT              | 74                    | 48                    | 36                    | 30                    | 25                    |
|             | CPU             | 4.0051                | 3.1378                | 2.8106                | 2.6807                | 2.4948                |
|             | RSE             | $8.77 \times 10^{-7}$ | $8.03 \times 10^{-7}$ | $8.02 \times 10^{-7}$ | $7.92 \times 10^{-7}$ | $7.84 \times 10^{-7}$ |
| FGBK ($p = 3$) | $\eta_{exp}$   | 0.05                  | 0.05                  | 0.05                  | 0.05                  | 0.05                  |
|             | IT              | 82                    | 55                    | 42                    | 35                    | 30                    |
|             | CPU             | 6.0167                | 5.6553                | 5.5993                | 6.0705                | 6.3334                |
|             | RSE             | $9.36 \times 10^{-7}$ | $9.38 \times 10^{-7}$ | $8.38 \times 10^{-7}$ | $8.77 \times 10^{-7}$ | $9.37 \times 10^{-7}$ |

Table 3  Information of the matrices from SuiteSparse Matrix Collection

| Name       | stat96v5       | crew1       | bibd_17_8     | bibd_16_8     |
|------------|----------------|-------------|---------------|---------------|
| $m \times n$ | 2307 $\times$ 75,779 | 135 $\times$ 6469 | 136 $\times$ 24,310 | 120 $\times$ 12,870 |
| Density    | 0.13%          | 5.38%       | 20.59%        | 23.33%        |
| Cond($A$)  | 19.52          | 18.20       | 9.04          | 9.54          |

From Table 2, it can be observed that GBK, FDBK, FGBK ($p = 1$), FGBK ($p = 2$) and FGBK ($p = 3$) methods converge successfully and the FGBK-type methods outperform the other two methods in aspects of both the iteration steps and CPU time. Moreover, the fast greedy block Kaczmarz method with $p = 1$ requires the least number of iterations. It shows that the efficiency of the modified greedy row selection strategy and indicates that a small value of $p$ may further accelerate the convergence.

In the third example, the matrices are taken from the SuiteSparse Matrix Collection (Davis and Hu 2011) to further compare the convergence performances of these Kaczmarz methods. The test matrices ‘stat96v5’ and ‘crew1’ come from linear programming problems while ‘bibd_17_8’ and ‘bibd_16_8’ come from combinatorial problems. In Table 3, the sizes ($m \times n$), density and condition number of the test matrices are listed, respectively.

In Table 4, the number of iterations and CPU time of the greedy block Kaczmarz method, the fast deterministic block Kaczmarz method and the fast greedy block Kaczmarz method with $p = 1$, 2 and 3 are reported respectively.

From Table 4, it is observed that the proposed methods require fewer iterations and less CPU time than the other two block Kaczmarz methods. For the matrix ‘stat96v5’, the proposed
Table 4 Numerical results for the matrices from SuiteSparse Matrix Collection

| Method | stat96v5 | crew1 | bibd_17_8 | bibd_16_8 |
|--------|----------|-------|-----------|-----------|
| GBK    | IT       | 80    | 547       | 237       | 280       |
|        | CPU      | 1.258 | 0.7068    | 19.7138   | 11.3608   |
|        | RSE      | $8.69 \times 10^{-7}$ | $9.85 \times 10^{-7}$ | $9.75 \times 10^{-7}$ | $9.83 \times 10^{-7}$ |
| FDBK   | IT       | 249   | 815       | 256       | 289       |
|        | CPU      | 0.2940| 0.1438    | 0.3103    | 0.1846    |
|        | RSE      | $9.22 \times 10^{-7}$ | $9.89 \times 10^{-7}$ | $9.66 \times 10^{-7}$ | $9.91 \times 10^{-7}$ |
| FGBK ($p = 1$) | $\eta_{exp}$ | 0.05 | 0.20 | 0.10 | 0.10 |
|        | IT       | 36    | 356       | 125       | 138       |
|        | CPU      | 0.1787| 0.1065    | 0.1829    | 0.1249    |
|        | RSE      | $8.46 \times 10^{-7}$ | $9.57 \times 10^{-7}$ | $9.98 \times 10^{-7}$ | $8.98 \times 10^{-7}$ |
| FGBK ($p = 2$) | $\eta_{exp}$ | 0.05 | 0.30 | 0.15 | 0.15 |
|        | IT       | 39    | 406       | 137       | 163       |
|        | CPU      | 0.1859| 0.0907    | 0.1735    | 0.1057    |
|        | RSE      | $8.27 \times 10^{-7}$ | $9.93 \times 10^{-7}$ | $9.44 \times 10^{-7}$ | $9.33 \times 10^{-7}$ |
| FGBK ($p = 3$) | $\eta_{exp}$ | 0.05 | 0.15 | 0.05 | 0.05 |
|        | IT       | 45    | 387       | 134       | 163       |
|        | CPU      | 0.2124| 0.0945    | 0.1802    | 0.1137    |
|        | RSE      | $9.26 \times 10^{-7}$ | $9.53 \times 10^{-7}$ | $9.17 \times 10^{-7}$ | $8.02 \times 10^{-7}$ |

