Networks and Heterogeneous Media

http://www.aimspress.com/journal/nhm

NHM, 19(1): 305–323.
DOI: 10.3934/nhm.2024014
Received: 11 December 2023
Revised: 14 March 2024
Accepted: 21 March 2024
Published: 27 March 2024

Research article

A class of pseudoinverse-free greedy block nonlinear Kaczmarz methods for nonlinear systems of equations

Ying Lv, Li-Li Xing*, Wen-Di Bao, Wei-Guo Li and Zhi-Wei Guo

College of Science, China University of Petroleum, Qingdao PR China

* **Correspondence:** Email: xinglily2010@upc.edu.cn.

Abstract: In this research, we constructed a class of nonlinear greedy average block Kaczmarz methods to solve nonlinear problems without computing the Moore-Penrose pseudoinverse of the Jacobian matrix. These kinds of methods adopt the average technique of the Gaussian Kaczmarz method and combine the greedy strategy, which greatly reduces the amount of computation. The local convergence analysis and numerical experiments of the proposed methods are given. The numerical results show the effectiveness of the proposed methods.

Keywords: nonlinear equations; average technique; nonlinear Kaczmarz algorithm; block nonlinear Kaczmarz algorithm

1. Introduction

Consider the problem of finding the roots of the system of the nonlinear equations

$$\mathbf{f}(\mathbf{x}) = \mathbf{0},$$

where $\mathbf{f} : \mathcal{D} \subseteq \mathbb{R}^n \to \mathbb{R}^m$. We assume throughout that $\mathbf{f}(\mathbf{x}) = [f_1(\mathbf{x}), \cdots, f_m(\mathbf{x})]^T$ is a continuously differentiable vector-valued function, and $\mathbf{x} = (\mathbf{x}_1, \cdots, \mathbf{x}_n)^T$ is an n-dimensional unknown vector. In this article, we exclusively study the overdetermined ($m \geq n$) nonlinear system, where there exists a single solution $\mathbf{x}_*$ such that $\mathbf{f}(\mathbf{x}_*) = \mathbf{0}$. This kind of nonlinear problem widely exists in practical applications, such as machine learning [6], differential equations [26], convex optimization, and deep neural networks [15].

The Newton-Raphson (NR) method, Broyden method [1], and directional secant method [2] are some iterative methods for solving nonlinear equations. The iterative formula of the Newton-Raphson method is as follows:

$$\mathbf{x}_{k+1} = \mathbf{x}_k - (\mathbf{f}'(\mathbf{x}_k))^{\dagger}\mathbf{f}(\mathbf{x}_k),$$

where $\mathbf{f}'(\mathbf{x}_k) = [\nabla \mathbf{f}_1(\mathbf{x}_k), \cdots, \nabla \mathbf{f}_m(\mathbf{x}_k)]^T \in \mathbb{R}^{m\times n}$ is the Jacobian matrix of $\mathbf{f}$ at $\mathbf{x}_k$. $\nabla \mathbf{f}_i(\mathbf{x}_k)^T$ is its i-th row and $(\mathbf{f}'(\mathbf{x}_k))^\dagger$ is the Moore-Penrose pseudoinverse of $\mathbf{f}'(\mathbf{x}_k)$. This approach, of course, is highly disadvantageous in the computation as it necessitates the computation of the full Jacobian matrix as well as its Moore-Penrose pseudoinverse. This leads to high calculation costs. In recent years, the nonlinear Kaczmarz method has also been developed. Wang, Li and Bao [28] proposed the nonlinear Kaczmarz (NK) method by generalizing the Kaczmarz method [14] to the nonlinear case, which still uses the core idea of the Kaczmarz method (i.e., only one row of the coefficient matrix is used in each iteration). Furthermore, for the case where the Jacobian matrix in the problem is singular, this method can be computed quickly and circumvents the shortcomings of classical methods such as the Newton method.

In contrast to the nonlinear Kaczmarz methods, the linear Kaczmarz methods are now more advanced. Next, we tried to get more efficient nonlinear Kaczmarz methods to use the optimization idea of the linear Kaczmarz methods. It is commonly known that the Kaczmarz method [14], whose main idea is to project the current point onto the solution space given by a row of the coefficient matrix, has drawn a lot of attention lately due to its computational simplicity and efficiency. In 2009, Strohmer and Vershynin proposed a randomized Kaczmarz (RK) [27] method, which selects row indexes in a random order, rather than a cyclic order, during each iteration. They proved the linear convergence of the method, which gave a great impetus to the research on the Kaczmarz methods [7–10, 16, 18, 23, 35]. Some researchers have also extended the Kaczmarz method to solve systems of inequality [21] and tensor equations [29].

Enhancing the Kaczmarz method includes two primary measures. One approach is to incorporate certain criteria to the selection of the working rows so that the larger residuals can be quickly eliminated. One of the most typical approaches is the greedy randomized Kaczmarz (GRK) method [4]. In addition, six typical work row selection rules are summarized in [3], which are the uniform, non-uniform, residual, distance, maximal residual, and maximal distance selection rules.

Another method is to select multiple working rows for each iteration, which we call the block method. By adopting the block idea [5, 11, 25], many researchers have sped up the standard Kaczmarz method's convergence. The block technique involves using a few rows of the coefficient matrix $\mathbf{A}$ for the linear system $\mathbf{Ax} = \mathbf{b}$ at each iteration. The following is a description of the block Kaczmarz technique [24]:

$$\mathbf{x}_{k+1} = \mathbf{x}_k + \mathbf{A}_{\tau_k}^\dagger(\mathbf{b}_{\tau_k} - \mathbf{A}_{\tau_k}\mathbf{x}_k), k = 0, 1, 2, \cdots,$$

where $\mathbf{A}_{\tau_k}^\dagger$ represents the Moore-Penrose pseudoinverse of the chosen submatrix $\mathbf{A}_{\tau_k}$ and τ_k is a subset of indicators selected according to some rule. However, the Moore-Penrose pseudoinverse must be computed for each iteration in the block Kaczmarz approach, and this is typically rather costly. Necoara [22] used some updated convex combinations as the new direction of the next iteration to build a unified framework for the randomized average block Kaczmarz method [30]. The Gaussian Kaczmarz method [13] can be regarded as another kind of block Kaczmarz method, that is

$$\mathbf{x}_{k+1} = \mathbf{x}_k + \frac{\eta^T(\mathbf{b} - \mathbf{A}\mathbf{x}_k)}{\|\mathbf{A}^T\eta\|_2^2}\mathbf{A}^T\eta,$$

where η is a Gaussian vector with mean $\mathbf{0} \in \mathbb{R}^m$ and the covariance matrix $\mathbf{I} \in \mathbb{R}^{m\times m}$ so that $\eta \sim N(\mathbf{0}, \mathbf{I})$.

In this paper, motivated by the above two ideas, we attempt to implement them in the nonlinear Kaczmarz method. The main contributions of this paper are as follows:

First, the common nonlinear block Kaczmarz method has the same drawback as the classical linear block Kaczmarz method in that it requires the computation of matrix pseudoinverses. The computation of the pseudoinverse of the Jacobian matrix requires a lot of computation, especially when the problem dimension is large. To solve this issue, this paper extends the averaging technique from linear to nonlinear methods.

Second, there are certain benefits for choosing the greedy criteria in this research. To avoid calculating the Frobenius norm of the entire Jacobian matrix, we used the second greedy rule in [33] as the criterion in the first method. In [32], Zhang et al. proposed a nonlinear Kaczmarz method with a greedy selection strategy, which is specified as follows:

$$i_k = arg \max_{1\le i\le m} |f_i(\mathbf{x}_k)|^2,$$

which aims at choosing the maximum component of the residual vector. They also showed that in terms of numerical experiments and theoretical analysis, the methods with greedy rules are faster than the NK method. Therefore, we used it as the second greedy criterion in this paper.

In summary, inspired by [32, 33], we extended the pseudoinverse-free block Kaczmarz method for solving linear equations to the nonlinear situation and incorporated greedy principles to accelerate the convergence of algorithms. We presented two kinds of pseudoinverse-free greedy block nonlinear Kaczmarz methods: the nonlinear greedy average block Kaczmarz (NGABK) method and the maximum residual nonlinear average block Kaczmarz (MRNABK) method. The convergence analyses of the two algorithms are given in detail. Numerical experiments showed that our proposed methods are more effective than the previous methods. In most cases, the MRNABK method was better than the NGABK method, and both of them were better than several state-of-the-art solvers.

The rest of this paper is organized as follows: In Section 2, the notations and preliminaries are provided. In Section 3, we provide the two pseudoinverse-free greedy block nonlinear Kaczmarz methods. We establish their convergence theorems in Section 4. The numerical experiments are given in Section 5. Finally, we make a summary of the whole work in Section 6.

2. Notations and preliminaries

For any matrix $\mathbf{A} \in \mathbb{R}^{m\times n}$, we use $\sigma_{max}(\mathbf{A})$, $\sigma_{min}(\mathbf{A})$, $\|\mathbf{A}\|_2$, $\|\mathbf{A}\|_F = \sqrt{\sum_{i=1}^{m}\sum_{j=1}^{n}|a_{ij}|^2}$, $\mathbf{A}^\dagger$, and $\mathbf{A}_\tau$ to denote the maximum and minimum nonzero singular values of $\mathbf{A}$, the spectral norm, the Frobenius norm, the Moore-Penrose pseudoinverse, and the row submatrix of matrix $\mathbf{A}$ indexed by the index set τ. $\mathbf{r}_k$ denotes the residual vector of the k-th iteration. $\mathbf{e}_i \in \mathbb{R}^m$ denotes the unit vector where the i-th element is 1 and the rest are 0. For an integer $m \ge 1$, let $[m] := \{1, \dots, m\}$. At the k-th iteration, $|\tau_k|$ is the cardinal number of the set τ_k. Set $\gamma > 0$, $\mathcal{B}(\mathbf{x}_*, \gamma) \triangleq \{\mathbf{x} \in \mathbb{R}^n \mid \|\mathbf{x} - \mathbf{x}_*\|_2 \le \gamma\}$.

