Filomat 40:6 (2026), 2191–2215
https://doi.org/10.2298/FIL2606191Z

Published by Faculty of Sciences and Mathematics,
University of Niš, Serbia
Available at: http://www.pmf.ni.ac.rs/filomat

Structure preserving Kaczmarz methods based on relaxed greedy selection for solving quaternion linear systems

Yuegui Zheng[a], Xiaojian Ding[a], Baohua Huang[a,*]

[a] *School of Mathematics and Statistics, Key Laboratory of Analytical Mathematics and Applications (Ministry of Education), Fujian Normal University, Fuzhou 350117, P.R. China*

Abstract. In this paper, we study randomized Kaczmarz iteration method for the quaternion linear systems and propose a structure preserving relaxed greedy randomized Kaczmarz iteration (QRGRK) method. In order to accelerate the convergence, we utilize Polyak's momentum acceleration technique and present the structure preserving PmQRGRK method. We provide the asymptotic convergence theory for the proposed iterative methods and demonstrate that they converge to the exact solution in expectation. Numerical examples are presented to illustrate the effectiveness of the proposed QRGRK methods compared with the ME-RGRK and RGRK methods for the real linear systems arising from the quaternion linear systems. In addition, we demonstrate the numerical advantage of PmQRGRK method over QRGRK method in terms of iteration counts and computing time.

1. Introduction

Quaternion linear systems have a wide range of applications in many problems in science and engineering, such as quantum mechanics [5], signal processing [31], color image restoration [11, 37] and neural network [24]. Therefore, more and more researchers are interested in such problems and have made a lot of valuable achievements [6, 9, 19, 34].

Let $\mathbb{Q}$ and $\mathbb{Q}^{m\times n}$ denote the quaternion skew-field and all $m \times n$ matrices over the quaternion skew-field, respectively. Given a quaternion coefficient matrix $\mathbf{A} \in \mathbb{Q}^{m\times n}$ and a right-hand side quaternion vector $\mathbf{c} \in \mathbb{Q}^m$, we aim to find an unknown quaternion vector $\mathbf{x} \in \mathbb{Q}^n$ which solves the following linear systems over the quaternion skew-field

$$\mathbf{A}\mathbf{x} = \mathbf{c}. \tag{1}$$

2020 *Mathematics Subject Classification*. Primary 65F10; Secondary 97N40.

Keywords. Quaternion linear systems, Relaxed greedy selection, Randomized Kaczmarz iteration, Polyak's momentum acceleration, Structure preserving algorithm.

Received: 02 September 2024; Revised: 28 September 2025; Accepted: 09 January 2026

Communicated by Marko Petković

Research supported by National Natural Science Foundation of China (12001211, 12071159, 12171168) and Natural Science Foundation of Fujian Province, China (2022J01194).

* Corresponding author: Baohua Huang

Email addresses: zyg2328280760@163.com (Yuegui Zheng), xiaojianding2023@126.com (Xiaojian Ding), baohuahuang@fjnu.edu.cn (Baohua Huang)

ORCID iDs: https://orcid.org/0009-0008-8655-9126 (Yuegui Zheng), https://orcid.org/0009-0009-2748-0589 (Xiaojian Ding), https://orcid.org/0000-0002-5011-0801 (Baohua Huang)

In contrast to traditional linear systems over real or complex number field, there are a lot of challenges in dealing with the quaternion linear systems (1) because of quaternion algebra is associative but noncommutative. It is obvious that the quaternion linear systems (1) can be equivalently transformed into a real matrix equation

$$\mathscr{R}(\mathbf{A})\mathscr{R}(\mathbf{x}) = \mathscr{R}(\mathbf{c}), \tag{2}$$

where $\mathscr{R}(\cdot)$ is a linear homeomorphic mapping from quaternion matrices (or vectors) to their real counterparts. To solve the general (real) linear systems (2), iterative methods can be employed, for example, Krylov subspace methods [15, 27, 29], splitting iterative method [18, 32] and randomized Kaczmarz-type iteration method [3, 30]. However, the amount of computation and storage of the above mentioned methods for solving (2) are greatly increased with the increasing of the dimension since the dimension is expanded to four times the original dimension, e.g., $\mathbf{A} \in \mathbb{Q}^{m\times n}$, whereas $\mathscr{R}(\mathbf{A}) \in \mathbb{R}^{4m\times 4n}$.

For solving the consistent linear systems, the Kaczmarz method [14] is one of the most popular iterative algorithms due to its simplicity and efficiency. In order to improve the convergence of the Kaczmarz algorithm, in 2009, Strohmer and Vershynin [30] proposed a randomized Kaczmarz (RK) algorithm based on "linear convergence", which processes the coefficient matrix A in a random order rather than a certain order. For more details on the convergence principle of the RK method and the algorithm induction, see [1, 2, 20].

The RK method for a real linear system $Ax = c$, where $A \in \mathbb{R}^{m\times n}$ and $c \in \mathbb{R}^m$, is simple yet powerful iterative solvers [30]. At the k-th iterate, the RK method selects the row index i_k with the probability $\|A_{i_k,:}\|^2/\|A\|_F^2$, and then updates x^k by

$$x^{k+1} = x^k + \frac{c_{i_k} - A_{i_k,:}x^k}{\|A_{i_k,:}\|^2}(A_{i_k,:})^T. \tag{3}$$

To improve the convergence rate of RK method, Bai and Wu [3] established a greedy randomized Kaczmarz (GRK) algorithm which performs a greedy probability strategy to extract larger entries of the residual vector at each iteration. By introducing a relaxation parameter in the probability criterion, the GRK method is further generalized to relaxed greedy randomized Kaczmarz (RGRK) method which outperforms the GRK method in terms of both iteration numbers and running times if a suitable relaxation parameter is selected [4]. There are other extensions of RK method for the real linear systems, such as block and randomized variants and heavy ball momentum variants, see [16, 17, 21, 35, 40] and the references therein. In addition, the RK algorithm has been extended to solve the system of matrix equations, e.g., ME-RGRK [33], PmRGRK [36], NmRGRK [36], GRABK [22] and hierarchical matrix form of RK algorithm [28]. When Kaczmarz-type method is directly applied to solve the resulting real linear systems arising from quaternion linear systems, the storage and computational operations are costly. Consequently, we aim to develop a structure preserving quaternion Kaczmarz-type iterative algorithm, which inherits the algebraic symmetry of $\mathscr{R}(\mathbf{A})$.

The main contributions of this paper are given below.

- Based on the Petrov-Galerkin conditions and the relaxed greedy probability strategy, we propose a structure preserving quaternion relaxed greedy randomized Kaczmarz (QRGRK) method for solving the quaternion linear systems (1).
- Due to the advantages of momentum acceleration technology, we incorporate the Polyak's momentum acceleration technique into the structure preserving QRGRK method for improving the convergence behavior and study the structure preserving QRGRK method with Polyak's momentum (PmQRGRK).
- We provide the theoretical proof of convergence for the structure preserving QRGRK and PmQRGRK methods and demonstrate that they can converge to the exact solution in expectation.

This paper is organized as follows. In Section 2, we give some notations and review some useful definitions and lemmas. Based on the Petrov-Galerkin conditions and the relaxed greedy probability strategy, we propose a structure preserving quaternion relaxed greedy randomized Kaczmarz (QRGRK) method for solving the quaternion linear systems (1) and provide its convergence analysis in Section 3. In Section 4, we cooperate structure preserving QRGRK with Polyak's momentum acceleration and present the PmQRGRK method. In Section 5, numerical examples are provided to illustrate the convergence of the proposed QRGRK method and demonstrate that its performance is better than that of the traditional RGRK and ME-RGRK methods. In addition, the effectiveness of PmQRGRK is supported by some numerical results. Finally, we give some conclusions in Section 6.

2. Preliminaries

In this section, we give some notations and introduce some basic definitions and lemmas. A quaternion $\mathbf{p} \in \mathbb{Q}$ composed of a real part and three imaginary parts, typically represented in the following form

$$\mathbf{p} = p_0 + p_1\mathbf{i} + p_2\mathbf{j} + p_3\mathbf{k}, \tag{4}$$

where $p_0, p_1, p_2, p_3 \in \mathbb{R}$, and $\mathbf{i}, \mathbf{j}, \mathbf{k}$ are three imaginary units such that

$$\mathbf{i}^2 = \mathbf{j}^2 = \mathbf{k}^2 = -1, \mathbf{ij} = -\mathbf{ji} = \mathbf{k}, \mathbf{jk} = -\mathbf{kj} = \mathbf{i}, \mathbf{ki} = -\mathbf{ik} = \mathbf{j}.$$

For a quaternion $\mathbf{p} \in \mathbb{Q}$ of the form (4), the scalar (real) part of is denoted by $\Re(\mathbf{p})$, whereas the vector part $\Im(\mathbf{p}) = p_1\mathbf{i} + p_2\mathbf{j} + p_3\mathbf{k}$ comprises three imaginary parts. The quaternion product of $\mathbf{p}$ and $\mathbf{q}$ is given by

$$\begin{aligned}\mathbf{pq} \ &= \ (p_0q_0 - p_1q_1 - p_2q_2 - p_3q_3) + (p_0q_1 + p_1q_0 + p_2q_3 - p_3q_2)\mathbf{i} \\ &\quad +(p_0q_2 - p_1q_3 + p_2q_0 + p_3q_1)\mathbf{j} + (p_0q_3 + p_1q_2 - p_2q_1 + p_3q_0)\mathbf{k}.\end{aligned}$$

The conjugate of a quaternion $\mathbf{p} \in \mathbb{Q}$ is defined by $\mathbf{p}^* = p_0 - p_1\mathbf{i} - p_2\mathbf{j} - p_3\mathbf{k}$, while the conjugate of the product satisfies $(\mathbf{pq})^* = \mathbf{q}^*\mathbf{p}^*$. The modulus of a quaternion $\mathbf{p} \in \mathbb{Q}$ is defined as $|\mathbf{p}| = \sqrt{p_0^2 + p_1^2 + p_2^2 + p_3^2}$ and $|\mathbf{pq}| = |\mathbf{p}||\mathbf{q}|$. The inverse of a quaternion $\mathbf{p} \neq \mathbf{0}$ is $\mathbf{p}^{-1} = \mathbf{p}^*/|\mathbf{p}|^2$. For any matrix $\mathbf{A} \in \mathbb{Q}^{m\times n}$, let $\mathbf{A}^T$, $\overline{\mathbf{A}}$, $\mathbf{A}^*$, $\mathbf{A}^\dagger$, $\sigma_{\max}(\mathbf{A})$ and $\sigma_{\min}(\mathbf{A})$ denote the transpose, the conjugate, the conjugate transpose, the Moore-Penrose inverse, the largest and the smallest nonzero singular values of the matrix $\mathbf{A}$, respectively. The column space and row space of $\mathbf{A}$ are respectively denoted as $\mathrm{Range}(\mathbf{A}) = \{\mathbf{v} : \mathbf{v} = \mathbf{Au}, \mathbf{u} \in \mathbb{Q}^{\mathrm{n}}\} \subseteq \mathbb{Q}^{\mathrm{m}}$ and $\mathrm{Null}(\mathbf{A}) = \{\mathbf{u} \in \mathbb{Q}^{\mathrm{n}} : \mathbf{Au} = 0\} \subseteq \mathbb{Q}^{\mathrm{n}}$. $I^{(n)}$ stands for the identity matrix of order n. If $\mathbf{A}$ is a square matrix, then $\mathrm{tr}(\mathbf{A})$ denotes the trace of $\mathbf{A}$. For a positive integer m, let $[m] = \{1, \cdots, m\}$. The symbol $\mathbb{E}_k[\cdot]$ denote the expected value conditional on the first k iterations, i.e.

$$\mathbb{E}_k[\cdot] = \mathbb{E}[\cdot|i_0, i_1, \cdots, i_{k-1}],$$

where $i_j (j = 0, 1, \cdots, k-1)$ is the jth row selected at the jth iterate. Then, from the law of iterated expectations, we have $\mathbb{E}[\mathbb{E}_k[\cdot]] = \mathbb{E}[\cdot]$.