Table 5 Information of the square matrices from SuiteSparse Matrix Collection

| Name  | poli      | add32     | fv1       |
|-------|-----------|-----------|-----------|
| $m \times n$ | 4008 $\times$ 4008 | 4960 $\times$ 4960 | 9604 $\times$ 9604 |
| Density | 0.05% | 0.08% | 0.09% |
| Cond($A$) | 311.53 | 136.68 | 8.81 |

method with $p = 1$ has the least number of iteration and the least CPU time. For the other three matrices, the proposed method with $p = 1$ requires the least number of iterations. It further indicates that a small value of $p$ may further improve the speed of convergence.

In Fig. 1, the curves of the relative solution error versus the number of iterations are plotted for GBK, FDBK, FGBK ($p = 1$), FGBK ($p = 2$) and FGBK ($p = 3$) respectively.

From Fig. 1, it is obviously observed that the fast greedy block Kaczmarz methods converge faster than the greedy block Kaczmarz method and the fast deterministic block Kaczmarz method, which confirms the numerical result in Table 4 and further shows the efficiency of the modified greedy row selection strategy.

In the fourth example, some square matrices are taken from the SuiteSparse Matrix Collection (Davis and Hu 2011). The test matrix ‘poli’ comes from an economic problem, ‘add32’ comes from a circuit simulation problem and ‘fv1’ comes from a two-dimensional problem. In Table 5, the sizes ($m \times n$), density and condition number of the test square matrices are listed respectively.

In Table 6, the number of iterations and CPU time of the greedy block Kaczmarz method, the fast deterministic block Kaczmarz method and the fast greedy block Kaczmarz method with $p = 1, 2$ and $3$ are reported respectively.
Fig. 1 Convergence curves for the matrices from SuiteSparse Matrix Collection

Table 6 Numerical results for the square matrices from SuiteSparse Matrix Collection

| Name   | GBK | FDBK | FGBK (p = 1) | FGBK (p = 2) | FGBK (p = 3) |
|--------|-----|------|--------------|--------------|--------------|
| Poli   |     |      | 0.10         | 0.05         | 0.50         |
| IT     | 37,496 | 62,330 | 14,886     | 30,688       | 21,892       |
| CPU    | 32.5671 | 20,6040 | 37,0953 | 18.0046     | 20.2644    |
| RSE    | 9.97 × 10^{-7} | 1.00 × 10^{-6} | 1.32 × 10^{-7} | 9.99 × 10^{-7} | 1.00 × 10^{-6} |
| Add32  |     |      | 0.05         | 0.05         | 0.10         |
| IT     |     |      | 34.988       | 39.991       | 42.443       |
| CPU    |     |      | 93.6431      | 94.3514      | 106.2767     |
| RSE    |     |      | 1.00 × 10^{-6} | 1.00 × 10^{-6} | 1.00 × 10^{-6} |
| Fv1    |     |      | 0.25         | 0.05         | 0.25         |
| IT     | 11,140 | 10,004 | 158         | 173          | 209          |
| CPU    | 30.6700 | 23.1204 | 13.9666 | 5.8104       | 9.6455       |
| RSE    | 9.99 × 10^{-7} | 1.00 × 10^{-6} | 9.76 × 10^{-7} | 9.89 × 10^{-7} | 9.80 × 10^{-7} |
From Table 6, it is observed that FGBK ($p = 1$), FGBK ($p = 2$) and FGBK ($p = 3$) methods converge successfully for all the square matrices, while GBK and FDBK method converge for all the matrices except ‘add32’. In addition, it is observed that the proposed methods require fewer number of iterations than the other two block Kaczmarz methods. The proposed method with $p = 1$ requires the least number of iterations for all the matrices, while the proposed method with $p = 2$ has the least CPU time for all the matrices except ‘add32’. It indicates that the fast greedy block Kaczmarz methods are efficient for solving linear systems with square coefficient matrices.