Definition 1 ([28]). *If for every $i \in [m]$ and $\forall \mathbf{x}_1, \mathbf{x}_2 \in \mathcal{D} \subseteq \mathbb{R}^n$, there exists $\xi_i \in [0, \xi)$ satisfying $\xi = \max_i \xi_i < \frac{1}{2}$ such that*

$$|f_i(\mathbf{x}_1) - f_i(\mathbf{x}_2) - \nabla \mathbf{f}_i(\mathbf{x}_1)^T(\mathbf{x}_1 - \mathbf{x}_2)| \le \xi_i |f_i(\mathbf{x}_1) - f_i(\mathbf{x}_2)|,$$

then the function $\mathbf{f} : \mathcal{D} \subseteq \mathbb{R}^n \to \mathbb{R}^m$ is referred to satisfy the local tangential cone condition.

Lemma 1 ([34]). *If the function* $\mathbf{f}$ *satisfies the local tangential cone condition, then for* $\forall \mathbf{x}_1, \mathbf{x}_2 \in \mathcal{D} \subseteq \mathbb{R}^n$ *and an index subset* $\tau \subseteq [m]$*, we have*

$$\|\mathbf{f}_\tau(\mathbf{x}_1) - \mathbf{f}_\tau(\mathbf{x}_2)\|_2^2 \geq \frac{1}{1+\xi^2}\|\mathbf{f}'_\tau(\mathbf{x}_1)(\mathbf{x}_1 - \mathbf{x}_2)\|_2^2.$$

Lemma 2 ([19]). *Let* $\rho_1 \geq \cdots \geq \rho_n$ *and* $\zeta_1 \geq \cdots \geq \zeta_n$ *be the singular values of the matrices* $\mathbf{A}$ *and* $\mathbf{B}$ *respectively. Then*

$$\|\mathbf{A} - \mathbf{B}\| \geq \|diag(\rho_1 - \zeta_1, \ldots, \rho_n - \zeta_n)\|.$$

Lemma 3. *Suppose* $\mathbf{f}'(x)$ *is a full column rank matrix for* $\forall \mathbf{x} \in \mathcal{D} \subseteq \mathbb{R}^n$*. Then there exist* $\underline{\sigma}$ *and* $\overline{\sigma}$ *such that*

$$\inf_{\mathbf{x}\in\mathcal{D}} \sigma_{min}(\mathbf{f}'(\mathbf{x})) = \underline{\sigma} > 0,$$
$$\sup_{\mathbf{x}\in\mathcal{D}} \sigma_{max}(\mathbf{f}'(\mathbf{x})) = \overline{\sigma} < \infty.$$

Proof. For any $\mathbf{x} \in \mathcal{D}$, $\mathbf{f}'(\mathbf{x})$ is full column rank, then we have $\sigma_{max}(\mathbf{f}'(\mathbf{x})) = \sigma_1(\mathbf{f}'(\mathbf{x})) \geq \cdots \geq \sigma_n(\mathbf{f}'(\mathbf{x})) = \sigma_{min}(\mathbf{f}'(\mathbf{x})) > 0$. By Lemma 2, $\sigma_i(\mathbf{f}'(\mathbf{x}))(i = 1, \ldots, n)$ is a continuous function of $\mathbf{f}'(\mathbf{x})$. In addition, $\mathbf{f}'(\mathbf{x})$ continuously depends on $\mathbf{x}$. Then, $\sigma_i(\mathbf{x})(i = 1, \ldots, n)$ is a continuous function of $\mathbf{x}$.

Since $\mathcal{D}$ is bounded closed and $\sigma_{min}(\mathbf{x})$ is a continuous function of $\mathbf{x}$, there exists a point $\underline{\mathbf{x}} \in \mathcal{D}$ such that $\underline{\sigma} = \sigma_{min}(\underline{\mathbf{x}}) = \inf_{\mathbf{x}\in\mathcal{D}} \sigma_{min}(\mathbf{x}) > 0$. Then, we have

$$\sigma_{min}(\mathbf{f}'(\mathbf{x})) \geq \inf_{\mathbf{x}\in\mathcal{D}} \sigma_{min}(\mathbf{f}'(\mathbf{x})) = \underline{\sigma} > 0.$$

Similarly, since $\mathcal{D}$ is bounded closed and $\sigma_{max}(\mathbf{x})$ is a continuous function of $\mathbf{x}$, there exists a point $\overline{\mathbf{x}} \in \mathcal{D}$ such that $\overline{\sigma} = \sigma_{max}(\overline{\mathbf{x}}) = \sup_{\mathbf{x}\in\mathcal{D}} \sigma_{max}(\mathbf{x}) < \infty$. Then, we have

$$\sigma_{max}(\mathbf{f}'(\mathbf{x})) \leq \sup_{\mathbf{x}\in\mathcal{D}} \sigma_{max}(\mathbf{f}'(\mathbf{x})) = \overline{\sigma} < \infty.$$

□

3. Pseudoinverse-free greedy block nonlinear Kaczmarz methods

Yuan et al. developed a randomized NR method based on the sketch-and-project technique [31], which is called the sketched Newton-Raphson (SNR) method. The formula of the SNR method is written as follows,

$$\mathbf{x}_{k+1} = \mathbf{x}_k - (\mathbf{f}'(\mathbf{x}_k))^T \mathbf{S}_k (\mathbf{S}_k^T f'(\mathbf{x}_k)(\mathbf{f}'(\mathbf{x}_k))^T \mathbf{S}_k)^\dagger \mathbf{S}_k^T \mathbf{f}(\mathbf{x}_k). \tag{3.1}$$

When $\mathbf{S}_k = \eta_k = \sum_{i\in\tau_k}(-f_i(\mathbf{x}_k))\mathbf{e}_i$ in Eq (3.1), we get an iterative formula as follows:

$$\mathbf{x}_{k+1} = \mathbf{x}_k - \frac{\eta_k^T \mathbf{f}(\mathbf{x}_k)}{\|\mathbf{f}'(\mathbf{x}_k)^T \eta_k\|_2^2} \mathbf{f}'(\mathbf{x}_k)^T \eta_k.$$

Based on the greedy randomized Kaczmarz method [4], the block indices $\mathcal{I}_k$ in distance-residual capped nonlinear Kaczmarz (DR-CNK) method [33] is chosen by

$$\mathcal{I}_k = \{i \Big| |f_i(\mathbf{x}_k)|^2 \geq \delta_k \|\mathbf{f}(\mathbf{x}_k)\|_2^2 \|\nabla \mathbf{f}_i(\mathbf{x}_k)\|_2^2\}$$

Algorithm 1: The NGABK/MRNABK algorithm

Require: The initial estimate $\mathbf{x}_0 \in \mathcal{B}(x^*, \gamma) \subseteq \mathcal{D} \subseteq \mathbb{R}^n$

1: **for** $k = 1, 2, \cdots$ until convergence, **do**

2: Compute and determine the index subset

case 1: the NGABK method

$$\tau_k = \{i \Big| |f_i(\mathbf{x}_k)|^2 \geq \delta_k \|\mathbf{f}(\mathbf{x}_k)\|_2^2\},$$

where $\delta_k = \frac{1}{2}(\frac{\max\limits_{i\in m}|f_i(\mathbf{x}_k)|^2}{\|\mathbf{f}(\mathbf{x}_k)\|_2^2} + \frac{1}{m})$.

case 2: the MRNABK method

$$\tau_k = \{i \Big| |f_i(\mathbf{x}_k)|^2 \geq \varrho \max_{1\leq i\leq m} |f_i(\mathbf{x}_k)|^2\}$$

3: Compute

$$\eta_k = \sum_{i\in\tau_k} (-f_i(\mathbf{x}_k))\mathbf{e}_i$$

4: Set

$$\mathbf{x}_{k+1} = \mathbf{x}_k - \frac{\eta_k^T \mathbf{f}(\mathbf{x}_k)}{\|\mathbf{f}'(\mathbf{x}_k)^T \eta_k\|_2^2} \mathbf{f}'(\mathbf{x}_k)^T \eta_k \tag{3.2}$$

5: **end for**

with

$$\delta_k = \frac{1}{2}\left(\frac{1}{\|\mathbf{f}(\mathbf{x}_k)\|_2^2} \max_{i\in[m]} \frac{|f_i(\mathbf{x}_k)|^2}{\|\nabla \mathbf{f}_i(\mathbf{x}_k)\|_2^2} + \frac{1}{\|\mathbf{f}'(\mathbf{x}_k)\|_F^2}\right),$$

that is, the information of the entire Jacobian matrix is required. This will result in a large amount of computation. Now, by choosing the τ_k by

$$\tau_k = \{i \Big| |f_i(\mathbf{x}_k)|^2 \geq \delta_k \|\mathbf{f}(\mathbf{x}_k)\|_2^2\}$$

with

$$\delta_k = \frac{1}{2}(\frac{\max\limits_{i\in m}|f_i(\mathbf{x}_k)|^2}{\|\mathbf{f}(\mathbf{x}_k)\|_2^2} + \frac{1}{m}),$$

we get a nonlinear greedy average block Kaczmarz (NGABK) method which is described in case 1 of Algorithm 1. At the k-th iteration, a random vector η_k is drawn and a search direction is formed by $\mathbf{f}'(\mathbf{x}_k)^T \eta_k$. When $\mathbf{f}'(\mathbf{x}_k)^T \eta_k = 0$, no line search is performed.