The inner product of two quaternion vectors $\mathbf{x}, \mathbf{y} \in \mathbb{Q}^n$ is defined as [7]

$$\langle \mathbf{x}, \mathbf{y}\rangle = \mathbf{y}^*\mathbf{x} = \sum_{i=1}^{n} \mathbf{y}_i^*\mathbf{x}_i. \tag{5}$$

The induced-norm of a vector $\mathbf{x}$ is of the form

$$\|\mathbf{x}\| = \sqrt{\langle \mathbf{x}, \mathbf{x}\rangle} = \sqrt{\mathbf{x}^*\mathbf{x}} = \sqrt{\mathbf{x}^T\overline{\mathbf{x}}} = \sqrt{\sum_{i=1}^{n} |\mathbf{x}_i|^2}.$$

It is seen that $\mathbb{Q}^n$ is a right quaternionic Hilbert space with the inner product in (5) and $\langle \mathbf{x}\boldsymbol{\alpha}_1 + \mathbf{y}\boldsymbol{\alpha}_2, \mathbf{z}\rangle = \langle \mathbf{x}, \mathbf{z}\rangle\boldsymbol{\alpha}_1 + \langle \mathbf{y}, \mathbf{z}\rangle\boldsymbol{\alpha}_2$ for $\mathbf{x}, \mathbf{y}, \mathbf{z} \in \mathbb{Q}^n$ and $\boldsymbol{\alpha}_1, \boldsymbol{\alpha}_2 \in \mathbb{Q}$ [7]. Direct calculations give

$$\langle \mathbf{x} + \mathbf{y}, \mathbf{x} + \mathbf{y}\rangle = \|\mathbf{x}\|^2 + \langle \mathbf{x}, \mathbf{y}\rangle + \langle \mathbf{y}, \mathbf{x}\rangle + \|\mathbf{y}\|^2 \geq 0.$$

This implies that

$$-\langle \mathbf{x}, \mathbf{y} \rangle - \langle \mathbf{y}, \mathbf{x} \rangle \leq \|\mathbf{x}\|^2 + \|\mathbf{y}\|^2.$$

The inner product of two quaternion matrices $\mathbf{A}, \mathbf{B} \in \mathbb{Q}^{m\times n}$ is defined by [7]

$$\langle \mathbf{A}, \mathbf{B} \rangle = \mathrm{tr}(\mathbf{B}^*\mathbf{A}).$$

The induced-norm of a matrix $\mathbf{A}$ is of the form

$$\|\mathbf{A}\|_{\mathrm{F}} = \sqrt{\langle \mathbf{A}, \mathbf{A} \rangle} = \sqrt{\mathrm{tr}(\mathbf{A}^*\mathbf{A})} = \sqrt{\sum_{i=1}^{m}\sum_{j=1}^{n} |\mathbf{a}_{ij}|^2}.$$

The R-product of two quaternion vectors $\mathbf{x}, \mathbf{y} \in \mathbb{Q}^n$ is defined as [25]

$$\mathbf{x} \cdot \mathbf{y} = \Re(\mathbf{y}^*\mathbf{x}) = \Re(\mathbf{x}^*\mathbf{y}).$$

Let $f : \mathbb{Q}^n \to \mathbb{R}$. We say f is differentiable at $\mathbf{x} = x_0 + x_1\mathbf{i} + x_2\mathbf{j} + x_3\mathbf{k}$ if $\dfrac{\partial f}{\partial x_i}$ exists at x_i for $i = 0, 1, 2, 3$, and we denote

$$\nabla f(\mathbf{x}) = \frac{\partial f}{\partial x_0} + \frac{\partial f}{\partial x_1}\mathbf{i} + \frac{\partial f}{\partial x_2}\mathbf{j} + \frac{\partial f}{\partial x_3}\mathbf{k}. \tag{6}$$

The following quaternion singular value decomposition (QSVD) is firstly provided in [38, Theorem 7.2].

Lemma 2.1 ([38]). *For a quaternion matrix $\mathbf{X} \in \mathbb{Q}^{m\times n}$, there exist two unitary quaternion matrices $\mathbf{U} \in \mathbb{Q}^{m\times m}$ and $\mathbf{V} \in \mathbb{Q}^{n\times n}$ such that*

$$\mathbf{X} = \mathbf{U}\begin{pmatrix} \Sigma_r & 0 \\ 0 & 0 \end{pmatrix}\mathbf{V}^*, \tag{7}$$

where $\Sigma_r = \mathrm{diag}(\sigma_1, \cdots, \sigma_r)$, $r \leq \min\{m, n\}$ and $\sigma_i (i = 1, 2, \cdots, r)$ are the nonzero singular values of $\mathbf{X}$. The set with all $\mathbf{U}$ and $\mathbf{V}$ satisfying the equation (7) is called the left singular matrix space and right singular matrix space.

By using the similar proof technique as that in [30, 39], we obtain the following result.

Lemma 2.2. *For a quaternion matrix $\mathbf{A} \in \mathbb{Q}^{m\times n}$ and a quaternion vector $\mathbf{x} \in \mathrm{Range}(\mathbf{A}^*)$, it holds*

$$\|\mathbf{A}\mathbf{x}\| \geq \sigma_{\min}(\mathbf{A})\|\mathbf{x}\| = \frac{\|\mathbf{A}\|_{\mathrm{F}}}{\kappa_{\mathrm{F},2}(\mathbf{A})}\|\mathbf{x}\|,$$

where $\kappa_{\mathrm{F},2}(\mathbf{A}) = \|\mathbf{A}\|_{\mathrm{F}}/\sigma_{\min}(\mathbf{A})$.

Proof. For a quaternion matrix $\mathbf{A}$, consider its singular value decomposition $\mathbf{A} = \mathbf{U}\Sigma\mathbf{V}^*$, where $\mathbf{U}^*\mathbf{U} = I^{(m)}$, $\mathbf{V}^*\mathbf{V} = I^{(n)}$, $\Sigma = \begin{pmatrix} \Sigma_r & 0 \\ 0 & 0 \end{pmatrix}$, $\Sigma_r = \mathrm{diag}(\sigma_1, \cdots, \sigma_r)$, $r \leq \min\{m, n\}$ and $\sigma_i (i = 1, 2, \cdots, r)$ are the nonzero singular values of $\mathbf{A}$.

If a quaternion vector $\mathbf{x} \in \mathrm{Range}(\mathbf{A}^*)$, then there exists $\mathbf{y} \in \mathbb{Q}^m$ such that

$$\mathbf{x} = \mathbf{A}^*\mathbf{y} = \mathbf{V}\Sigma^T\mathbf{U}^*\mathbf{y}.$$

This gives

$$\|\mathbf{x}\|^2 = \mathbf{y}^*\mathbf{U}\Sigma\mathbf{V}^*\mathbf{V}\Sigma^T\mathbf{U}^*\mathbf{y} = \mathbf{s}^*\Sigma\Sigma^T\mathbf{s} = \sum_{i=1}^{r} \sigma_i^2 |\mathbf{s}_i|^2,$$

where $\mathbf{s} = (\mathbf{s}_1, \mathbf{s}_2, \cdots, \mathbf{s}_m)^T = \mathbf{U}^*\mathbf{y} \in \mathbb{Q}^m$. Then

$$\begin{aligned}
\|\mathbf{A}\mathbf{x}\|^2 &= \|\mathbf{A}\mathbf{A}^*\mathbf{y}\|^2 = \|\mathbf{U}\Sigma\mathbf{V}^*\mathbf{V}\Sigma^T\mathbf{U}^*\mathbf{y}\|^2 = \mathbf{s}^*(\Sigma\Sigma^T)^2\mathbf{s} = \sum_{i=1}^{r}\sigma_i^4|\mathbf{s}_i|^2 \\
&\geq \sigma_{\min}^2(\mathbf{A})\sum_{i=1}^{r}\sigma_i^2|\mathbf{s}_i|^2 = \sigma_{\min}^2(\mathbf{A})\|\mathbf{x}\|^2 = \frac{\|\mathbf{A}\|_{\mathrm{F}}^2}{\kappa_{\mathrm{F},2}^2(\mathbf{A})}\|\mathbf{x}\|^2.
\end{aligned}$$

The proof is completed. □

Lemma 2.3 ([8]). *For a given quaternion matrix $\mathbf{A} \in \mathbb{Q}^{m\times n}$ and a quaternion vector $\mathbf{c} \in \mathbb{Q}^m$, the quaternion linear systems $\mathbf{A}\mathbf{x} = \mathbf{c}$ is consistent if and only if $\mathbf{A}\mathbf{A}^\dagger\mathbf{c} = \mathbf{c}$. In this case, its general solution can be expressed as*

$$\mathbf{x} = \mathbf{A}^\dagger\mathbf{c} + \mathbf{y} - \mathbf{A}^\dagger\mathbf{A}\mathbf{y},$$

where $\mathbf{y} \in \mathbb{Q}^n$ is an arbitrary quaternion vector.

According to the definition of inner product and the property of Moore-Penrose inverse, we have

$$\begin{aligned}
&\langle\mathbf{A}^\dagger\mathbf{c}, \mathbf{y} - \mathbf{A}^\dagger\mathbf{A}\mathbf{y}\rangle = \langle\mathbf{A}^\dagger\mathbf{c}, \mathbf{y}\rangle - \langle\mathbf{A}^\dagger\mathbf{c}, \mathbf{A}^\dagger\mathbf{A}\mathbf{y}\rangle = \langle\mathbf{A}^\dagger\mathbf{c}, \mathbf{y}\rangle - \langle(\mathbf{A}^\dagger\mathbf{A})^*\mathbf{A}^\dagger\mathbf{c}, \mathbf{y}\rangle \\
= \; &\langle\mathbf{A}^\dagger\mathbf{c}, \mathbf{y}\rangle - \langle\mathbf{A}^\dagger\mathbf{A}\mathbf{A}^\dagger\mathbf{c}, \mathbf{y}\rangle = \langle\mathbf{A}^\dagger\mathbf{c}, \mathbf{y}\rangle - \langle\mathbf{A}^\dagger\mathbf{c}, \mathbf{y}\rangle = 0.
\end{aligned}$$

Then

$$\|\mathbf{x}\|^2 = \|\mathbf{A}^\dagger\mathbf{c} + \mathbf{y} - \mathbf{A}^\dagger\mathbf{A}\mathbf{y}\|^2 = \|\mathbf{A}^\dagger\mathbf{c}\|^2 + \|\mathbf{y} - \mathbf{A}^\dagger\mathbf{A}\mathbf{y}\|^2.$$

This implies that $\mathbf{A}^\dagger\mathbf{c}$ is the least norm solution of the quaternion linear systems (1). In the following, we always assume that the quaternion linear systems (1) is consistent and $\mathbf{x}_*$ is an arbitrary solution of (1).

For a quaternion matrix $\mathbf{A} = A_0 + A_1\mathbf{i} + A_2\mathbf{j} + A_3\mathbf{k} \in \mathbb{Q}^{m\times n}$, its real counterpart is of the form [12, 13]

$$\mathscr{R}(\mathbf{A}) = \begin{pmatrix} A_0 & A_2 & A_1 & A_3 \\ -A_2 & A_0 & A_3 & -A_1 \\ -A_1 & -A_3 & A_0 & A_2 \\ -A_3 & A_1 & -A_2 & A_0 \end{pmatrix} \in \mathbb{R}^{4m\times 4n} \tag{8}$$

and the first block column of $\mathscr{R}(\mathbf{A})$ is denoted by

$$\mathscr{R}(\mathbf{A})_c = \begin{pmatrix} A_0 \\ -A_2 \\ -A_1 \\ -A_3 \end{pmatrix} \in \mathbb{R}^{4m\times n}. \tag{9}$$

Assume that

$$J_n = \begin{pmatrix} 0 & 0 & -I_n & 0 \\ 0 & 0 & 0 & -I_n \\ I_n & 0 & 0 & 0 \\ 0 & I_n & 0 & 0 \end{pmatrix}, R_n = \begin{pmatrix} 0 & -I_n & 0 & 0 \\ I_n & 0 & 0 & 0 \\ 0 & 0 & 0 & I_n \\ 0 & 0 & -I_n & 0 \end{pmatrix}, S_n = \begin{pmatrix} 0 & 0 & 0 & -I_n \\ 0 & 0 & I_n & 0 \\ 0 & -I_n & 0 & 0 \\ I_n & 0 & 0 & 0 \end{pmatrix}.$$

Then a real matrix $M \in \mathbb{R}^{4m\times 4n}$ is called JRS-symmetric if $J_mMJ_n^T = M$, $R_mMR_n^T = M$ and $S_mMS_n^T = M$ [13]. It is easy to see that a matrix $M \in \mathbb{R}^{4m\times 4n}$ is JRS-symmetric if and only if M is a real counterpart of a quaternion matrix [13]. Thus we define the inverse mapping of $\mathscr{R}$ to the JRS-symmetric matrix by $\mathscr{R}^{-1}(\mathscr{R}(\mathbf{A})) = \mathbf{A}$.