4 Conclusions

A fast greedy block Kaczmarz method is presented for solving large consistent linear systems. Theoretical analysis proves the convergence of the proposed methods and show that the upper bound of the convergence rate is related to the geometric properties of the coefficient matrix and its block submatrices. Numerical experiments further illustrate that the proposed methods are efficient and faster than the fast deterministic block Kaczmarz method.

Acknowledgements This work is supported by the National Natural Science Foundation of China (Grant No. 11971354).

Declarations

Conflict of interest The authors declare that they have no conflict of interest.

References

Bai Z-Z, Wu W-T (2018a) On greedy randomized Kaczmarz method for solving large sparse linear systems. SIAM J Sci Comput 40(1):A592–A606
Bai Z-Z, Wu W-T (2018b) On relaxed greedy randomized Kaczmarz methods for solving large sparse linear systems. Appl Math Lett 83:21–26
Bai Z-Z, Wu W-T (2019) On partially randomized extended Kaczmarz method for solving large sparse over-determined inconsistent linear systems. Linear Algebra Appl 578(1):225–250
Chen J-Q, Huang Z-D (2022) On a fast deterministic block Kaczmarz method for solving large-scale linear systems. Numer Algorithms 89(3):1007–1029
Davis TA, Hu Y-F (2011) The University of Florida sparse matrix collection. ACM Trans Math Softw 38(1):1–25
Du Y-S, Hayami K, Zheng N, Morikuni K, Yin J-F (2021) Kaczmarz-type inner-iteration preconditioned flexible GMRES methods for consistent linear systems. SIAM J Sci Comput 43(5):S345–S366
Elfving T (1980) Block-iterative methods for consistent and inconsistent linear equations. Numer Math 35(1):1–12
Filipović D, Glau K, Nakatsukasa Y, Statti F (2019) Weighted Monte Carlo with least squares and randomized extended Kaczmarz for option pricing. Swiss Finance Institute Research Paper, (19–54)
Herman GT, Davidi R (2008) Image reconstruction from a small number of projections. Inverse Probl 24(4):045011
Jia Z-G, Ng MK (2021) Structure preserving quaternion generalized minimal residual method. SIAM J Matrix Anal Appl 42(2):616–634
Kak AC, Slaney M (2001) Principles of computerized tomographic imaging. SIAM, Philadelphia
Karczmarz S (1937) Angenäherte auflösung von systemen linearer gleichungen. Bull Int Acad Pol Sic Lett A 35:355–357
Miao C-Q (2022) On greedy randomized average block Kaczmarz method for solving large linear systems. J Comput Appl Math 413:114372
Necoara I (2019) Faster randomized block Kaczmarz algorithms. SIAM J Matrix Anal Appl 40(4):1425–1452
Needell D, Tropp JA (2014) Paved with good intentions: analysis of a randomized block Kaczmarz method. Linear Algebra Appl 441:199–221
Needell D, Srebro N, Ward R (2016) Stochastic gradient descent, weighted sampling, and the randomized Kaczmarz algorithm. Math Program 155(1):549–573
Niu Y-Q, Zheng B (2020) A greedy block Kaczmarz algorithm for solving large-scale linear systems. Appl Math Lett 104:106294
Saad Y (2003) Iterative methods for sparse linear systems. Society for Industrial and Applied Mathematics
Strohmer T, Vershynin R (2009) A randomized Kaczmarz algorithm with exponential convergence. J Fourier Anal Appl 15(2):262–278
Yin J-F, Li N, Zheng N (2022) Restarted randomized surrounding methods for solving large linear equations. Appl Math Lett 133:108290

Publisher’s Note Springer Nature remains neutral with regard to jurisdictional claims in published maps and institutional affiliations.

Springer Nature or its licensor (e.g. a society or other partner) holds exclusive rights to this article under a publishing agreement with the author(s) or other rightsholder(s); author self-archiving of the accepted manuscript version of this article is solely governed by the terms of such publishing agreement and applicable law.