Remark 1. *The computational complexity of the iterative formula of our proposed method is $O(|\tau_k|n)$ much smaller than that of the Newton-Raphson method $O(n^2)$ at each iteration. Here, in conjunction with the numerical experiments later, the size of τ_k is usually m/3.*

Remark 2. *The index set τ_k is always nonempty. Because*

$$\max_{i\in[m]} |f_i(\mathbf{x}_k)|^2 \geq \sum_{i=1}^{m} \frac{|f_i(\mathbf{x}_k)|^2}{m} = \frac{\|\mathbf{f}(\mathbf{x}_k)\|_2^2}{m}$$

and then

$$|f_{i_k}(\mathbf{x}_k)|^2 = \max_{i\in[m]} |f_i(\mathbf{x}_k)|^2 \geq \frac{1}{2}\left(\max_{i\in[m]} |f_i(\mathbf{x}_k)|^2 + \frac{\|\mathbf{f}(\mathbf{x}_k)\|_2^2}{m}\right)$$

implies $i_k \in \tau_k$. Therefore, the method is well-defined.

According to the maximum residual rule, we establish the maximum residual nonlinear average block Kaczmarz (MRNABK) method in case 2 of Algorithm 1. In this method, $\varrho \in [0, 1]$ is the relaxation parameter, which can be determined in numerical experiments. It is obvious that in the k-th iteration of the MRNABK Algorithm, the set τ_k is also non-empty. This is because

$$|f_{i_k}(\mathbf{x}_k)| = \max_{1\leq i\leq m} |f_i(\mathbf{x}_k)|.$$

That is to say, the largest residual component i_k is always in the set τ_k.

From the MRNABK method, we can see that the larger components of the residual are eliminated preferentially, which greatly improves the efficiency of the algorithm. We establish its convergence theorem in Section 4.

4. Convergence analysis

Lemma 4. *If the function* $\mathbf{f}$ *satisfies the local tangential cone condition, then for* $i \in [m]$, $\xi = \max_{i\in[m]} \xi_i < \frac{1}{2}$, $\forall \mathbf{x}_1, \mathbf{x}_2 \in \mathcal{B}(\mathbf{x}^*, \gamma) \subseteq \mathcal{D}$ *and the updating formula (3.2), we have*

$$\|\mathbf{x}_{k+1} - \mathbf{x}_*\|_2^2 \leq \|\mathbf{x}_k - \mathbf{x}_*\|_2^2 - (1 - 2\xi)\frac{|\eta_k^T \mathbf{f}(\mathbf{x}_k)|^2}{\|\mathbf{f}'(\mathbf{x}_k)^T \eta_k\|_2^2}.$$

Proof. From the updating formula (3.2), we have

$$\begin{aligned}\|\mathbf{x}_{k+1} - \mathbf{x}_*\|_2^2 &= \|\mathbf{x}_k - \frac{\eta_k^T \mathbf{f}(\mathbf{x}_k)}{\|\mathbf{f}'(\mathbf{x}_k)^T \eta_k\|_2^2}\mathbf{f}'(\mathbf{x}_k)^T \eta_k - \mathbf{x}_*\|_2^2 \\ &= \|\mathbf{x}_k - \mathbf{x}_*\|_2^2 - 2\left\langle \frac{\eta_k^T \mathbf{f}(\mathbf{x}_k)}{\|\mathbf{f}'(\mathbf{x}_k)^T \eta_k\|_2^2}\mathbf{f}'(\mathbf{x}_k)^T \eta_k, \mathbf{x}_k - \mathbf{x}_* \right\rangle + \frac{|\eta_k^T \mathbf{f}(\mathbf{x}_k)|^2}{\|\mathbf{f}'(\mathbf{x}_k)^T \eta_k\|_2^2}\end{aligned}$$

According to the definition of η_k and $f_i(\mathbf{x}_*) = 0$, we have

$$\begin{aligned}
&\|\mathbf{x}_{k+1} - \mathbf{x}_*\|_2^2 \\
&= \|\mathbf{x}_k - \mathbf{x}_*\|_2^2 - 2\frac{\eta_k^T \mathbf{f}(\mathbf{x}_k)}{\|\mathbf{f}'(\mathbf{x}_k)^T \eta_k\|_2^2}(-\sum_{i\in\tau_k} f_i(\mathbf{x}_k)\mathbf{e}_i^T(\mathbf{f}'(\mathbf{x}_k)))(\mathbf{x}_k - \mathbf{x}_*) + \frac{|\eta_k^T \mathbf{f}(\mathbf{x}_k)|^2}{\|\mathbf{f}'(\mathbf{x}_k)^T \eta_k\|_2^2} \\
&= \|\mathbf{x}_k - \mathbf{x}_*\|_2^2 + 2\frac{\eta_k^T \mathbf{f}(\mathbf{x}_k)}{\|\mathbf{f}'(\mathbf{x}_k)^T \eta_k\|_2^2}(\sum_{i\in\tau_k} f_i(\mathbf{x}_k)\nabla\mathbf{f}_i(\mathbf{x}_k)^T))(\mathbf{x}_k - \mathbf{x}_*) + \frac{|\eta_k^T \mathbf{f}(\mathbf{x}_k)|^2}{\|\mathbf{f}'(\mathbf{x}_k)^T \eta_k\|_2^2} \\
&= \|\mathbf{x}_k - \mathbf{x}_*\|_2^2 - 2\frac{\eta_k^T \mathbf{f}(\mathbf{x}_k)}{\|\mathbf{f}'(\mathbf{x}_k)^T \eta_k\|_2^2}\sum_{i\in\tau_k} f_i(\mathbf{x}_k)(f_i(\mathbf{x}_k) - f_i(\mathbf{x}_*) - \nabla\mathbf{f}_i(\mathbf{x}_k)^T(\mathbf{x}_k - \mathbf{x}_*)) \\
&+ 2\frac{\eta_k^T \mathbf{f}(\mathbf{x}_k)}{\|\mathbf{f}'(\mathbf{x}_k)^T \eta_k\|_2^2}\sum_{i\in\tau_k} f_i^2(\mathbf{x}_k) + \frac{|\eta_k^T \mathbf{f}(\mathbf{x}_k)|^2}{\|\mathbf{f}'(\mathbf{x}_k)^T \eta_k\|_2^2}.
\end{aligned}$$

From Definition 1, we have

$$\begin{aligned}
\|\mathbf{x}_{k+1} - \mathbf{x}_*\|_2^2 &\le \|\mathbf{x}_k - \mathbf{x}_*\|_2^2 + 2\frac{|\eta_k^T \mathbf{f}(\mathbf{x}_k)|^2}{\|\mathbf{f}'(\mathbf{x}_k)^T \eta_k\|_2^2}\xi - \frac{|\eta_k^T \mathbf{f}(\mathbf{x}_k)|^2}{\|\mathbf{f}'(\mathbf{x}_k)^T \eta_k\|_2^2} \\
&= \|\mathbf{x}_k - \mathbf{x}_*\|_2^2 - (1-2\xi)\frac{|\eta_k^T \mathbf{f}(\mathbf{x}_k)|^2}{\|\mathbf{f}'(\mathbf{x}_k)^T \eta_k\|_2^2}.
\end{aligned}$$

□

Remark 3. *It follows from Lemma 4 that $\mathbf{x}_{k+1} \in \mathcal{B}(\mathbf{x}^*, \gamma) \subseteq \mathcal{D}$ when $\mathbf{x}_k \in \mathcal{B}(\mathbf{x}^*, \gamma) \subseteq \mathcal{D}$. So, if $\mathbf{f}'(\mathbf{x})$ is a full column rank matrix, the iterative sequence $\{\mathbf{x}_k\}$ generated by the algorithm is well-defined.*

Theorem 1. *Consider that the nonlinear system of equations $\mathbf{f}(\mathbf{x}) = 0$, $\mathbf{f} : \mathcal{D} \subseteq \mathbb{R}^n \to \mathbb{R}^m$ on a bounded closed set $\mathcal{D}$, and there exists $\mathbf{x}_*$ such that $\mathbf{f}(\mathbf{x}_*) = \mathbf{0}$. For $\forall \mathbf{x} \in \mathcal{D}$, the nonlinear function $\mathbf{f}$ satisfies the local tangential cone condition given in Definition 1, $\xi = \max_{i\in[m]} \xi_i < \frac{1}{2}$ and $\mathbf{f}'(\mathbf{x})$ is a full column rank matrix. Assume that $\mathbf{x}_0 \in \mathcal{B}(\mathbf{x}^*, \gamma) \subseteq \mathcal{D} \subseteq \mathbb{R}^n$, then the iterations of the NGABK method in case 1 of Algorithm 1 satisfy*

$$\|\mathbf{x}_{k+1} - \mathbf{x}_*\|_2^2 \le \left(1 - \frac{1-2\xi}{1+\xi^2}\frac{\underline{\sigma}^2}{m\overline{\sigma}^2}\right)\|\mathbf{x}_k - \mathbf{x}_*\|_2^2.$$