3. The QRGRK method

In this section, we give the Petrov-Galerkin conditions on the quaternion skew-field and present the structure preserving QRGRK method. The proposed method has higher computational and storage efficiency since it is only necessary to store or overwrite four components of quaternion matrices/vectors, rather than their real counterparts. Theoretically, the JRS-symmetry of the real counterpart is inherited.

Assume that Γ and Λ are the search subspace in $\mathbb{Q}^n$ and the constrained subspace in $\mathbb{R}^m$, respectively. The approximate solution $\mathbf{x}^{k+1}$ to (1) is found by imposing two constraints: it belongs to the affine space $\mathbf{x}^k + \Gamma$, and the residual vector $\mathbf{r}^{k+1} = \mathbf{c} - \mathbf{A}\mathbf{x}^{k+1}$ is orthogonal to the constrained subspace Λ with respect to the inner product. In other words,

$$\mathbf{x}^{k+1} \in \mathbf{x}^k + \Gamma \quad \text{and} \quad \mathbf{c} - \mathbf{A}\mathbf{x}^{k+1} \perp \Lambda. \tag{10}$$

Denote $\boldsymbol{\alpha}_i^T$ the ith rows of $\mathbf{A}$ and e_i the ith columns of $I^{(m)}$ for $i = 1, 2, \cdots, m$. We select two subspaces as follows

$$\Gamma = \mathrm{span}\{\bar{\boldsymbol{\alpha}}_i\}, \Lambda = \mathrm{span}\{e_i\},$$

where the index $i \in [m]$. Then the next iterate can be written as

$$\mathbf{x}^{k+1} = \mathbf{x}^k + \bar{\boldsymbol{\alpha}}_i \boldsymbol{\lambda} \tag{11}$$

for a quaternion $\boldsymbol{\lambda} \in \mathbb{Q}$. Using the orthogonality condition in (10), we have

$$\langle \mathbf{c} - \mathbf{A}\mathbf{x}^{k+1}, e_i \rangle = 0. \tag{12}$$

Substituting (11) into (12) yields

$$0 = \langle \mathbf{c} - \mathbf{A}\mathbf{x}^k - \mathbf{A}\bar{\boldsymbol{\alpha}}_i\boldsymbol{\lambda}, e_i \rangle = \langle \mathbf{c} - \mathbf{A}\mathbf{x}^k, e_i \rangle - \langle \mathbf{A}\bar{\boldsymbol{\alpha}}_i\boldsymbol{\lambda}, e_i \rangle = \langle \mathbf{c} - \mathbf{A}\mathbf{x}^k, e_i \rangle - \langle \mathbf{A}\bar{\boldsymbol{\alpha}}_i, e_i \rangle \boldsymbol{\lambda}.$$

This means that

$$\boldsymbol{\lambda} = \langle \mathbf{A}\bar{\boldsymbol{\alpha}}_i, e_i \rangle^{-1} \langle \mathbf{c} - \mathbf{A}\mathbf{x}^k, e_i \rangle = (e_i^T \mathbf{A}\bar{\boldsymbol{\alpha}}_i)^{-1} [e_i^T (\mathbf{c} - \mathbf{A}\mathbf{x}^k)] = \frac{\mathbf{c}_i - \boldsymbol{\alpha}_i^T \mathbf{x}^k}{\|\boldsymbol{\alpha}_i\|^2},$$

where $\mathbf{c}_i$ is the ith component of $\mathbf{c} \in \mathbb{Q}^m$. Thus we obtain the Kaczmarz iterative scheme for the quaternion linear systems (1) as follows

$$\mathbf{x}^{k+1} = \mathbf{x}^k + \bar{\boldsymbol{\alpha}}_i \frac{\mathbf{c}_i - \boldsymbol{\alpha}_i^T \mathbf{x}^k}{\|\boldsymbol{\alpha}_i\|^2}, \tag{13}$$

where the index $i \in [m]$. We see from (13) that the $(k+1)$th approximate solution $\mathbf{x}^{k+1}$ is the projection of the current iterate $\mathbf{x}^k$ onto the solution space

$$\mathbb{S}_i = \{\mathbf{x} \in \mathbb{Q}^n : \boldsymbol{\alpha}_i^T \mathbf{x} = \mathbf{c}_i, i \in [m]\}.$$

Define the following real value function of the quaternion vector variable

$$f_i(\mathbf{x}) = \frac{|\mathbf{c}_i - \boldsymbol{\alpha}_i^T \mathbf{x}|^2}{\|\boldsymbol{\alpha}_i\|^2}. \tag{14}$$

For $\mathbf{\Delta x} \in \mathbb{Q}^n$, we have

$$\begin{aligned} f_i(\mathbf{x} + \mathbf{\Delta x}) - f_i(\mathbf{x}) &= \frac{|\mathbf{c}_i - \boldsymbol{\alpha}_i^T(\mathbf{x} + \mathbf{\Delta x})|^2}{\|\boldsymbol{\alpha}_i\|^2} - \frac{|\mathbf{c}_i - \boldsymbol{\alpha}_i^T\mathbf{x}|^2}{\|\boldsymbol{\alpha}_i\|^2} \\ &= \frac{|(\mathbf{c}_i - \boldsymbol{\alpha}_i^T\mathbf{x}) - \boldsymbol{\alpha}_i^T\mathbf{\Delta x}|^2}{\|\boldsymbol{\alpha}_i\|^2} - \frac{|\mathbf{c}_i - \boldsymbol{\alpha}_i^T\mathbf{x}|^2}{\|\boldsymbol{\alpha}_i\|^2} \\ &= \frac{\Re[-(\mathbf{c}_i - \boldsymbol{\alpha}_i^T\mathbf{x})^*(\boldsymbol{\alpha}_i^T\mathbf{\Delta x})] + |\boldsymbol{\alpha}_i^T\mathbf{\Delta x}|^2}{\|\boldsymbol{\alpha}_i\|^2} \\ &= \frac{\Re[-(\bar{\boldsymbol{\alpha}}_i(\mathbf{c}_i - \boldsymbol{\alpha}_i^T\mathbf{x}))^*\mathbf{\Delta x}]}{\|\boldsymbol{\alpha}_i\|^2} + o(\|\mathbf{\Delta x}\|) \\ &= -\bar{\boldsymbol{\alpha}}_i\frac{\mathbf{c}_i - \boldsymbol{\alpha}_i^T\mathbf{x}^k}{\|\boldsymbol{\alpha}_i\|^2} \cdot \mathbf{\Delta x} + o(\|\mathbf{\Delta x}\|). \end{aligned}$$

It then follows from Proposition 4.2 in [25] that

$$\nabla f_i(\mathbf{x}) = -\bar{\boldsymbol{\alpha}}_i\frac{\mathbf{c}_i - \boldsymbol{\alpha}_i^T\mathbf{x}^k}{\|\boldsymbol{\alpha}_i\|^2}.$$

Thus the $(k+1)$th approximate solution $\mathbf{x}^{k+1}$ in (13) can be thought of as an exact linear search along the negative gradient direction from the current iterate $\mathbf{x}^k$.

Note that $\mathbf{c} = \mathbf{A}\mathbf{x}_*$ and $\mathbf{c}_i = \boldsymbol{\alpha}_i^T\mathbf{x}_*$. By the relation (13), we have

$$\mathbf{x}^{k+1} = \mathbf{x}^k + \frac{\bar{\boldsymbol{\alpha}}_i\boldsymbol{\alpha}_i^T}{\|\boldsymbol{\alpha}_i\|^2}(\mathbf{x}_* - \mathbf{x}^k). \tag{15}$$

Denote the kth error matrix by $\mathbf{h}^k = \mathbf{x}^k - \mathbf{x}_*$. We immediately have

$$\mathbf{h}^{k+1} = \mathbf{h}^k - \frac{\bar{\boldsymbol{\alpha}}_i\boldsymbol{\alpha}_i^T}{\|\boldsymbol{\alpha}_i\|^2}(\mathbf{x}^k - \mathbf{x}_*) = \mathbf{h}^k - \frac{\bar{\boldsymbol{\alpha}}_i\boldsymbol{\alpha}_i^T}{\|\boldsymbol{\alpha}_i\|^2}\mathbf{h}^k.$$

Then

$$\begin{aligned} \|\mathbf{h}^{k+1}\|^2 &= \left\langle \mathbf{h}^k - \frac{\bar{\boldsymbol{\alpha}}_i\boldsymbol{\alpha}_i^T}{\|\boldsymbol{\alpha}_i\|^2}\mathbf{h}^k, \mathbf{h}^k - \frac{\bar{\boldsymbol{\alpha}}_i\boldsymbol{\alpha}_i^T}{\|\boldsymbol{\alpha}_i\|^2}\mathbf{h}^k \right\rangle \\ &= \langle \mathbf{h}^k, \mathbf{h}^k\rangle - \left\langle \frac{\bar{\boldsymbol{\alpha}}_i\boldsymbol{\alpha}_i^T}{\|\boldsymbol{\alpha}_i\|^2}\mathbf{h}^k, \mathbf{h}^k \right\rangle - \left\langle \mathbf{h}^k, \frac{\bar{\boldsymbol{\alpha}}_i\boldsymbol{\alpha}_i^T}{\|\boldsymbol{\alpha}_i\|^2}\mathbf{h}^k \right\rangle + \left\langle \frac{\bar{\boldsymbol{\alpha}}_i\boldsymbol{\alpha}_i^T}{\|\boldsymbol{\alpha}_i\|^2}\mathbf{h}^k, \frac{\bar{\boldsymbol{\alpha}}_i\boldsymbol{\alpha}_i^T}{\|\boldsymbol{\alpha}_i\|^2}\mathbf{h}^k \right\rangle \\ &= \|\mathbf{h}^k\|^2 - (\mathbf{h}^k)^*\frac{\bar{\boldsymbol{\alpha}}_i\boldsymbol{\alpha}_i^T}{\|\boldsymbol{\alpha}_i\|^2}\mathbf{h}^k - (\mathbf{h}^k)^*\frac{\bar{\boldsymbol{\alpha}}_i\boldsymbol{\alpha}_i^T}{\|\boldsymbol{\alpha}_i\|^2}\mathbf{h}^k + (\mathbf{h}^k)^*\frac{\bar{\boldsymbol{\alpha}}_i\boldsymbol{\alpha}_i^T\bar{\boldsymbol{\alpha}}_i\boldsymbol{\alpha}_i^T}{\|\boldsymbol{\alpha}_i\|^2\|\boldsymbol{\alpha}_i\|^2}\mathbf{h}^k \\ &= \|\mathbf{h}^k\|^2 - \frac{(\boldsymbol{\alpha}_i^T\mathbf{h}^k)^*(\boldsymbol{\alpha}_i^T\mathbf{h}^k)}{\|\boldsymbol{\alpha}_i\|^2} \\ &= \|\mathbf{h}^k\|^2 - \frac{|\mathbf{c}_i - \boldsymbol{\alpha}_i^T\mathbf{x}^k|^2}{\|\boldsymbol{\alpha}_i\|^2} = \|\mathbf{h}^k\|^2 - w_i(\mathbf{x}^k), \end{aligned} \tag{16}$$

where

$$w_i(\mathbf{x}^k) = \frac{|\mathbf{c}_i - \boldsymbol{\alpha}_i^T\mathbf{x}^k|^2}{\|\boldsymbol{\alpha}_i\|^2}.$$

Formula (16) inspires us to choose index $i \in [m]$ to make the corresponding loss as large as possible.

Next, we present our structure preserving quaternion Kaczmarz method for the quaternion linear systems (1) by using the relaxed greedy selection strategies as in [3, 33].

Let the row index $i \in [m]$ be chosen with probability

$$\mathbb{P}(\text{Row} = i) = \frac{\|\boldsymbol{\alpha}_i\|^2}{\|\mathbf{A}\|_F^2}.$$

The main idea of the relaxed greedy selection strategy is to choose an index i_k in a compromise way, whose loss is convex combination of mean and largest ones. That is, we select a relaxation parameter $\theta \in [0, 1]$ and compute

$$w_{i_k}(\mathbf{x}^k) = \theta \max_{i\in[m]} \left\{ w_i(\mathbf{x}^k) \right\} + (1-\theta)\mathbb{E}[w_i(\mathbf{x}^k)],$$

where

$$\mathbb{E}[w_i(\mathbf{x}^k)] = \sum_{i\in[m]} \frac{\|\boldsymbol{\alpha}_i\|^2}{\|\mathbf{A}\|_F^2} \frac{|\mathbf{c}_i - \boldsymbol{\alpha}_i^T \mathbf{x}^k|^2}{\|\boldsymbol{\alpha}_i\|^2} = \frac{\|\mathbf{c} - \mathbf{A}\mathbf{x}^k\|^2}{\|\mathbf{A}\|_F^2}$$

measures the mean loss. Then we define the following index set

$$\mathcal{U}_k = \left\{ i_k : w_{i_k}(\mathbf{x}^k) \geq \theta \max_{i\in[m]} w_i(\mathbf{x}^k) + (1-\theta) \frac{\|\mathbf{c} - \mathbf{A}\mathbf{x}^k\|^2}{\|\mathbf{A}\|_F^2} \right\}. \tag{17}$$

Since $\max_{i\in[m]} w_i(\mathbf{x}^k) \geq \mathbb{E}[w_i(\mathbf{x}^k)]$, the index set $\mathcal{U}_k$ is nonempty and there always exists $i \in \mathcal{U}_k$. Let

$$\delta_k = \frac{\theta}{\|\mathbf{c} - \mathbf{A}\mathbf{x}^k\|^2} \max_{i\in[m]} \left\{ w_i(\mathbf{x}^k) \right\} + \frac{1-\theta}{\|\mathbf{A}\|_F^2}. \tag{18}$$

Then the index set $\mathcal{U}_k$ can be rewritten as

$$\mathcal{U}_k = \left\{ i_k : w_{i_k}(\mathbf{x}^k) \geq \delta_k \|\mathbf{c} - \mathbf{A}\mathbf{x}^k\|^2 \right\} = \left\{ i_k : w_{i_k}(\mathbf{x}^k) \geq \delta_k \|\mathbf{A}\mathbf{h}^k\|^2 \right\}. \tag{19}$$

When we randomly select an index from $\mathcal{U}_k$, the QRGRK method is stated as in Algorithm 3.1.

Remark 3.2. *As a special case of* QRGRK, *when $\theta = 1$, the row index i_k is selected determinately from the index set*

$$\mathcal{U}_k = \left\{ i_k : w_{i_k}(\mathbf{x}^k) = \max_{i\in[m]} w_i(\mathbf{x}^k) \right\}. \tag{20}$$

In this setting, we call it the QRGRKIMAX *method. When $\theta = 0.5$, the contributions of $\max_{i\in[m]} \left\{ w_i(\mathbf{x}^k) \right\}$ and $\mathbb{E}[w_i(\mathbf{x}^k)]$ are equal, we call it the* QGRK *method.*

In the calculation, the required quaternion matrix-vector multiplication, quaternion vector-number multiplication, the norm of a quaternion vector and the norm of a quaternion matrix are realized by the function (8) as follows:

$$\mathbf{A}\mathbf{x}_k = \mathscr{R}^{-1}(\mathscr{R}(\mathbf{A}\mathbf{x}_k)) = \mathscr{R}^{-1}(\mathscr{R}(\mathbf{A})\mathscr{R}(\mathbf{x}_k)), \tag{21a}$$

$$\bar{\boldsymbol{\alpha}}_{i_k} \frac{\mathbf{c}_{i_k} - \boldsymbol{\alpha}_{i_k}^T \mathbf{x}^k}{\|\boldsymbol{\alpha}_{i_k}\|^2} = \mathscr{R}^{-1}\left(\mathscr{R}\left(\bar{\boldsymbol{\alpha}}_{i_k} \frac{\mathbf{c}_{i_k} - \boldsymbol{\alpha}_{i_k}^T \mathbf{x}^k}{\|\boldsymbol{\alpha}_{i_k}\|^2} \right) \right) = \mathscr{R}^{-1}\left(\mathscr{R}(\bar{\boldsymbol{\alpha}}_{i_k}) \mathscr{R}\left(\frac{\mathbf{c}_{i_k} - \boldsymbol{\alpha}_{i_k}^T \mathbf{x}^k}{\|\boldsymbol{\alpha}_{i_k}\|^2} \right) \right), \tag{21b}$$

$$\|\boldsymbol{\alpha}_{i_k}\|^2 = \mathscr{R}^{-1}(\mathscr{R}(\boldsymbol{\alpha}_{i_k}^* \boldsymbol{\alpha}_{i_k})) = \mathscr{R}^{-1}(\mathscr{R}(\boldsymbol{\alpha}_{i_k}^*)\mathscr{R}(\boldsymbol{\alpha}_{i_k})), \tag{21c}$$

$$\|\mathbf{A}\|_F^2 = \mathscr{R}^{-1}(\mathscr{R}(\mathbf{A}^*\mathbf{A})) = \mathscr{R}^{-1}(\mathscr{R}(\mathbf{A}^*)\mathscr{R}(\mathbf{A})). \tag{21d}$$

Algorithm 3.1 The QRGRK method

Input: A coefficient matrix $\mathbf{A} \in \mathbb{Q}^{m\times n}$ and a right-hand side vector $\mathbf{c} \in \mathbb{Q}^m$, an initial vector $\mathbf{x}^0 \in \mathbb{Q}^n$ and a relaxation parameter $\theta \in [0, 1]$.
Input: last iterate $\mathbf{x}^{k+1}$.

1: **for** $k = 0, 1, 2, \cdots$, **do**
2: Compute

$$\delta_k = \frac{\theta}{\|\mathbf{c} - \mathbf{A}\mathbf{x}^k\|^2} \max_{i\in[m]} \left\{ w_i(\mathbf{x}^k) \right\} + \frac{1-\theta}{\|\mathbf{A}\|_F^2}.$$

3: Determine the index set $\mathcal{U}_k$ according to

$$\mathcal{U}_k = \left\{ i_k : w_{i_k}(\mathbf{x}^k) \geq \delta_k \|\mathbf{c} - \mathbf{A}\mathbf{x}^k\|^2 \right\}.$$

4: Compute the ith entry $\widetilde{\mathbf{r}}_i^k$ of the quaternion vector $\widetilde{\mathbf{r}}^k$ according to

$$\widetilde{\mathbf{r}}_i^k = \begin{cases} \mathbf{c}_i - \boldsymbol{\alpha}_i^T \mathbf{x}, & \text{if } i \in \mathcal{U}_k, \\ 0, & \text{otherwise.} \end{cases}$$

5: Select an index $i_k \in \mathcal{U}_k$ with probability $\mathbb{P}(\text{Row} = i_k) = \dfrac{|\widetilde{\mathbf{r}}_i^k|^2}{\|\widetilde{\mathbf{r}}^k\|^2}$.
6: Compute $\mathbf{x}^{k+1} = \mathbf{x}^k + \bar{\boldsymbol{\alpha}}_{i_k} \frac{\mathbf{c}_{i_k} - \boldsymbol{\alpha}_{i_k}^T \mathbf{x}^k}{\|\boldsymbol{\alpha}_{i_k}\|^2}$.
7: **end for**

We can implement the above calculations by only storing the first block columns of their real counterparts, which saves three-quarters of the computational operations. That is,

$$\mathscr{R}(\mathbf{A}\mathbf{x}_k)_c = \mathscr{R}(\mathbf{A})\mathscr{R}(\mathbf{x}_k)_c, \tag{22a}$$

$$\mathscr{R}\left(\bar{\boldsymbol{\alpha}}_{i_k} \frac{\mathbf{c}_{i_k} - \boldsymbol{\alpha}_{i_k}^T \mathbf{x}^k}{\|\boldsymbol{\alpha}_{i_k}\|^2} \right)_c = \mathscr{R}(\bar{\boldsymbol{\alpha}}_{i_k}) \mathscr{R}\left(\frac{\mathbf{c}_{i_k} - \boldsymbol{\alpha}_{i_k}^T \mathbf{x}^k}{\|\boldsymbol{\alpha}_{i_k}\|^2} \right)_c, \tag{22b}$$

$$\|\boldsymbol{\alpha}_{i_k}\|^2 = \|\mathscr{R}(\boldsymbol{\alpha}_{i_k})_c\|^2, \|\mathbf{A}\|_F^2 = \|\mathscr{R}(\mathbf{A})_c\|_F^2. \tag{22c}$$

This is the main idea of structure preserving reduction.

For the convergence property of the QRGRK method, we can obtain the following result.

Theorem 3.3. *Assume that the quaternion linear systems* (1)*, with the coefficient matrix* $\mathbf{A} \in \mathbb{Q}^{m\times n}$ *and the right-hand side* $\mathbf{c} \in \mathbb{Q}^m$*, is consistent. Then the iteration sequence* $\left\{\mathbf{x}^k\right\}_{k=0}^{\infty}$*, generated by the* QRGRK *method starting from an initial vector* $\mathbf{x}^0 \in \mathbb{Q}^n$*, converges to the solution* $\mathbf{x}_*$ *in expectation. Moreover, for all* $k \geq 1$*, the mean squared error in expectation satisfies*

$$\mathbb{E}[\|\mathbf{x}^{k+1} - \mathbf{x}_*\|_2^2] \leq \prod_{l=0}^{k} \tau_l \|\mathbf{x}^0 - \mathbf{x}_*\|_2^2,$$

where the parameters

$$\tau_0 = 1 - \frac{1}{\kappa_{F,2}^2(\mathbf{A})}, \quad \tau_k = 1 - \frac{\eta_k}{\kappa_{F,2}^2(\mathbf{A})}$$

with

$$\eta_k = \frac{\theta}{\gamma_k}\|\mathbf{A}\|_F^2 + (1-\theta),\ \ \gamma_k = \|\mathbf{A}\|_F^2 - \sum_{i\in\Omega_k}\|\boldsymbol{\alpha}_i\|^2 \ \textit{ and } \ \Omega_k = \left\{i : w_i(\mathbf{x}^k) = 0, i \in [m]\right\}.$$

Proof. From the relation (19), the expectation of the randomized loss with i_k is bounded by

$$\begin{aligned}\mathbb{E}_k[w_{i_k}(\mathbf{x}^k)] &= \sum_{i_k\in\mathcal{U}_k}\mathbb{P}(\text{Row} = i_k)w_{i_k}(\mathbf{x}^k) \geq \delta_k\|\mathbf{A}\mathbf{h}^k\|^2\sum_{i_k\in\mathcal{U}_k}\mathbb{P}(\text{Row} = i_k)\\ &= \delta_k\|\mathbf{A}\mathbf{h}^k\|^2 \geq \delta_k\frac{\|\mathbf{A}\|_F^2}{\kappa_{F,2}^2(\mathbf{A})}\|\mathbf{h}^k\|^2,\end{aligned} \tag{23}$$

where the second inequality follows from Lemma 2.2. This, together with the relation (16), yields

$$\mathbb{E}[\|\mathbf{h}^{k+1}\|^2] = \mathbb{E}[\|\mathbf{h}^k\|^2] - \mathbb{E}_k[w_{i_k}(\mathbf{x}^k)] \leq \left(1 - \delta_k\frac{\|\mathbf{A}\|_F^2}{\kappa_{F,2}^2(\mathbf{A})}\right)\mathbb{E}[\|\mathbf{h}^k\|^2]. \tag{24}$$