Proof. Let $\mathbf{E}_k \in \mathbb{R}^{m\times|\tau_k|}$ be the matrix whose columns consist of all the vectors $\mathbf{e}_i \in \mathbb{R}^m$ with $i \in \tau_k$. Denote $\mathbf{f}'_{\tau_k}(\mathbf{x}_k) = \mathbf{E}_k^T \mathbf{f}'(\mathbf{x}_k)$, $\hat{\eta}_k = \mathbf{E}_k^T \eta_k$, then

$$\|\hat{\eta}_k\|_2^2 = \eta_k^T \mathbf{E}_k \mathbf{E}_k^T \eta_k = \|\eta_k\|_2^2 = \sum_{i\in\tau_k} |f_i(\mathbf{x}_k)|^2$$

and

$$\begin{aligned}
\|\mathbf{f}'(\mathbf{x}_k)^T \eta_k\|_2^2 &= \eta_k^T \mathbf{f}'(\mathbf{x}_k)\mathbf{f}'(\mathbf{x}_k)^T \eta_k = \hat{\eta}_k^T \mathbf{E}_k^T \mathbf{f}'(\mathbf{x}_k)\mathbf{f}'(\mathbf{x}_k)^T \mathbf{E}_k \hat{\eta}_k \\
&= \hat{\eta}_k^T \mathbf{f}'_{\tau_k}(\mathbf{x}_k)\mathbf{f}'_{\tau_k}(\mathbf{x}_k)^T \hat{\eta}_k = \|\mathbf{f}'_{\tau_k}(\mathbf{x}_k)\hat{\eta}_k\|_2^2.
\end{aligned}$$

Therefore, we have

$$\|f'_{\tau_k}(\mathbf{x}_k)^T \hat{\eta}_k\|_2^2 = \hat{\eta}_k^T f'_{\tau_k}(\mathbf{x}_k) f'_{\tau_k}(\mathbf{x}_k)^T \hat{\eta}_k \le \sigma_{max}^2(f'_{\tau_k}(\mathbf{x}_k))\|\hat{\eta}_k\|_2^2,$$

where $\sigma_{max}(\mathbf{f}'_{\tau_k}(\mathbf{x}_k))$ is the largest singular value of submatrix $\mathbf{f}'_{\tau_k}(\mathbf{x}_k)$ of the Jacobian matrix $\mathbf{f}'(\mathbf{x}_k)$. From the definition of η_k, we have

$$\begin{aligned}\eta_k^T(-\mathbf{f}(\mathbf{x}_k)) &= \left(\sum_{i\in\tau_k}(-f_i(\mathbf{x}_k))\mathbf{e}_i^T\right)(-\mathbf{f}(\mathbf{x}_k))\\ &= \sum_{i\in\tau_k} f_i(\mathbf{x}_k)\mathbf{e}_i^T(\mathbf{f}(\mathbf{x}_k))\\ &= \sum_{i\in\tau_k}|f_i(\mathbf{x}_k)|^2\\ &= \|\hat{\eta}_k\|_2^2.\end{aligned}$$

From the definition of τ_k, we have

$$\begin{aligned}\frac{|\eta_k^T(-\mathbf{f}(\mathbf{x}_k))|^2}{\|\mathbf{f}'(\mathbf{x}_k)^T\eta_k\|_2^2} &= \frac{\left(\sum_{i\in\tau_k}|f_i(\mathbf{x}_k)|^2\right)\|\hat{\eta}_k\|_2^2}{\|\mathbf{f}'_{\tau_k}(\mathbf{x}_k)^T\hat{\eta}_k\|_2^2}\\ &\geq \frac{\sum_{i\in\tau_k}|f_i(\mathbf{x}_k)|^2}{\sigma_{max}^2(\mathbf{f}'_{\tau_k}(\mathbf{x}_k))}\\ &\geq \frac{\sum_{i\in\tau_k}\delta_k\|\mathbf{f}(\mathbf{x}_k)\|_2^2}{\sigma_{max}^2(\mathbf{f}'_{\tau_k}(\mathbf{x}_k))}\\ &= \frac{\delta_k|\tau_k|}{\sigma_{max}^2(\mathbf{f}'_{\tau_k}(\mathbf{x}_k))}\|\mathbf{f}(\mathbf{x}_k)-\mathbf{f}(\mathbf{x}_*)\|_2^2.\end{aligned}$$

$$\begin{aligned}\frac{|\eta_k^T\mathbf{f}(\mathbf{x}_k)|^2}{\|\mathbf{f}'(\mathbf{x}_k)^T\eta_k\|_2^2} &\geq \frac{\delta_k|\tau_k|}{\sigma_{max}^2(\mathbf{f}'_{\tau_k}(\mathbf{x}_k))}\cdot\frac{1}{1+\xi^2}\|\mathbf{f}'(\mathbf{x}_k)(\mathbf{x}_k-\mathbf{x}_*)\|_2^2\\ &\geq \frac{\delta_k|\tau_k|}{\sigma_{max}^2(\mathbf{f}'_{\tau_k}(\mathbf{x}_k))}\cdot\frac{1}{1+\xi^2}\cdot\sigma_{min}^2(\mathbf{f}'(\mathbf{x}_k))\|\mathbf{x}_k-\mathbf{x}_*\|_2^2.\end{aligned}$$

Further, using Lemma 4, we can obtain

$$\begin{aligned}\|\mathbf{x}_{k+1}-\mathbf{x}_*\|_2^2 &\leq \|\mathbf{x}_k-\mathbf{x}_*\|_2^2-\frac{(1-2\xi)\delta_k|\tau_k|\sigma_{min}^2(\mathbf{f}'(\mathbf{x}_k))}{(1+\xi^2)\sigma_{max}^2(\mathbf{f}'_{\tau_k}(\mathbf{x}_k))}\|\mathbf{x}_k-\mathbf{x}_*\|_2^2\\ &= \left(1-\frac{(1-2\xi)\delta_k|\tau_k|\sigma_{min}^2(\mathbf{f}'(\mathbf{x}_k))}{(1+\xi^2)\sigma_{max}^2(\mathbf{f}'_{\tau_k}(\mathbf{x}_k))}\right)\|\mathbf{x}_k-\mathbf{x}_*\|_2^2.\end{aligned}$$

In addition, we have $\sigma_{max}^2(\mathbf{f}'_{\tau_k}(\mathbf{x}_k)) = \|\mathbf{f}'_{\tau_k}(\mathbf{x}_k)\|_2^2 \leq \|\mathbf{f}'(\mathbf{x}_k)\|_2^2 \leq \overline{\sigma}^2$ and $\delta_k = \frac{1}{2}(\frac{\max_{i\in m}|f_i(\mathbf{x}_k)|^2}{\|\mathbf{f}(\mathbf{x}_k)\|_2^2}+\frac{1}{m}) \geq \frac{1}{m}$ and use Lemma 3, so

$$\begin{aligned}\|\mathbf{x}_{k+1}-\mathbf{x}_*\|_2^2 &\leq \left(1-\frac{(1-2\xi)\delta_k|\tau_k|\sigma_{min}^2(\mathbf{f}'(\mathbf{x}_k))}{(1+\xi^2)\sigma_{max}^2(\mathbf{f}'_{\tau_k}(\mathbf{x}_k))}\right)\|\mathbf{x}_k-\mathbf{x}_*\|_2^2\\ &\leq \left(1-\frac{1-2\xi}{1+\xi^2}\frac{\underline{\sigma}^2}{m\overline{\sigma}^2}\right)\|\mathbf{x}_k-\mathbf{x}_*\|_2^2.\end{aligned}$$

So, the convergence of the NGABK method is proved. □

Remark 4. Since $1-2\xi < 1 < 1+\xi^2$ and $\underline{\sigma}^2 < m\overline{\sigma}^2$, we have

$$\rho_{NGABK} = 1 - \frac{1-2\xi}{1+\xi^2}\frac{\underline{\sigma}^2}{m\overline{\sigma}^2} < 1.$$

This shows that the convergence factor of our method is strictly smaller than 1.

Now, we give the convergence theorem of the MRNABK method.