For $k = 0$, by the relation (18), we have

$$\begin{aligned}\delta_0\|\mathbf{A}\|_F^2 &= \theta\frac{\|\mathbf{A}\|_F^2}{\|\mathbf{c}-\mathbf{A}\mathbf{x}_0\|^2}\max_{i\in[m]}\left\{w_i(\mathbf{x}^0)\right\} + (1-\theta) = \theta\frac{\|\mathbf{A}\|_F^2\max_{i\in[m]}\left\{w_i(\mathbf{x}^0)\right\}}{\sum_{i\in[m]}\frac{|\mathbf{c}_i-\boldsymbol{\alpha}_i^T\mathbf{x}^0|^2}{\|\boldsymbol{\alpha}_i\|^2}\|\boldsymbol{\alpha}_i\|^2} + (1-\theta)\\ &\geq \theta\frac{\|\mathbf{A}\|_F^2\max_{i\in[m]}\left\{w_i(\mathbf{x}^0)\right\}}{\max_{i\in[m]}\{w_i(\mathbf{x}^0)\}\sum_{i\in[m]}\|\boldsymbol{\alpha}_i\|^2} + (1-\theta) = \theta + (1-\theta) = 1.\end{aligned} \tag{25}$$

For $k \geq 1$, it follows from the relation (18) that

$$\begin{aligned}\delta_k\|\mathbf{A}\|_F^2 &= \theta\frac{\|\mathbf{A}\|_F^2}{\|\mathbf{c}-\mathbf{A}\mathbf{x}^k\|^2}\max_{i\in[m]}\{w_i(\mathbf{x}^k)\} + (1-\theta) = \theta\frac{\|\mathbf{A}\|_F^2\max_{i\in[m]}\left\{w_i(\mathbf{x}^k)\right\}}{\sum_{i\in[m]}\frac{|\mathbf{c}_i-\boldsymbol{\alpha}_i^T\mathbf{x}^k|^2}{\|\boldsymbol{\alpha}_i\|^2}\|\boldsymbol{\alpha}_i\|^2} + (1-\theta)\\ &= \theta\frac{\|\mathbf{A}\|_F^2\max_{i\in[m]}\left\{w_i(\mathbf{x}^k)\right\}}{\sum_{i\in[m]}w_i(\mathbf{x}^k)\|\boldsymbol{\alpha}_i\|^2} + (1-\theta) \geq \theta\frac{\|\mathbf{A}\|_F^2\max_{i\in[m]}\{w_i(\mathbf{x}^k)\}}{\max_{i\in[m]}\{w_i(\mathbf{x}^k)\}(\sum_{i\in[m]} - \sum_{i\in\Omega_k})\|\boldsymbol{\alpha}_i\|^2} + (1-\theta)\\ &= \theta\frac{\|\mathbf{A}\|_F^2}{\|\mathbf{A}\|_F^2 - \sum_{i\in\Omega_k}\|\boldsymbol{\alpha}_i\|^2} + (1-\theta) = \frac{\theta\|\mathbf{A}\|_F^2}{\gamma_k} + (1-\theta) = \eta_k,\end{aligned} \tag{26}$$

where $\Omega_k = \left\{i : w_i(\mathbf{x}^k) = 0, i \in [m]\right\}$ and $\gamma_k = \|\mathbf{A}\|_F^2 - \sum_{i\in\Omega_k}\|\boldsymbol{\alpha}_i\|^2$. Combining (24)-(26) yields

$$\mathbb{E}[\|\mathbf{h}^{k+1}\|^2] \leq \left(1 - \delta_k\frac{\|\mathbf{A}\|_F^2}{\kappa_{F,2}^2(\mathbf{A})}\right)\mathbb{E}[\|\mathbf{h}^k\|^2] \leq \begin{cases}\left(1 - \frac{1}{\kappa_{F,2}^2(\mathbf{A})}\right)\mathbb{E}[\|\mathbf{h}^k\|^2], & \text{if } k = 0,\\ \left(1 - \frac{\eta_k}{\kappa_{F,2}^2(\mathbf{A})}\right)\mathbb{E}[\|\mathbf{h}^k\|^2], & \text{if } k \geq 1.\end{cases}$$

The proof is completed. □

Remark 3.4. *The set Ω_k is nonempty for $k \geq 1$. In fact, if we select the same rows at the $(k-1)$th and kth iterations, i.e., $i_k = i_{k-1}$, then*

$$w_{i_k}(\mathbf{x}^k) = \frac{|\mathbf{c}_{i_k} - \boldsymbol{\alpha}_{i_k}^T \mathbf{x}^k|^2}{\|\boldsymbol{\alpha}_{i_k}\|^2} = \frac{\left|\mathbf{c}_{i_k} - \boldsymbol{\alpha}_{i_k}^T\left(\mathbf{x}^{k-1} + \bar{\boldsymbol{\alpha}}_{i_{k-1}} \frac{\mathbf{c}_{i_{k-1}} - \boldsymbol{\alpha}_{i_{k-1}}^T \mathbf{x}^{k-1}}{\|\boldsymbol{\alpha}_{i_{k-1}}\|^2}\right)\right|^2}{\|\boldsymbol{\alpha}_{i_k}\|^2} = \frac{|\mathbf{c}_{i_k} - \boldsymbol{\alpha}_{i_k}^T \mathbf{x}^{k-1} - (\mathbf{c}_{i_k} - \boldsymbol{\alpha}_{i_k}^T \mathbf{x}^{k-1})|^2}{\|\boldsymbol{\alpha}_{i_k}\|^2} = 0.$$

This means that $i_{k-1} \in \Omega_k$.

Remark 3.5. *Since $\gamma_k = \|\mathbf{A}\|_F^2 - \sum_{i \in \Omega_k} \|\boldsymbol{\alpha}_i\|^2 \leq \|\mathbf{A}\|_F^2$, we have*

$$\eta_k = \frac{\theta}{\gamma_k}\|\mathbf{A}\|_F^2 + (1-\theta) \geq \frac{\theta}{\mathbf{A}\|_F^2}\|\mathbf{A}\|_F^2 + (1-\theta) = 1$$

Then

$$\tau_k = 1 - \frac{\eta_k}{\kappa_{F,2}^2(\mathbf{A})} \leq 1 - \frac{1}{\kappa_{F,2}^2(\mathbf{A})}.$$

It follows from Theorem 3.3 *that*

$$\mathbb{E}[\|\mathbf{x}^{k+1} - \mathbf{x}_*\|_2^2] \leq \prod_{l=0}^{k} \tau_l \|\mathbf{x}^0 - \mathbf{x}_*\|_2^2 \leq \left(1 - \frac{1}{\kappa_{F,2}^2(\mathbf{A})}\right)^{k+1} \|\mathbf{x}^0 - \mathbf{x}_*\|_2^2 \tag{27}$$

for all $k \geq 0$. This shows that the convergence factor of QRGRK *method given by* Theorem 3.3 *is slightly better than that given by the relation* (27).

Corollary 3.6. *The expectation of $w_i(\mathbf{x}^k)$ with respect to $i \in \mathcal{U}_k$ is bounded by*

$$\mathbb{E}[w_i(\mathbf{x}^k)] \geq \frac{\eta_k}{\kappa_{F,2}^2(\mathbf{A})}\mathbb{E}[\|\mathbf{x}^k - \mathbf{x}_*\|_2^2] \geq \frac{1}{\kappa_{F,2}^2(\mathbf{A})}\mathbb{E}[\|\mathbf{x}^k - \mathbf{x}_*\|_2^2], \tag{28}$$

where

$$\eta_k = \frac{\theta}{\gamma_k}\|\mathbf{A}\|_F^2 + (1-\theta), \quad \gamma_k = \|\mathbf{A}\|_F^2 - \sum_{i \in \Omega_k} \|\boldsymbol{\alpha}_i\|^2 \text{ and } \Omega_k = \left\{i : w_i(\mathbf{x}^k) = 0, i \in [m]\right\}.$$

Proof. The first inequality follows from the relations (23), (25) and (26). Since $\gamma_k \leq \|\mathbf{A}\|_F^2$, we have

$$\eta_k = \frac{\theta}{\gamma_k}\|\mathbf{A}\|_F^2 + (1-\theta) \geq \frac{\theta}{\|\mathbf{A}\|_F^2}\|\mathbf{A}\|_F^2 + (1-\theta) \geq 1.$$

The proof is completed. □

4. The PmQRGRK method

In this section, we incorporate Polyak's momentum acceleration technique into the structure preserving QRGRK method and propose the structure preserving PmQRGRK method for solving the quaternion linear systems (1). The Polyak's momentum acceleration technique is first introduced to modify the gradient descent (GD) method for improving the convergence [23]. As described earlier, Kaczmarz iterative scheme can be viewed as a special kind of gradient descent algorithm for minimizing the objective function

$$f_i(\mathbf{x}) = \frac{|\mathbf{c}_i - \boldsymbol{\alpha}_i^T \mathbf{x}|^2}{\|\boldsymbol{\alpha}_i\|^2}.$$

By using the Polyak's momentum technique, the explicit expression for the Polyak's momentum variant of quaternion linear system Kaczmarz iteration format is given by

$$\mathbf{x}^{k+1} = \mathbf{x}^k + \alpha \bar{\boldsymbol{\alpha}}_{i_k} \frac{\mathbf{c}_{i_k} - \boldsymbol{\alpha}_{i_k}^T \mathbf{x}^k}{\|\boldsymbol{\alpha}_{i_k}\|^2} + \beta(\mathbf{x}^k - \mathbf{x}^{k-1}),$$

where α is a positive real number that represents the step size and β is nonnegative real number that represents the momentum parameter.

Based on the relaxed greedy index selection strategy in our proposed structure preserving QRGRK algorithm, the structure preserving PmQRGRK method for solving the quaternion linear systems (1) is described as in Algorithm 4.1.

Algorithm 4.1 The PmQRGRK method

Input: A coefficient matrix $\mathbf{A} \in \mathbb{Q}^{m\times n}$ and a right-hand side vector $\mathbf{c} \in \mathbb{Q}^m$, two initial vectors $\mathbf{x}^0, \mathbf{x}^1 \in \mathbb{Q}^n$, a relaxation parameter $\theta \in [0, 1]$, a step size $\alpha > 0$ and a momentum parameter $\beta \geq 0$.
Input: last iterate $\mathbf{x}^{k+1}$.

1: **for** $k = 0, 1, 2, \cdots$, **do**
2: Compute

$$\delta_k = \frac{\theta}{\|\mathbf{c} - \mathbf{A}\mathbf{x}^k\|^2} \max_{i\in[m]} \left\{ w_i(\mathbf{x}^k) \right\} + \frac{1-\theta}{\|\mathbf{A}\|_F^2}.$$

3: Determine the index set $\mathcal{U}_k$ according to

$$\mathcal{U}_k = \left\{ i_k : w_{i_k}(\mathbf{x}^k) \geq \delta_k \|\mathbf{c} - \mathbf{A}\mathbf{x}^k\|^2 \right\}.$$

4: Compute the ith entry $\widetilde{\mathbf{r}}_i^k$ of the quaternion vector $\widetilde{\mathbf{r}}^k$ according to

$$\widetilde{\mathbf{r}}_i^k = \begin{cases} \mathbf{c}_i - \boldsymbol{\alpha}_i^T \mathbf{x}, & \text{if } i \in \mathcal{U}_k, \\ 0, & \text{otherwise.} \end{cases}$$

5: Select an index $i_k \in \mathcal{U}_k$ with probability $\mathbb{P}(\text{Row} = i_k) = \frac{|\widetilde{\mathbf{r}}_{i_k}^k|^2}{\|\widetilde{\mathbf{r}}^k\|^2}$.
6: Compute $\mathbf{x}^{k+1} = \mathbf{x}^k + \alpha \bar{\boldsymbol{\alpha}}_{i_k} \frac{\mathbf{c}_{i_k} - \boldsymbol{\alpha}_{i_k}^T \mathbf{x}^k}{\|\boldsymbol{\alpha}_{i_k}\|^2} + \beta(\mathbf{x}^k - \mathbf{x}^{k-1})$.
7: **end for**

Remark 4.2. *As a special case of* PmQRGRK, *when* $\theta = 1$, *the row index* i_k *is selected determinately from the index set*

$$\mathcal{U}_k = \left\{ i_k : w_{i_k}(\mathbf{x}^k) = \max_{i\in[m]} w_i(\mathbf{x}^k) \right\}. \tag{29}$$

In this setting, we call it the PmQRGRKIMAX *method. When* $\theta = 0.5$, *the contributions of* $\max_{i\in[m]} \left\{ w_i(\mathbf{x}^k) \right\}$ *and* $\mathbb{E}[w_i(\mathbf{x}^k)]$ *are equal, we call it the* PmQGRK *method.*

In the implementation of PmQRGRK method, the required quaternion matrix-vector multiplication, quaternion vector-number multiplication, the norm of a quaternion vector and the norm of a quaternion matrix are realized as in (22) by using the idea of structure preserving in order to solve the computations.

Next we analyze the convergence property of structure preserving PmQRGRK method. First we give the following lemma which will be used in our analysis.

Lemma 4.3. [17, Lemma 9] *Let $a_1 = a_0 \geq 0$ and $\{a_k\}_{k=0}^{\infty}$ be a sequence of nonnegative real numbers satisfying the relation*

$$a_{k+1} \leq w_1 v_k + w_2 v_{k-1}, \quad \forall k \geq 1,$$

where $w_2 \geq 0$, $w_1 + w_2 < 1$ and at least one of the coefficients w_1 and w_2 is positive. Then

$$a_{k+1} \leq q_1^k (1 + q_2) a_0, \quad \forall k \geq 0,$$

where $q_1 = (w_1 + \sqrt{w_1^2 + 4w_2})/2$ and $q_2 = q_1 - w_1 \geq 0$. Moreover, $q_1 \geq w_1 + w_2$, with equality if and only if $w_2 = 0$.

Theorem 4.4. *Assume that the quaternion linear systems* (1)*, with the coefficient matrix $\mathbf{A} \in \mathbb{Q}^{m\times n}$ and the right-hand side $\mathbf{c} \in \mathbb{Q}^m$, is consistent. Let $w_1 = (1 + 3\beta + 2\beta^2) - (2\alpha + \alpha\beta - \alpha^2)/\kappa_{F,2}^2(\mathbf{A})$ and $w_2 = 2\beta^2 + (1 + \alpha)\beta$. If $0 < \alpha < 2$, $\beta > 0$ and $w_1 + w_2 < 1$, then the iteration sequence $\left\{\mathbf{x}^k\right\}_{k=0}^{\infty}$, generated by* Algorithm 4.1 *starting from two initial vectors $\mathbf{x}^0 = \mathbf{x}^1 \in \mathbb{Q}^n$, satisfies*

$$\mathbb{E}[\|\mathbf{x}^{k+1} - \mathbf{x}_*\|^2] \leq \left(\frac{\sqrt{w_1^2 + 4w_2} + w_1}{2}\right)^k \left(\frac{\sqrt{w_1^2 + 4w_2} - w_1}{2} + 1\right) \|\mathbf{x}^0 - \mathbf{x}_*\|^2.$$

Proof. From Algorithm 4.1, we have

$$\begin{aligned}
\|\mathbf{x}^{k+1} - \mathbf{x}_*\|^2 &= \|\mathbf{x}^k + \alpha\bar{\boldsymbol{\alpha}}_{i_k}\frac{\mathbf{c}_{i_k} - \boldsymbol{\alpha}_{i_k}^T\mathbf{x}^k}{\|\boldsymbol{\alpha}_{i_k}\|^2} + \beta(\mathbf{x}^k - \mathbf{x}^{k-1}) - \mathbf{x}_*\|^2 \\
&= \|(\mathbf{x}^k - \mathbf{x}_*) + \alpha\bar{\boldsymbol{\alpha}}_{i_k} v_{i_k}(\mathbf{x}^k) + \beta(\mathbf{x}^k - \mathbf{x}_*) - \beta(\mathbf{x}^{k-1} - \mathbf{x}_*)\|^2,
\end{aligned}$$

where

$$v_{i_k}(\mathbf{x}^k) = \frac{\mathbf{c}_{i_k} - \boldsymbol{\alpha}_{i_k}^T\mathbf{x}^k}{\|\boldsymbol{\alpha}_{i_k}\|^2}. \tag{30}$$

It follows that

$$\begin{aligned}
\|\mathbf{h}^{k+1}\|^2 &= \|\mathbf{h}^k + \alpha\bar{\boldsymbol{\alpha}}_{i_k} v_{i_k}(\mathbf{x}^k) + \beta(\mathbf{h}^k - \mathbf{h}^{k-1})\|^2 \\
&= \|\mathbf{h}^k + \alpha\bar{\boldsymbol{\alpha}}_{i_k} v_{i_k}(\mathbf{x}^k)\|^2 + \beta\langle \mathbf{h}^k + \alpha\bar{\boldsymbol{\alpha}}_{i_k} v_{i_k}(\mathbf{x}^k), \mathbf{h}^k - \mathbf{h}^{k-1}\rangle \\
&+ \beta\langle \mathbf{h}^k - \mathbf{h}^{k-1}, \mathbf{h}^k + \alpha\bar{\boldsymbol{\alpha}}_{i_k} v_{i_k}(\mathbf{x}^k)\rangle + \beta^2\|\mathbf{h}^k - \mathbf{h}^{k-1}\|^2 \\
&= s_{k,1} + s_{k,2} + s_{k,3},
\end{aligned} \tag{31}$$

where

$$\begin{cases}
s_{k,1} = \|\mathbf{h}^k + \alpha\bar{\boldsymbol{\alpha}}_{i_k} v_{i_k}(\mathbf{x}^k)\|^2, \\
s_{k,2} = \beta\langle \mathbf{h}^k + \alpha\bar{\boldsymbol{\alpha}}_{i_k} v_{i_k}(\mathbf{x}^k), \mathbf{h}^k - \mathbf{h}^{k-1}\rangle + \beta\langle \mathbf{h}^k - \mathbf{h}^{k-1}, \mathbf{h}^k + \alpha\bar{\boldsymbol{\alpha}}_{i_k} v_{i_k}(\mathbf{x}^k)\rangle \\
s_{k,3} = \beta^2\|\mathbf{h}^k - \mathbf{h}^{k-1}\|^2
\end{cases}$$

Notice that

$$\|\bar{\boldsymbol{\alpha}}_{i_k} v_{i_k}(\mathbf{x}^k)\|^2 = -\langle \mathbf{h}^k, \bar{\boldsymbol{\alpha}}_{i_k} v_{i_k}(\mathbf{x}^k)\rangle = -\langle \bar{\boldsymbol{\alpha}}_{i_k} v_{i_k}(\mathbf{x}^k), \mathbf{h}^k\rangle = \frac{|\mathbf{c}_{i_k} - \boldsymbol{\alpha}_{i_k}^T\mathbf{x}^k|^2}{\|\boldsymbol{\alpha}_{i_k}\|^2} = w_{i_k}(\mathbf{x}^k). \tag{32}$$

Then

$$\begin{aligned} s_{k,1} &= \|\mathbf{h}^k + \alpha\bar{\boldsymbol{\alpha}}_{i_k} v_{i_k}(\mathbf{x}^k)\|^2 = \|\mathbf{h}^k\|^2 + \alpha\langle \mathbf{h}^k, \bar{\boldsymbol{\alpha}}_{i_k} v_{i_k}(\mathbf{x}^k)\rangle + \alpha\langle \bar{\boldsymbol{\alpha}}_{i_k} v_{i_k}(\mathbf{x}^k), \mathbf{h}^k\rangle + \alpha^2\|\bar{\boldsymbol{\alpha}}_{i_k} v_{i_k}(\mathbf{x}^k)\|_2^2 \\ &= \|\mathbf{h}^k\|^2 - \alpha w_{i_k}(\mathbf{x}^k) - \alpha w_{i_k}(\mathbf{x}^k) + \alpha^2 w_{i_k}(\mathbf{x}^k) = \|\mathbf{h}^k\|^2 + (\alpha^2 - 2\alpha) w_{i_k}(\mathbf{x}^k). \end{aligned} \tag{33}$$

Direct calculations give

$$\begin{aligned} & s_{k,2} = \beta\langle \mathbf{h}^k + \alpha\bar{\boldsymbol{\alpha}}_{i_k} v_{i_k}(\mathbf{x}^k), \mathbf{h}^k - \mathbf{h}^{k-1}\rangle + \beta\langle \mathbf{h}^k - \mathbf{h}^{k-1}, \mathbf{h}^k + \alpha\bar{\boldsymbol{\alpha}}_{i_k} v_{i_k}(\mathbf{x}^k)\rangle \\ &= \beta\left[\langle \mathbf{h}^k, \mathbf{h}^k - \mathbf{h}^{k-1}\rangle + \langle \mathbf{h}^k - \mathbf{h}^{k-1}, \mathbf{h}^k\rangle\right] + \alpha\beta\left[\langle \bar{\boldsymbol{\alpha}}_{i_k} v_{i_k}(\mathbf{x}^k), \mathbf{h}^k - \mathbf{h}^{k-1}\rangle + \langle \mathbf{h}^k - \mathbf{h}^{k-1}, \bar{\boldsymbol{\alpha}}_{i_k} v_{i_k}(\mathbf{x}^k)\rangle\right] \\ &= \beta[2\|\mathbf{h}^k\|^2 - \langle \mathbf{h}^k, \mathbf{h}^{k-1}\rangle - \langle \mathbf{h}^{k-1}, \mathbf{h}^k\rangle] + \alpha\beta[\langle \bar{\boldsymbol{\alpha}}_{i_k} v_{i_k}(\mathbf{x}^k), \mathbf{h}^k\rangle + \langle \mathbf{h}^k, \bar{\boldsymbol{\alpha}}_{i_k} v_{i_k}(\mathbf{x}^k)\rangle] \\ & -\alpha\beta[\langle \bar{\boldsymbol{\alpha}}_{i_k} v_{i_k}(\mathbf{x}^k), \mathbf{h}^{k-1}\rangle + \langle \mathbf{h}^{k-1}, \bar{\boldsymbol{\alpha}}_{i_k} v_{i_k}(\mathbf{x}^k)\rangle] \\ &= [3\beta\|\mathbf{h}^k\|^2 + \beta\|\mathbf{h}^{k-1}\|^2] + \alpha\beta[-w_{i_k}(\mathbf{x}^k) - w_{i_k}(\mathbf{x}^k)] + \alpha\beta[\|\bar{\boldsymbol{\alpha}}_{i_k} v_{i_k}(\mathbf{x}^k)\|_2^2 + \|\mathbf{h}^{k-1}\|^2] \\ &\le [3\beta\|\mathbf{h}^k\|^2 + \beta\|\mathbf{h}^{k-1}\|^2] + \alpha\beta[-w_{i_k}(\mathbf{x}^k) - w_{i_k}(\mathbf{x}^k)] + \alpha\beta[w_{i_k}(\mathbf{x}^k) + \|\mathbf{h}^{k-1}\|^2] \\ &= 3\beta\|\mathbf{h}^k\|^2 + \beta\|\mathbf{h}^{k-1}\|^2 + \alpha\beta\|\mathbf{h}^{k-1}\|^2 - \alpha\beta w_{i_k}(\mathbf{x}^k). \end{aligned} \tag{34}$$

Using the inequality$\|\mathbf{x} - \mathbf{y}\|^2 \le 2(\|\mathbf{x} - \mathbf{z}\|^2 + \|\mathbf{y} - \mathbf{z}\|^2)$ for any vectors $\mathbf{x}, \mathbf{y}$ and $\mathbf{z}$ with compatible dimension, it holds that

$$s_{k,3} = \beta^2\|\mathbf{h}^k - \mathbf{h}^{k-1}\|^2 \le 2\beta^2(\|\mathbf{h}^k\|^2 + \|\mathbf{h}^{k-1}\|^2). \tag{35}$$