Theorem 2. *Consider that the nonlinear system of equations* $\mathbf{f}(\mathbf{x}) = 0$, $\mathbf{f} : \mathcal{D} \subseteq \mathbb{R}^n \to \mathbb{R}^m$ *on a bounded closed set* $\mathcal{D}$*, and there exists* $\mathbf{x}_*$ *such that* $\mathbf{f}(\mathbf{x}_*) = \mathbf{0}$*. For* $\forall \mathbf{x} \in \mathcal{D}$*, the nonlinear function* $\mathbf{f}$ *satisfies the local tangential cone condition given in Definition 1,* $\xi = \max\limits_{i\in[m]} \xi_i < \dfrac{1}{2}$ *and* $\mathbf{f}'(\mathbf{x})$ *is a full column rank matrix. Assume that* $\mathbf{x}_0 \in \mathcal{B}(\mathbf{x}^*, \gamma) \subseteq \mathcal{D} \subseteq \mathbb{R}^n$*, then the iterations of the MRNABK method in case 2 of Algorithm 1 satisfy*

$$\|\mathbf{x}_{k+1} - \mathbf{x}_*\|_2^2 \le \left(1 - \frac{(1-2\xi)\varrho\underline{\sigma}^2}{(1+\xi^2)m\overline{\sigma}^2}\right)\|\mathbf{x}_k - \mathbf{x}_*\|_2^2.$$

Proof. Following an analogous proof process to the NGABK method, we get the following formula:

$$\begin{aligned}
\frac{|\eta_k^T(-\mathbf{f}(\mathbf{x}_k))|^2}{\|\mathbf{f}'(\mathbf{x}_k)^T\eta_k\|_2^2} &= \frac{\left(\sum_{i\in\tau_k}|f_i(\mathbf{x}_k)|^2\right)\|\hat{\eta}_k\|_2^2}{\|\mathbf{f}'_{\tau_k}(\mathbf{x}_k)^T\hat{\eta}_k\|_2^2} \\
&\ge \frac{\sum_{i\in\tau_k}|f_i(\mathbf{x}_k)|^2}{\sigma_{max}^2(\mathbf{f}'_{\tau_k}(\mathbf{x}_k))} \\
&\ge \frac{\sum_{i\in\tau_k}\varrho\max\limits_{1\le i\le m}|f_i(\mathbf{x}_k)|^2}{\sigma_{max}^2(\mathbf{f}'_{\tau_k}(\mathbf{x}_k))} \\
&\ge \frac{\varrho|\tau_k|}{m\sigma_{max}^2(\mathbf{f}'_{\tau_k}(\mathbf{x}_k))}\|\mathbf{f}(\mathbf{x}_k) - \mathbf{f}(\mathbf{x}_*)\|_2^2.
\end{aligned}$$

The second inequality follows from the definition of τ_k. Using $\max\limits_{1\le i\le m}|f_i(\mathbf{x}_k)|^2 \ge \frac{1}{m}\|\mathbf{f}(\mathbf{x}_k)\|_2^2$, we can get the third inequality.

From Lemma 1, it follows that

$$\begin{aligned}
\frac{|\eta_k^T\mathbf{f}(\mathbf{x}_k)|^2}{\|\mathbf{f}'(\mathbf{x}_k)^T\eta_k\|_2^2} &\ge \frac{\varrho|\tau_k|}{m\sigma_{max}^2(\mathbf{f}'_{\tau_k}(\mathbf{x}_k))}\cdot\frac{1}{1+\xi^2}\|\mathbf{f}'(\mathbf{x}_k)(\mathbf{x}_k - \mathbf{x}_*)\|_2^2 \\
&\ge \frac{\varrho|\tau_k|}{m\sigma_{max}^2(\mathbf{f}'_{\tau_k}(\mathbf{x}_k))}\cdot\frac{1}{1+\xi^2}\cdot\sigma_{min}^2(\mathbf{f}'(\mathbf{x}_k))\|\mathbf{x}_k - \mathbf{x}_*\|_2^2.
\end{aligned}$$

Further, using Lemmas 4 and 3, we obtain

$$\begin{aligned}
\|\mathbf{x}_{k+1} - \mathbf{x}_*\|_2^2 &\le \|\mathbf{x}_k - \mathbf{x}_*\|_2^2 - \frac{(1-2\xi)\varrho|\tau_k|\sigma_{min}^2(\mathbf{f}'(\mathbf{x}_k))}{(1+\xi^2)m\sigma_{max}^2(\mathbf{f}'_{\tau_k}(\mathbf{x}_k))}\|\mathbf{x}_k - \mathbf{x}_*\|_2^2 \\
&= \left(1 - \frac{(1-2\xi)\varrho|\tau_k|\sigma_{min}^2(\mathbf{f}'(\mathbf{x}_k))}{(1+\xi^2)m\sigma_{max}^2(\mathbf{f}'_{\tau_k}(\mathbf{x}_k))}\right)\|\mathbf{x}_k - \mathbf{x}_*\|_2^2 \\
&\le \left(1 - \frac{(1-2\xi)\varrho\underline{\sigma}^2}{(1+\xi^2)m\overline{\sigma}^2}\right)\|\mathbf{x}_k - \mathbf{x}_*\|_2^2.
\end{aligned}$$

So, the convergence of the MRNABK method is proved. □

Remark 5. Similarly, we have

$$\rho_{MRNABK} = 1 - \frac{(1-2\xi)\varrho\underline{\sigma}^2}{(1+\xi^2)m\overline{\sigma}^2}.$$

This shows that the convergence factor of our method is strictly smaller than 1.

5. Numerical examples

In this section, we primarily compare the efficiency of our new methods with the Broyden method [1], the NRK method [28], the residual-distance capped nonlinear Kaczmarz (RD-CNK) method, and the residual-based block capped nonlinear Kaczmarz (RB-CNK) method [33] for solving the nonlinear systems of equations in the iteration steps (denoted as 'IT') and computing time in seconds (denoted as 'CPU'). The RD-CNK method and the NRK method are based on a single sample. The RB-CNK method is based on multi-sampling and uses the following iteration scheme:

$$\mathbf{x}_{k+1} = \mathbf{x}_k - (\mathbf{f}'_{\mathcal{I}_k}(\mathbf{x}_k))^{\dagger}\mathbf{f}_{\mathcal{I}_k}(\mathbf{x}_k),$$

where $\mathcal{I}_k$ is the selected index subset. The target block in the MRNABK method is calculated by

$$\mathcal{I}_k = \{i \Big| |f_i(\mathbf{x}_k)|^2 \geq \varrho \max_{1\leq i\leq m} |f_i(\mathbf{x}_k)|^2\},$$

where $\varrho \in (0, 1]$. However, the choice of ϱ is only for the experiments in this paper.

In the numerical experiment, IT and CPU are the average of the results of 10 times repeated runs of the corresponding method. All experiments are terminated when the number of iterations exceeds 200,000 or $\|\mathbf{f}(\mathbf{x}_k)\|_2^2 < 10^{-6}$. All of the tables below show the number of IT and CPU required for several algorithms to reach the stop preparation ("–" indicates that the convergence cannot be achieved under the stop criterion). Additionally, the logarithm diagram of the norm of the nonlinear residual and the number of IT, as well as the connection diagram between the number of CPU or IT and the number of equations, are provided for each experiment. Our experiment is implemented on MATLAB (version R2018b).

Example 1. In this example, we consider the following equations:

$$f_i(\mathbf{x}) = \mathbf{x}_i - (1 - \frac{c}{2N}\sum_{j=1}^{N}\frac{\mu_i\mathbf{x}_j}{\mu_i+\mu_j})^{-1}, i = 1, 2, \ldots, N.$$

The system of equations is called H-equation, which is usually used to solve the problem of outlet distribution in radiation transmission [28]. In this problem, N represents the number of equations and $\mu_i = (i - \frac{1}{2})/N$. When $c \in (0, 1)$, the discrete problem has solutions. We set $\mathbf{x}_0$ be the zero vector and $c = 0.9$. In the Broyden method, we set the approximate matrix for the Jacobian matrix to be the identity matrix. First of all, we tested the value of parameter ϱ. In Table 1, we observed that in most cases, the MRNABK method required relatively less computing time, when $\varrho = 0.1, 0.2, 0.3$. When the number of equations was fixed, we found that the larger ϱ was, the longer the calculation time of the MRNABK method was. So, in this example, we set $\varrho = 0.1$. Next, we tested the performance of

our methods and other methods. The results of the numerical experiments are listed in Tables 2 and 3. The results show that the NGABK method and the RB-CNK method based on multiple sampling were substantially faster than the RD-CNK method and the NRK method based on single sampling, as shown in Figure 1. Figure 2 plots the running time (CPU) and iteration steps (IT) of different methods for different N. According to Figure 2, the NGABK method outperformed the RB-CNK method in terms of CPU. From Table 4, the MRNABK method converged faster than the NGABK method in terms of computing time and iteration steps. For H-equation with $\mathbf{f} : \mathbb{R}^n \to \mathbb{R}^n$, the Broyden method had a better numerical result.

Table 1. CPU of MRNABK for the H-equation with $c = 0.9$, $\mathbf{x}_0 = \mathbf{0}$, and different ϱ.

ϱ	0.1	0.3	0.5	0.7	0.8	0.9
$m = 50$	**0.018**	0.0260	0.0218	0.0282	0.0334	0.0441
$m = 100$	0.0958	**0.0863**	0.1607	0.1440	0.2222	0.1946
$m = 500$	1.2712	**1.2321**	1.3973	1.5922	1.7479	2.0144
$m = 1000$	**3.9016**	4.1044	4.8689	5.1860	5.9891	6.7390
$m = 1500$	**8.2839**	8.9858	10.4272	11.0643	11.8178	13.9577

Table 2. IT comparison of Broyden method, NRK, RD-CNK, NGABK, RB-CNK, and MRNABK for the H-equation.

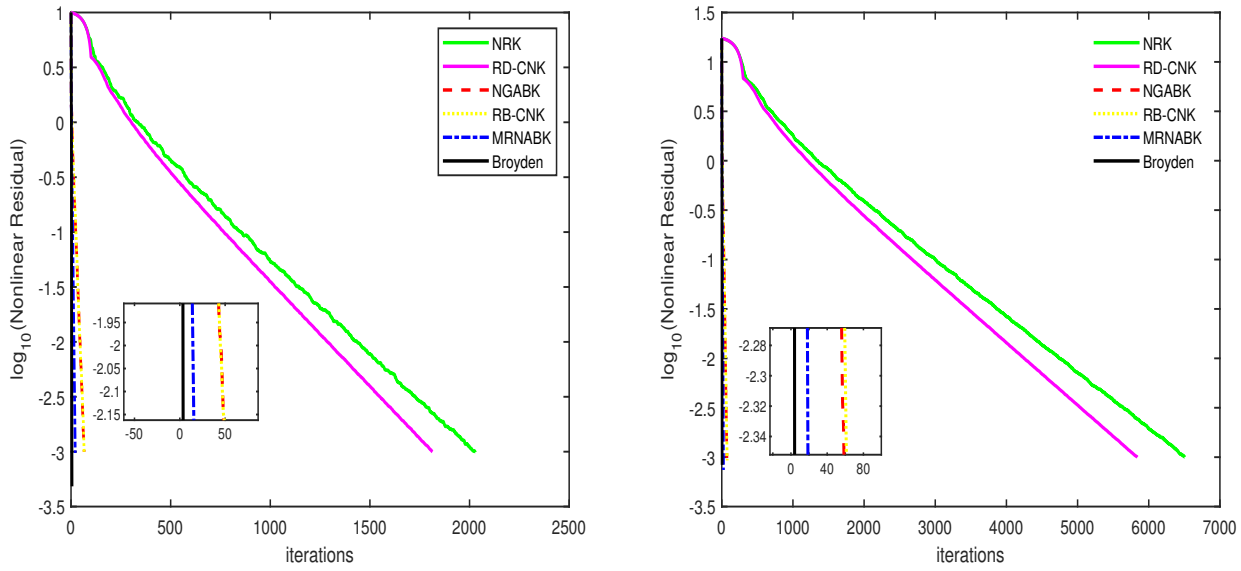
$m \times n$	Broyden	NRK	RD-CNK	**NGABK**	RB-CNK	**MRNABK**
50×50	**5**	970	864	70	62	21
100×100	**5**	2022	1814	66	66	21
300×300	**5**	6518	5838	72	76	24
500×500	**6**	11239	10027	78	81	24

Table 3. CPU comparison of Broyden method, NRK, RD-CNK, NGABK, RB-CNK, and MRNABK for the H-equation.

$m \times n$	Broyden	NRK	RD-CNK	**NGABK**	RB-CNK	**MRNABK**
50×50	**0.0092**	0.2646	0.5149	0.0640	0.0989	0.0593
100×100	**0.0201**	0.4751	1.6045	0.1064	0.1680	0.0754
300×300	**0.1404**	4.3069	26.9104	0.6370	0.8359	0.4770
500×500	**0.5794**	12.2161	95.7063	1.4577	1.9005	1.0813

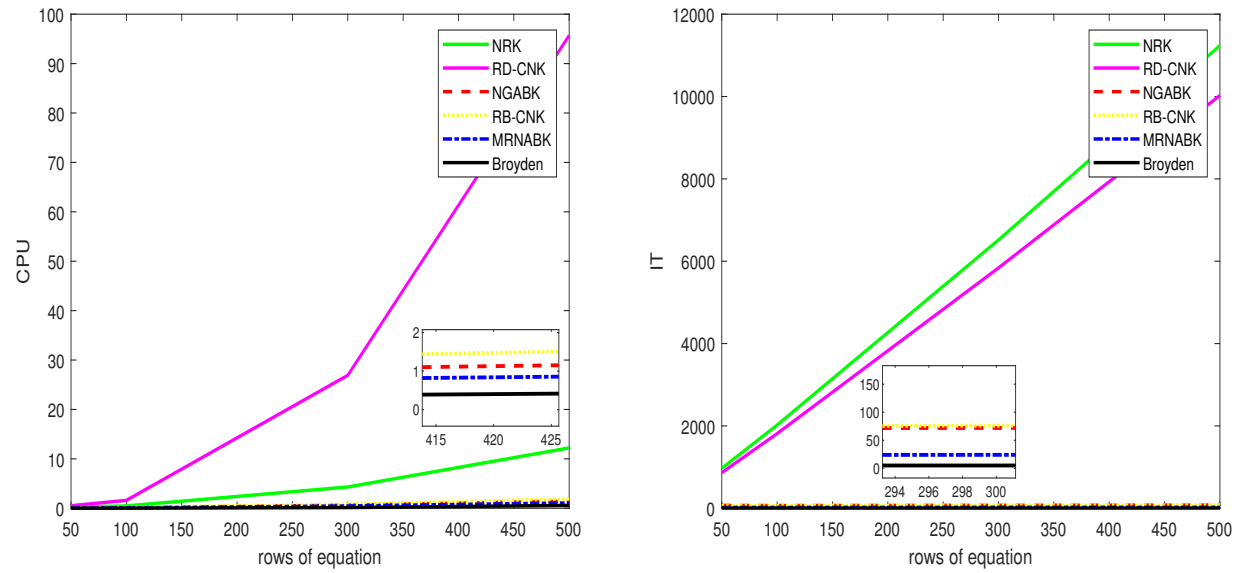
Table 4. IT and CPU comparison of MRNABK and NGABK with $\varrho = 0.1$ and $c = 0.9$ for the H-equation.

$m \times n$	MRNABK (IT)	NGABK (IT)	MRNABK (CPU)	NGABK (CPU)
50×50	21	70	0.0433	0.0742
100×100	21	66	0.1067	0.1631
500×500	24	78	1.6064	1.1954
1000×1000	25	78	3.8946	5.0264



(a) $x_0 = zero(100, 1)$ (b) $x_0 = zero(300, 1)$

Figure 1. The results of different methods for H-equation.



(a) CPU (b) IT

Figure 2. CPU (left) and IT (right) of different methods with different N for H-equation.

Example 2. In this example, we consider the Brown almost linear function [20],

$$f_k(\mathbf{x}) = \mathbf{x}^{(k)} + \sum_{i=1}^{n} \mathbf{x}^{(i)} - (n+1), \quad 1 \le k < n;$$
$$f_k(\mathbf{x}) = \left(\prod_{i=1}^{n} \mathbf{x}^{(i)}\right) - 1, \qquad k = n.$$

In this experiment, we set the initial value $\mathbf{x}_0 = 0.5 * ones(n, 1)$. The solution of this problem is $\mathbf{x}_* = (1, 1, ..., 1)^T$. The number of equations and the number of unknowns is set to 50×50, 100×100, 150×150, 200×200, 250×250, 300×300, 350×350, and 400×400. We list the computing time and iteration numbers of these methods, respectively, in Tables 5 and 6. The outcomes demonstrate how much better our new approaches perform than the NRK method. Table 6 shows that the NGABK method and the MRNABK method have almost identical iteration times, yet they are both superior to the RB-CNK method.

Table 5. IT comparison of NRK, RD-CNK, NGABK, RB-CNK, and MRNABK for the Brown almost linear function.

$m \times n$	NRK	RD-CNK	**NGABK**	RB-CNK	**MRNABK**
50×50	4660	755	**1**	**1**	**1**
100×100	15881	1308	**1**	**1**	**1**
150×150	35398	1904	**1**	**1**	**1**
200×200	58127	2506	**1**	**1**	**1**
250×250	85937	3128	**1**	**1**	**1**
300×300	116851	3750	**1**	**1**	**1**
350×350	156027	4372	**1**	**1**	**1**
400×400	196134	4992	**1**	**1**	**1**

Table 6. CPU comparison of NRK, RD-CNK, NGABK, RB-CNK, and MRNABK for the Brown almost linear function.

$m \times n$	NRK	RD-CNK	**NGABK**	RB-CNK	**MRNABK**
50×50	0.7222	0.2290	0.0024	0.0049	**0.0017**
100×100	2.5929	1.2108	0.0050	0.0063	**0.0045**
150×150	7.5696	2.7863	**0.0108**	0.0149	0.0112
200×200	15.4563	5.9908	**0.0187**	0.0250	0.0188
250×250	24.4239	11.2057	0.0321	0.0446	**0.0320**
300×300	35.7111	17.4006	0.0630	0.0737	**0.0530**
350×350	53.0089	25.8307	0.0508	0.0775	**0.0476**
400×400	71.1793	36.0714	**0.0626**	0.0982	0.0699

Example 3. In this example, we consider the Singular Broyden problem [12],

$$f_k(\mathbf{x}) = ((3 - 2\mathbf{x}_k)\mathbf{x}_k - 2\mathbf{x}_{k+1} + 1)^2, \qquad k = 1;$$
$$f_k(\mathbf{x}) = ((3 - 2x_k)\mathbf{x}_k - \mathbf{x}_{k-1} - 2\mathbf{x}_{k+1} + 1)^2, \quad 1 < k < n;$$
$$f_k(\mathbf{x}) = ((3 - 2\mathbf{x}_k)\mathbf{x}_k - \mathbf{x}_{k-1} + 1)^2, \qquad k = n.$$

In this experiment, we set the initial value $\mathbf{x}_0 = -0.5*ones(n, 1)$ and n is the number of the equations. The Singular Broyden problem [12] is a square nonlinear system of equations, and its Jacobian matrix is singular at the solution. First, we conducted an experiment on the values of parameter ϱ. From Table 7, we can see that for a fixed number of equations, the CPU of the MRNABK method was better when $\varrho \in [0.1, 0.3]$. So, we set $\varrho = 0.2$ in this example. Tables 8 and 9 demonstrate that, in terms of both computation time and iteration steps, the NGABK approach converged more quickly than the other three methods. The MRNABK method's residuals fell the fastest, whereas the NRK method's residuals declined the slowest, as seen in Figure 3. It is evident from Figure 4 that all five approaches may get the approximate results.