Combining the relations (33), (34) and (35) yields

$$\begin{aligned} \|\mathbf{h}^{k+1}\|^2 &= s_{k,1} + s_{k,2} + s_{k,3} \\ &\le \|\mathbf{h}^k\|^2 + (\alpha^2 - 2\alpha) w_{i_k}(\mathbf{x}^k) + 3\beta\|\mathbf{h}^k\|^2 + \beta\|\mathbf{h}^{k-1}\|^2 \\ &\quad +\alpha\beta\|\mathbf{h}^{k-1}\|^2 - \alpha\beta w_{i_k}(\mathbf{x}^k) + 2\beta^2(\|\mathbf{h}^k\|^2 + \|\mathbf{h}^{k-1}\|^2) \\ &= (1 + 3\beta + 2\beta^2)\|\mathbf{h}^k\|^2 + (\alpha^2 - 2\alpha - \alpha\beta) w_{i_k}(\mathbf{x}^k) + [2\beta^2 + (1+\alpha)\beta]\|\mathbf{h}^{k-1}\|^2. \end{aligned} \tag{36}$$

Since $0 < \alpha < 2$ and $\beta > 0$, we have $\alpha^2 - 2\alpha - \alpha\beta < 0$. Taking expectation with respect to $i_k \in \mathcal{U}_k$ yields

$$\begin{aligned} & \mathbb{E}_k[\|\mathbf{h}^{k+1}\|^2] \le (1 + 3\beta + 2\beta^2)\mathbb{E}_k[\|\mathbf{h}^k\|^2] + (\alpha^2 - 2\alpha - \alpha\beta)\mathbb{E}_k[w_{i_k}(\mathbf{x}^k)] + [2\beta^2 + (1+\alpha)\beta]\mathbb{E}_k[\|\mathbf{h}^{k-1}\|^2] \\ \le\ & (1 + 3\beta + 2\beta^2)\mathbb{E}_k[\|\mathbf{h}^k\|^2] - (2\alpha + \alpha\beta - \alpha^2)/\kappa_{F,2}^2(\mathbf{A})\mathbb{E}_k[\|\mathbf{h}^k\|^2] + [2\beta^2 + (1+\alpha)\beta]\mathbb{E}_k[\|\mathbf{h}^{k-1}\|^2] \\ =\ & [(1 + 3\beta + 2\beta^2) - (2\alpha + \alpha\beta - \alpha^2)/\kappa_{F,2}^2(\mathbf{A})\mathbb{E}_k[\|\mathbf{h}^k\|^2] + [2\beta^2 + (1+\alpha)\beta]\mathbb{E}_k[\|\mathbf{h}^{k-1}\|^2]. \end{aligned}$$

where the second inequality follows from Corollary 3.6. Taking expectation over the whole history, we have

$$E[\|\mathbf{x}^{k+1} - \mathbf{x}_*\|_2^2] \le w_1 E[\|\mathbf{x}^k - \mathbf{x}_*\|_2^2] + w_2 E[\|\mathbf{x}^{k-1} - \mathbf{x}_*\|_2^2].$$

It is seen that $w_2 = 2\beta^2 + (1+\alpha)\beta > 0$ since $0 < \alpha < 2$ and $\beta > 0$. By the assumption that $w_1 + w_2 < 1$, the desired result is obtained according to Lemma 4.3. □

Remark 4.5. *Let* $\xi_1 = 4 + \alpha - \alpha/\kappa_{F,2}^2(\mathbf{A})$ *and* $\xi_2 = \alpha(2-\alpha)/\kappa_{F,2}^2(\mathbf{A})$*. If* $0 < \alpha < 2$ *and*

$$0 < \beta < \left(\sqrt{\xi_1^2 + 16\xi_2} - \xi_1\right)/8$$

then

$$w_1 + w_2 = 4\beta^2 + \xi_1\beta - \xi_2 + 1 < 1$$

and the condition in Theorem 4.4 *is satisfied.*

5. Numerical experiments

In this section, we demonstrate the efficiency of our proposed QRGRK and PmQRGRK in solving the quaternion linear systems (1). All tests were conducted using MATLAB (R2022b) on a personal computer with Intel Core i5-13420H CPU @2.10GHz/16.00GB.

Let $\mathbf{A} = A_0 + A_1\mathbf{i} + A_2\mathbf{j} + A_3\mathbf{k} \in \mathbb{Q}^{m\times n}$, $\mathbf{x} = x_0 + x_1\mathbf{i} + x_2\mathbf{j} + x_3\mathbf{k} \in \mathbb{Q}^n$, and $\mathbf{c} = c_0 + c_1\mathbf{i} + c_2\mathbf{j} + c_3\mathbf{k} \in \mathbb{Q}^m$. ME-RGRK and ME-MWRK [33] are utilized to solve the following real linear matrix equation arising from the quaternion linear systems

$$\mathscr{R}(\mathbf{A})\mathscr{R}(\mathbf{x}) = \mathscr{R}(\mathbf{c}).$$

GRK [3] and RGRK [4] are used to solve the following real linear systems

$$\mathscr{R}(\mathbf{A})[\begin{array}{cccc} x_0 & x_1 & x_2 & x_3 \end{array}]^T = [\begin{array}{cccc} c_0 & c_1 & c_2 & c_3 \end{array}]^T.$$

We also report the numerical performance of QRK method for solving the quaternion linear systems (1), which is a natural generalization of RK method [30] over the quaternion skew-field. We comment here that QRGRK reduces to QRK if the row index $i_k \in [m]$ is selected with the probability $\|\boldsymbol{\alpha}_{i_k,:}\|^2/\|A\|_F^2$ without using the relaxed greedy selection strategy.

For the above-metioned methods, we count the number of iterations (denoted by "IT"), the computation time in seconds (denoted by "CPU"), and the relative residual norm (denoted by "RRN"). Here RRN is defined as

$$\mathrm{RRN} = \frac{\|\mathbf{c} - \mathbf{A}\mathbf{x}\|}{\|\mathbf{c}\|}.$$

The stopping criterion is that the relative residual norm is less than tol $= 10^{-6}$ or the maximum number of iterations 80000 is exceeded.

5.1. Random data

In this subsection, we report the numerical performance of the above methods on random data.

Example 5.1. *Consider the quaternion linear systems* (1)*, where the coefficient matrix* $\mathbf{A} \in \mathbb{Q}^{m\times n}$ *and the solution vector* $\mathbf{x} \in \mathbb{Q}^n$ *are randomly generated by* MATLAB *built-in functions. Then the right-hand side vector* $\mathbf{c} \in \mathbb{Q}^m$ *is generated by* $\mathbf{c} = \mathbf{A}\mathbf{x}$.

We compare the numerical results of QRGRK, GRK, RGRK, ME-RGRK and ME-MWRK methods. With a special focus on the case when $\theta = 0.3, 0.5, 0.7$ and 1. QRGRK is designated as QGRK when $\theta = 0.5$, and QRGRK is referred to as QRGRKIMAX when $\theta = 1$, see Remark 3.2. We report the IT, CPU and RNN for different values θ and (m, n) in Table 1. From Table 1, we see that QRGRK converges in fewer iterations and costs less running time than GRK, RGRK, ME-RGRK and ME-MWRK, while their relative residual norms are comparable with each other. The convergence curves of QRGRK, RGRK and ME-RGRK are shown in Figure 1, which indicates that QRGRK stops earlier than RGRK and ME-RGRK.

The result in Table 1 and Figure 1 reveals a clear trend: the QRGRK method, due to its property of preserving the quaternion structure, outperforms the ME-RGRK and RGRK methods.

Example 5.2. *Consider the quaternion linear systems* (1)*, where the coefficient matrix* $\mathbf{A}$ *and the right-hand side vector* $\mathbf{c}$ *are the same as in* Example 5.1.

Table 1. Numerical comparison results for Example 5.1.

(m,n)		(400,400)	(600,600)	(800,800)	(150,4000)	(150,5000)	(200,4000)	(200,5000)
$\theta = 0.3$								
QRGRK	IT	3586	3618	3981	557	483	640	577
	CPU	0.7649	2.1041	5.1449	0.8977	1.0457	1.4016	1.6997
	RRN	9.4849e-07	9.2795e-07	9.9509e-07	9.4976e-07	9.8304e-07	9.4570e-07	9.9997e-07
RGRK[4]	IT	7793	8068	8877	1323	1224	1662	1516
	CPU	2.9179	10.9711	32.7239	2.6794	3.1788	4.5792	5.2544
	RRN	9.9641e-07	9.9566e-07	9.6466e-07	9.5285e-07	9.9270e-07	9.9496e-07	9.8057e-07
ME-RGRK[33]	IT	31740	31771	37878	5409	5240	6743	6199
	CPU	24.2748	81.2881	426.0728	22.2953	22.0684	34.8505	40.9924
	RRN	9.9564e-07	9.9371e-07	9.9112e-07	9.9607e-07	9.9811e-07	9.7769e-07	9.9714e-07
$\theta = 0.5$								
QGRK	IT	3198	3740	3782	507	498	616	587
	CPU	0.4887	1.6832	5.6177	0.8719	1.0977	1.3660	1.7889
	RRN	9.6792e-07	9.7637e-07	9.5683e-07	9.6395e-07	9.6389e-07	9.6698e-07	9.7827e-07
GRK[3]	IT	7339	8759	8699	1235	1245	1460	1299
	CPU	2.4794	10.6485	19.5935	2.5477	3.2641	4.2001	4.6311
	RRN	9.7916e-07	9.8588e-07	9.6214e-07	9.6005e-07	9.3264e-07	9.4163e-07	9.6461e-07
ME-RGRK[33]	IT	30126	34850	36193	4939	4938	5779	5551
	CPU	21.3598	63.3470	210.5820	17.6127	23.3785	29.6373	32.8931
	RRN	9.9834e-07	9.9976e-07	9.9645e-07	9.9025e-07	9.8659e-07	9.9323e-07	9.9157e-07
$\theta = 0.7$								
QRGRK	IT	3256	3616	3814	483	491	587	570
	CPU	0.5454	1.5020	4.9530	0.8310	1.0554	1.3242	1.6931
	RRN	9.9340e-07	9.9597e-07	9.9024e-07	9.6521e-07	9.8343e-07	9.9699e-07	9.5663e-07
RGRK[4]	IT	7818	7961	8287	1091	1113	1370	1153
	CPU	2.6853	8.5555	18.2414	2.3594	2.8142	3.6814	3.8872
	RRN	9.9007e-07	9.9057e-07	9.9870e-07	9.7020e-07	9.8320e-07	9.8475e-07	9.5831e-07
ME-RGRK[33]	IT	30137	33775	34211	4314	4410	5580	4989
	CPU	21.6233	142.1919	171.8004	14.9318	19.3515	29.3696	30.6972
	RRN	9.8423e-07	9.9583e-07	9.9976e-07	9.9485e-07	9.5948e-07	9.9180e-07	9.8019e-07
$\theta = 1$								
QRGRKIMAX	IT	3193	3700	3887	470	486	611	549
	CPU	0.3965	1.4888	4.6311	0.7402	0.9956	1.3370	1.5759
	RRN	9.9892e-07	9.9084e-07	9.9358e-07	9.3283e-07	9.7406e-07	9.6450e-07	9.6162e-07
RGRK[4]	IT	7164	8744	8852	997	981	1241	1157
	CPU	2.2798	9.9804	20.7625	2.1318	2.5756	3.2859	3.9586
	RRN	9.8478e-07	9.9266e-07	9.9468e-07	9.1634e-07	9.9480e-07	9.6102e-07	9.5268e-07
ME-MWRK[33]	IT	30184	35025	35434	4040	4129	5159	4738
	CPU	19.4501	147.6971	183.5147	15.2816	18.2747	25.4311	34.3186
	RRN	9.9972e-07	9.9820e-07	9.9897e-07	9.7379e-07	9.5924e-07	9.9913e-07	9.8552e-07

In this example, we report the numerical results of PmQRGRK, QRGRK and QRK for different dimension (m,n) and relaxation parameter θ. For the PmQRGRK method, we choose $(\alpha,\beta) = (1.6, 0.5)$. PmQRGRK is designated as PmQGRK when $\theta = 0.5$, and PmQRGRK is referred to as PmQRGRKIMAX when $\theta = 1$, see Remark 4.2. We list the computing time in seconds, the iterations, and the relative residual norm in Table 2. From Table 2, it is clearly seen that both PmQRGRK and QRGRK methods are more efficient than QRK method in terms of the number of iterations and elapsed CPU time. In Figure 2, we plot the convergence curves of PmQRGRK, QRGRK and QRK with the relaxation parameter $\theta = 0.7$ and $m = n = 800$. Figure 2 shows that the relative residual norm of PmQRGRK and QRGRK methods decreases faster than that of QRK method with the increasing of the iteration number. In all cases, PmQRGRK method converges faster and needs less running time than QRGRK by an appropriate selection of step-size α and momentum parameter β.