Table 7. CPU of MRNABK for the Singular Broyden problem with $\mathbf{x}_0 = (-0.5, -0.5, \ldots, -0.5)^T$ and different ϱ.

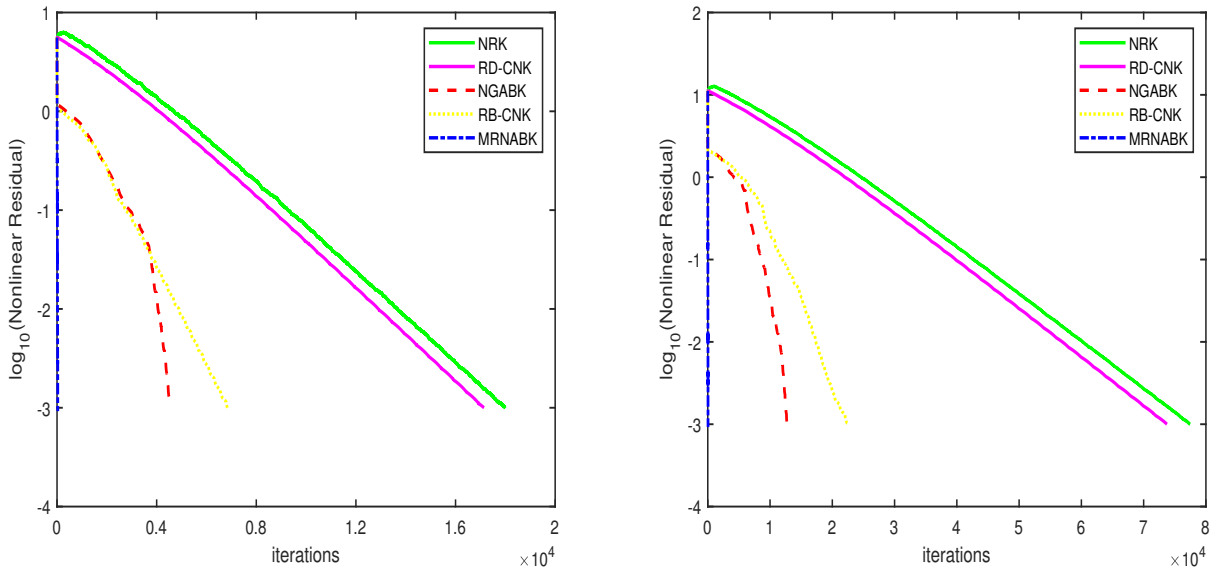
ϱ	0.1	0.2	0.3	0.4	0.5	0.6	0.7	0.8
$m = 100$	0.3300	**0.1458**	0.2690	0.2493	0.1607	0.1902	0.2775	0.4601
$m = 500$	1.1946	0.5796	**0.5683**	0.9989	3.0169	3.5255	4.1331	6.1652
$m = 1000$	1.9083	**1.5407**	1.5582	1.6074	8.0733	9.8453	15.4725	19.0392
$m = 2000$	5.3491	**5.0097**	5.0497	18.2991	24.9766	35.6343	83.4121	72.1692

Table 8. IT comparison of NRK, RD-CNK, NGABK, RB-CNK, and MRNABK for the Singular Broyden problem.

$m \times n$	NRK	RD-CNK	**NGABK**	RB-CNK	**MRNABK**
500×500	18026	17138	4531	6841	**31**
1000×1000	37385	35613	8807	13502	**37**
1500×1500	57360	54562	13502	22743	**34**
2000×2000	77442	73764	12756	22528	**42**

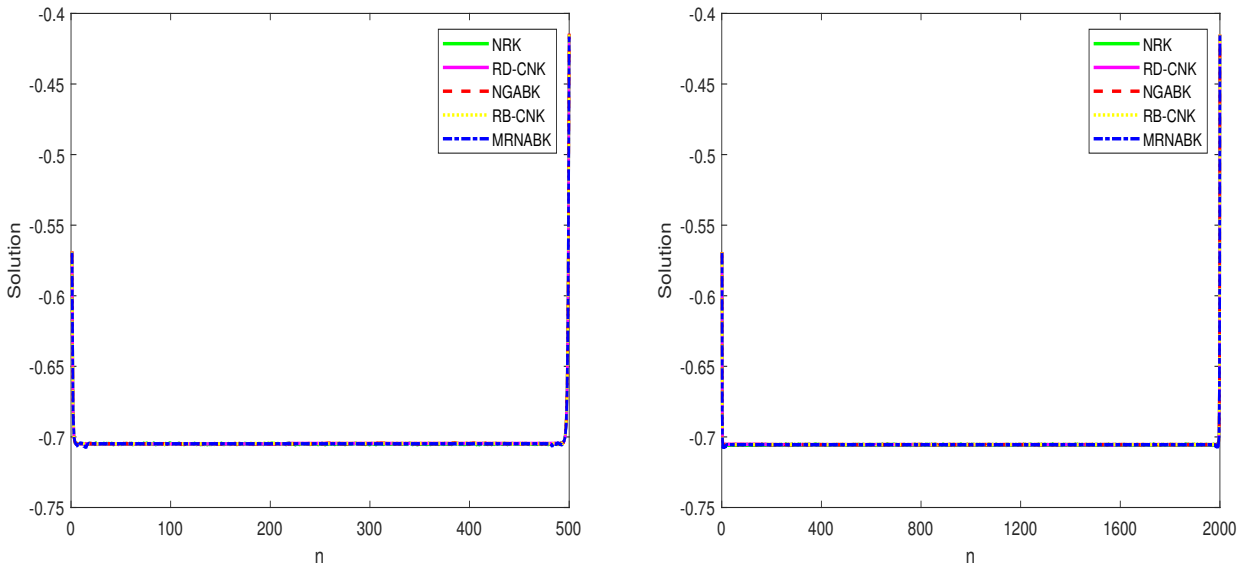
Table 9. CPU comparison of NRK, RD-CNK, NGABK, RB-CNK, and MRNABK for the Singular Broyden problem.

$m \times n$	NRK	RD-CNK	**NGABK**	RB-CNK	**MRNABK**
500×500	3.9115	7.7637	1.9330	3.8200	**0.6114**
1000×1000	12.2036	25.9053	6.0330	8.5914	**1.2652**
1500×1500	23.2143	47.4775	12.3048	19.0035	**0.1047**
2000×2000	38.2803	77.2462	19.5523	28.5388	**4.7409**



(a) $n = 500$ (b) $n = 2000$

Figure 3. The results of different methods for the Singular Broyden problem with $n = 500$ (left), 2000 (right).



(a) $n = 500$ (b) $n = 2000$

Figure 4. The results of the Singular Broyden problem with $n = 500$ (left), 2000 (right).

Example 4. Consider the following problem, which is a chained serpentine overdetermined

problem [17],

$$
\begin{aligned}
f_k(\mathbf{x}) &= 10\left(\frac{2\mathbf{x}_i}{(1+(\mathbf{x}_i)^2)^2} - \mathbf{x}_{i+1}\right), && mod(k,2) = 1,\\
f_k(\mathbf{x}) &= \mathbf{x}_i - 1, && mod(k,2) = 0,\\
m &= 2(n-1), && i = div(k+1,2),
\end{aligned}
$$

where m represents the number of equations. In this experiment, we set the initial value $\mathbf{x}_0 = 0.5 * ones(n, 1)$, and $n = 100, 300, 500, 1000, 2000$, respectively. From Table 10, we set $\varrho = 0.2$. As we can see, when compared to the other four approaches, the NGABK method produced better numerical results from Tables 11 and 12. Furthermore, we note that the NGABK method and the MRNABK method required fewer iterations as the dimension of overdetermined issues grew, and the NRK method's iteration time was nearly equal to that of the NGABK method. The Broyden method took too long when the problem's dimension grew. The Broyden method took more than 10 minutes to iterate when there were 1998 equations.

Table 10. CPU of MRNABK for the overdetermined nonlinear problem with $\mathbf{x}_0 = (0.5, 0.5, \ldots, 0.5)^T$ and different ϱ.

ϱ	0.1	0.2	0.3	0.4	0.5	0.6	0.7	0.8
$m = 200$	-	**0.0204**	0.0210	0.0621	0.0714	0.0564	0.0631	0.0206
$m = 1000$	0.3791	0.8691	0.9148	0.2926	0.2762	**0.2720**	0.2789	0.2865
$m = 1500$	1.2465	**1.0414**	1.0892	1.0592	1.0487	1.0503	1.0440	3.7909
$m = 2000$	2.6784	2.3893	2.3766	**2.3360**	2.3923	2.3536	2.4215	8.3677

Table 11. IT comparison of NRK, RD-CNK, NGABK, RB-CNK, and MRNABK for the overdetermined nonlinear problem with $\mathbf{x}_0 = (0.5, 0.5, \ldots, 0.5)^T$.

n	Broyden	NRK	RD-CNK	**NGABK**	RB-CNK	**MRNABK**
100	420	270	108	**33**	106	221
300	1180	769	827	**29**	344	742
500	1725	1336	2023	**20**	526	525
1000	-	2706	1857	**18**	1026	22
2000	-	5473	3137	19	2064	**18**

Table 12. CPU comparison of NRK, RD-CNK, NGABK, RB-CNK, and MRNABK for the overdetermined nonlinear problem with $\mathbf{x}_0 = (0.5, 0.5, \ldots, 0.5)^T$.

n	Broyden	NRK	RD-CNK	**NGABK**	RB-CNK	**MRNABK**
100	1.6608	0.0546	0.0517	**0.0312**	0.0998	0.1155
300	44.7083	0.1871	0.3519	**0.1786**	0.3565	1.5205
500	241.7063	0.2970	2.4785	**0.3102**	0.6332	1.3255
1000	-	1.3988	1.8885	**1.0953**	2.8976	1.7912
2000	-	**3.9915**	7.5685	4.0058	11.1190	4.1889

6. Conclusion

Based on the Gaussian Kaczmarz method and the RD-CNK method, we introduced a new class of nonlinear Kaczmarz block approaches to solve nonlinear equations and investigated their convergence theories. By employing an averaging technique, these methods avoid computing the Moore–Penrose pseudoinverse of the Jacobian matrix at each iteration, significantly reducing the computational cost. Experimental results demonstrated that the NGABK and MRNABK methods performed better in terms of CPU and IT than the NRK method (and other methods) for the singular Jacobian matrix issue and the overdetermined problem. For problems like the H-equation, several traditional methods, including the Broyden method, and the Newton method perform well, but our proposed methods also worked better than the other Kaczmarz methods. Additionally, selecting the ideal parameter ϱ was a crucial matter. We plan to keep working on the more significant research of the pseudoinverse-free approach and the more effective greedy rules in the future.

Use of AI tools declaration

The authors declare that they have not used Artificial Intelligence (AI) tools in the creation of this article.

Conflict of interest

The authors declare that there are no conflicts of interest.

References

1. H. B. An, Z. Z. Bai, Broyden method for nonlinear equation in several variables, *Mathematica Numerica Sinica (Chinese Journal)*, **26** (2004), 385–400.
2. H. B. An, Z. Z. Bai, Directional secant method for nonlinear equations, *J. Comput. Appl. Math.*, **175** (2005), 291–304. https://doi.org/10.1016/j.cam.2004.05.013
3. Z. Z. Bai, L. Wang, On convergence rates of Kaczmarz-type methods with different selection rules of working rows, *Appl. Numer. Math.*, **186** (2023), 289–319. https://doi.org/10.1016/j.apnum.2023.01.013
4. Z. Z. Bai, W. T. Wu, On greedy randomized Kaczmarz method for solving large sparse linear systems, *SIAM J. Sci. Comput.*, **40** (2018), A592–A606. https://doi.org/10.1137/17M1137747
5. J. Q. Chen, Z. D. Huang, On the error estimate of the randomized double block Kaczmarz method, *Appl. Math. Comput.*, **370** (2020), 124907. https://doi.org/10.1016/j.amc.2019.124907
6. Q. P. Chen, W. R. Hao, A homotopy training algorithm for fully connected neural networks, *P. Roy. Soc. A-Math. Phy.*, **475** (2019), 20190662. https://doi.org/10.1098/rspa.2019.0662
7. K. Du, W. T. Si, X. H. Sun, Randomized extended average block Kaczmarz for solving least squares, *SIAM J. Sci. Comput.*, **42** (2020), A3541–A3559. https://doi.org/10.1137/20M1312629

8. K. Du, X. H. Sun, *Pseudoinverse-free randomized block iterative algorithms for consistent and inconsistent linear systems*, arXiv: 2011.10353 [preprint], (2020), [cited 2024 March 27]. Available from: https://doi.org/10.48550/arXiv.2011.10353

9. K. Du, X. H. Sun, A doubly stochastic block Gauss–Seidel algorithm for solving linear equations, *Appl. Math. Comput.*, **408** (2021), 126373. https://doi.org/10.1016/j.amc.2021.126373

10. Y. C. Eldar, D. Needell, Acceleration of randomized Kaczmarz method via the Johnson-Lindenstrauss lemma, *Numer. Algorithms*, **58** (2011), 163–177. https://doi.org/10.1007/s11075-011-9451-z

11. T. Elfving, Block-iterative methods for consistent and inconsistent linear equations, *Numer. Math.*, **35** (1980), 1–12. https://doi.org/10.1007/BF01396365

12. M. A. Gomes-Ruggiero, J. M. Martínez, A. C. Moretti, Comparing algorithms for solving sparse nonlinear systems of equations, *SIAM J. Sci. Stat. Comput.*, **13** (1992), 459–483. https://doi.org/10.1137/0913025

13. R. M. Gower, P. Richtárik, Randomized iterative methods for linear systems, *SIAM J. Matrix Anal. A.*, **36** (2015), 1660–1690. https://doi.org/10.1137/15M1025487

14. S. Karczmarz, Angenaherte auflosung von systemen linearer glei-chungen, *Bull. Int. Acad. Pol. Sic. Let., Cl. Sci. Math. Nat.*, **35** (1937), 355–357.

15. K. Kawaguchi, Deep learning without poor local minima, In: D. D. Lee, M. Sugiyama, U. Luxburg, I. Guyon, R. Garnett, *Advances in neural information processing systems*, New York: Curran Associates Inc., **29** (2016), 586–594.

16. J. Liu, S. J. Wright, An accelerated randomized Kaczmarz algorithm, *Math. Comput.*, **85** (2016), 153–178. http://dx.doi.org/10.1090/mcom/2971

17. L. Lukšan, Hybrid methods for large sparse nonlinear least squares, *J. Optimiz. Theory App.*, **89** (1996), 575–595.

18. A. Ma, D. Needell, A. Ramdas, Convergence properties of the randomized extended Gauss–Seidel and Kaczmarz methods, *SIAM J. Matrix Anal. A.*, **36** (2015), 1590–1604. https://doi.org/10.1137/15M1014425

19. L. Mirsky, Symmetric gauge functions and unitarily invariant norms, *Q. J. Math.*, **11** (1960), 50–59. https://doi.org/10.1093/qmath/11.1.50

20. J. J. Moré, B. S. Garbow, K. E. Hillstrom, Testing unconstrained optimization software, *ACM T. Math. Software*, **7** (1981), 17–41.

21. M. S. Morshed, M. S. Islam, M. Noor-E-Alam, Accelerated sampling Kaczmarz Motzkin algorithm for the linear feasibility problem, *J. Global Optim.*, **77** (2020), 361–382. https://doi.org/10.1007/s10898-019-00850-6

22. I. Necoara, Faster randomized block Kaczmarz algorithms, *SIAM J. Matrix Anal. A.*, **40** (2019), 1425–1452. https://doi.org/10.1137/19M1251643

23. D. Needell, Randomized Kaczmarz solver for noisy linear systems, *BIT*, **50** (2010), 395–403. https://doi.org/10.1007/s10543-010-0265-5

24. D. Needell, J. A. Tropp, Paved with good intentions: Analysis of a randomized block Kaczmarz method, *Linear Algebra Appl.*, **441** (2014), 199–221. https://doi.org/10.1016/j.laa.2012.12.022

25. D. Needell, R. Zhao, A. Zouzias, Randomized block Kaczmarz method with projection for solving least squares, *Linear Algebra Appl*, **484** (2015), 322–343. https://doi.org/10.1016/j.laa.2015.06.027

26. J. M. Ortega, W. C. Rheinboldt, *Iterative Solution of Nonlinear Equations in Several Variables*, Philadelphia: Society for Industrial and Applied Mathematics, 2000. https://doi.org/10.1137/1.9780898719468

27. T. Strohmer, R. Vershynin, A randomized Kaczmarz algorithm with exponential convergence, *J. Fourier Anal. Appl.*, **15** (2009) 262–278. https://doi.org/10.1007/s00041-008-9030-4

28. Q. F. Wang, W. G. Li, W. D. Bao, X. Q. Gao, Nonlinear Kaczmarz algorithms and their convergence, *J. Comput. Appl. Math.*, **399** (2022), 113720. https://doi.org/10.1016/j.cam.2021.113720

29. X. Z. Wang, M. L. Che, C. X. Mo, Y. M. Wei, Solving the system of nonsingular tensor equations via randomized Kaczmarz-like method, *J. Comput. Appl. Math.*, **421** (2023), 114856. https://doi.org/10.1016/j.cam.2022.114856

30. A. Q. Xiao, J. F. Yin, N. Zheng, On fast greedy block Kaczmarz methods for solving large consistent linear systems, *Comput. Appl. Math.*, **42** (2023), 119. https://doi.org/10.1007/s40314-023-02232-x

31. R. Yuan, A. Lazaric, R. M. Gower, Sketched Newton-Raphson, *SIAM J. Optimiz.*, **32** (2022), 1555–1583. https://doi.org/10.1137/21M139788X

32. J. H. Zhang, Y. Q. Wang, J. Zhao, On maximum residual nonlinear Kaczmarz-type algorithms for large nonlinear systems of equations, *J. Comput. Appl. Math.*, 425 (2023), 115065. https://doi.org/10.1016/j.cam.2023.115065

33. Y. J. Zhang, H. Y. Li, *Greedy capped nonlinear Kaczmarz methods*, arXiv: 2210.00653 [preprint], (2022), [cited 2024 March 27]. Available from: https://doi.org/10.48550/arXiv.2210.00653

34. Y. J. Zhang, H. Y. Li, L. Tang, *Greedy randomized sampling nonlinear Kaczmarz methods*, arXiv: 2209.06082 [preprint], (2022), [cited 2024 March 27]. Available from: https://doi.org/10.48550/arXiv.2209.06082

35. A. Zouzias, N. M. Freris, Randomized extended Kaczmarz for solving least squares, *SIAM J. Matrix Anal. A.*, **34** (2013), 773–793. https://doi.org/10.1137/120889897

AIMS Press