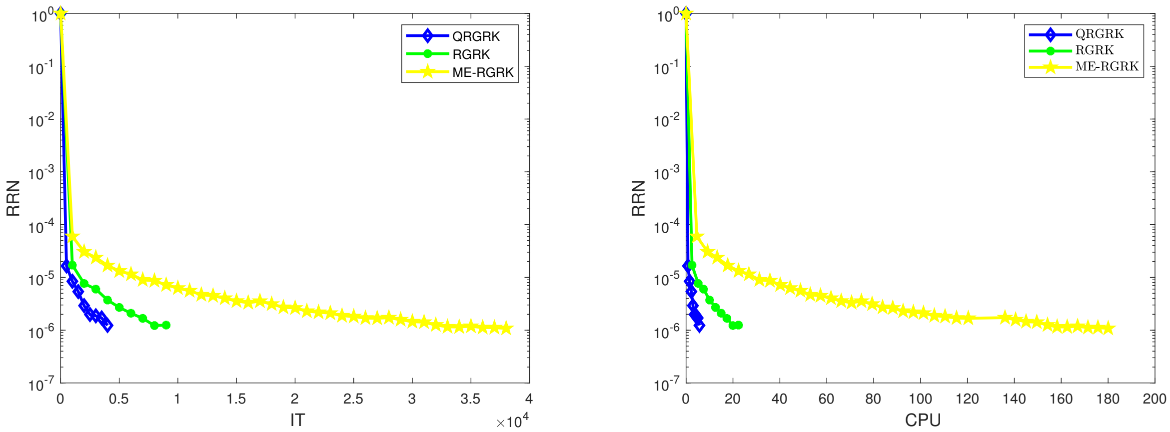


Figure 1. The convergence behaviors of RRN versus IT (left) and CPU (right) given by the QRGRK, RGRK and ME-RGRK methods for Example 5.1 with $m = n = 800$ and $\theta = 0.7$.

Table 2. Numerical comparison results for Example 5.2.

(m, n)		(400,400)	(600,600)	(800,800)	(150,4000)	(150,5000)	(200,4000)	(200,5000)
QRK	IT	7679	8928	10259	2447	2545	2081	3062
	CPU	1.3604	6.1008	15.2766	3.4540	4.6180	2.9677	8.0580
	RRN	9.8507e-07	9.9655e-07	9.9515e-07	9.9975e-07	9.8337e-07	9.9719e-07	9.6126e-07
$\theta = 0.3$								
PmQRGRK	IT	3067	3360	4297	481	448	433	552
	CPU	0.5826	1.9800	5.6979	0.8292	0.9774	0.6980	1.5773
	RRN	9.4557e-07	9.6007e-07	9.8160e-07	9.0167e-07	9.0846e-07	9.6128e-07	9.4848e-07
QRGRK	IT	3138	3656	4223	518	508	496	594
	CPU	0.6108	2.4656	6.1829	0.8634	1.0579	0.7292	1.7379
	RRN	9.4588e-07	9.2891e-07	9.3432e-07	9.9069e-07	9.9952e-07	9.6150e-07	9.5386e-07
$\theta = 0.5$								
PmQGRK	IT	2962	3598	3913	420	422	515	506
	CPU	0.5828	2.2469	6.4063	0.6905	0.9302	1.1428	1.4943
	RRN	9.9704e-07	9.3380e-07	9.8940e-07	9.5296e-07	9.2194e-07	9.3331e-07	9.4577e-07
QGRK	IT	3301	3581	3988	497	519	574	589
	CPU	0.6556	2.5739	6.4691	0.7792	1.1016	1.2299	1.6604
	RRN	9.6964e-07	9.8490e-07	9.9023e-07	9.7964e-07	9.8190e-07	9.6647e-07	9.3163e-07
$\theta = 0.7$								
PmQRGRK	IT	2930	3348	3660	387	402	518	475
	CPU	0.5602	2.0363	5.2507	0.6811	0.8557	1.1860	1.4062
	RRN	9.8689e-07	9.8221e-07	9.5437e-07	8.8962e-07	9.0705e-07	9.4005e-07	9.6977e-07
QRGRK	IT	3321	3623	3967	489	487	646	542
	CPU	0.6521	2.0702	5.4902	0.8023	1.0021	1.3672	1.5143
	RRN	9.9365e-07	9.7034e-07	9.8987e-07	9.9388e-07	9.5261e-07	9.4812e-07	9.7740e-07
$\theta = 1$								
PmQRGRKIMAX	IT	3262	3427	3509	383	369	463	440
	CPU	0.6393	1.9017	4.8326	0.6180	0.7789	1.0308	1.2701
	RRN	9.7905e-07	9.8603e-07	9.1515e-07	7.9726e-07	8.6315e-07	9.8607e-07	9.1267e-07
QRGRKIMAX	IT	3560	3580	3938	500	470	579	540
	CPU	0.6800	2.4889	5.4521	0.7518	0.9128	1.0979	1.5146
	RRN	9.5027e-07	9.8655e-07	9.6314e-07	9.8550e-07	9.9251e-07	9.7178e-07	9.4985e-07

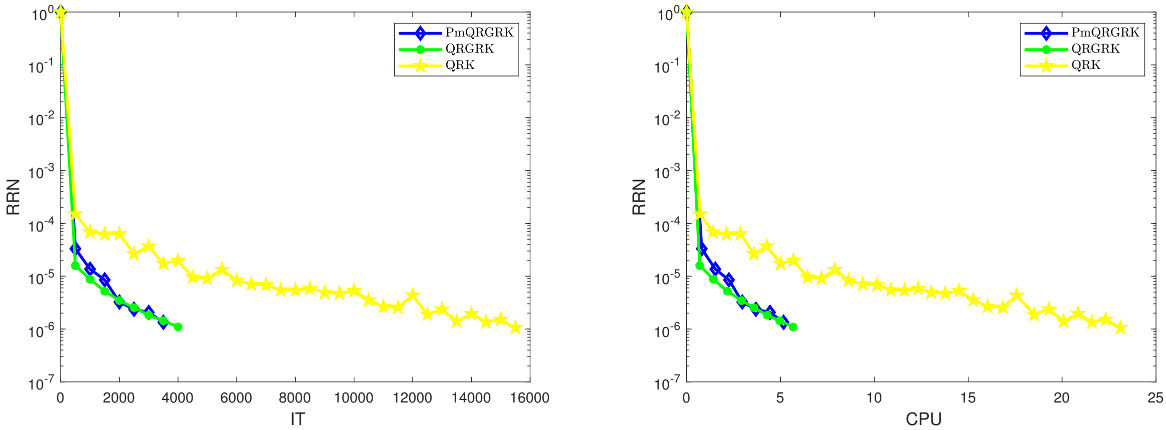


Figure 2. The convergence behaviors of RRN versus IT (left) and CPU (right) given by the PmQRGRK, QRGRK and QRK methods for Example 5.2 with $m = n = 800$, $\theta = 0.7$ and $(\alpha, \beta) = (1.6, 0.5)$.

In the following, we use PmQRGRK with various (α, β) to solve the quaternion linear systems, as shown in Figure 3. Especially, we assign the input parameters $m = 150$ and $n = 4000$, and to (α, β) as (0.5,0.6), (0.7,0.6), (1.0,0.4), (1.4,0.5), (1.6,0.3), and (1.6,0.5). It can be seen that PmQRGRK method successfully calculates the approximate solution for all cases, and the satisfactory parameter pairs are selected by $(\alpha, \beta) = (1.4, 0.5)$ and $(1.6, 0.3)$, respectively. In this case, the number of iterations of the PmQRGRK method is 450, 406, 409, and 357, respectively, when $\theta = 0.3, 0.5, 0.7$, and 1. In a word, PmQRGRK can converge faster by choosing the appropriate iterative pair of parameters.

5.2. *The signal filter problem*

In this section, we implement the QRGRK, PmQRGRK, QRK, GRK, RGRK, ME-RGRK and ME-MWRK methods to solve the signal filter problem introduced by [10].

Example 5.3. *Consider the Lorenz attractor, i.e., a three-dimensional nonlinear system, applied in atmospheric turbulence* [26]. *It can be formulated as the following system of coupled differential equations*

$$\frac{\partial x}{\partial t} = \phi(y - x), \ \frac{\partial y}{\partial t} = x(\rho - z) - y, \ \frac{\partial z}{\partial t} = xy - \psi z, \tag{37}$$

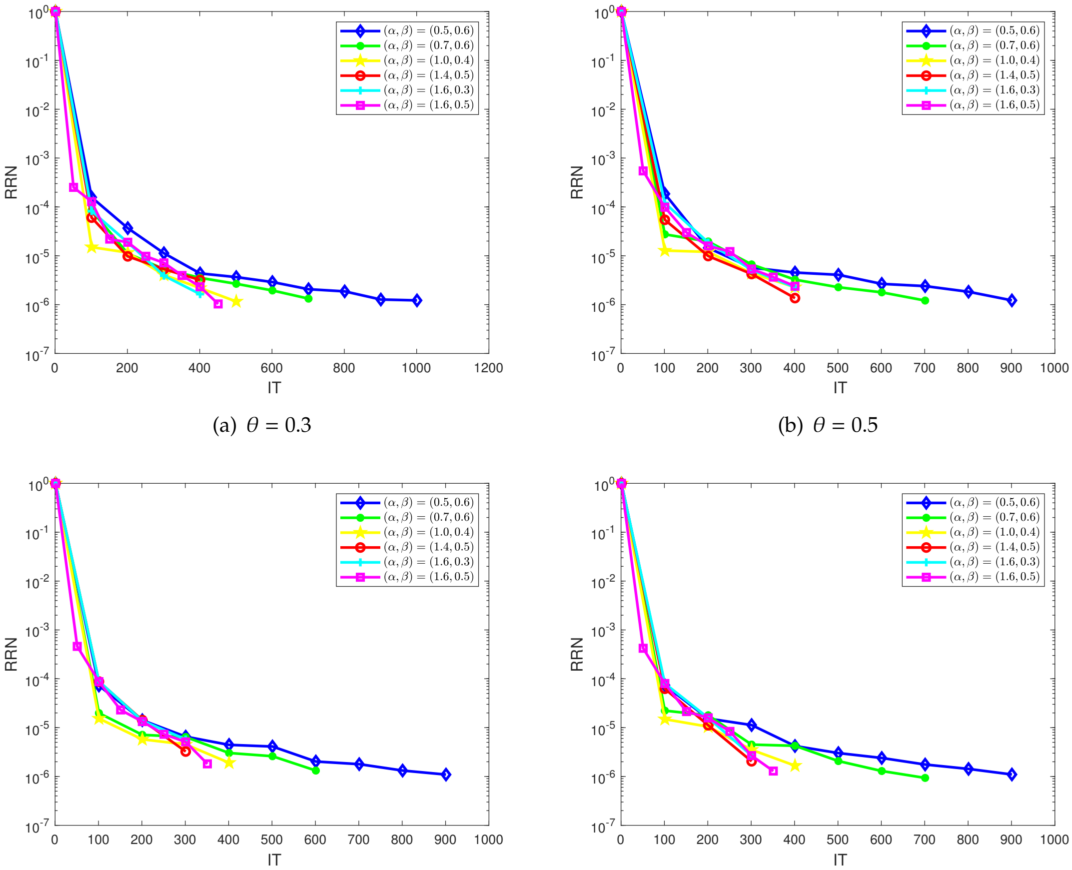
where $\phi, \psi, \rho > 0$. *Here we choose the parameters* $\phi = 10$, $\psi = 8/3$, $\rho = 28$ *to characterize the chaotic behavior of the Lorenz attractor. By using the* MATLAB *command* ODE45$(f(t, [x, y, z]), [0, T], [1, 1, 1])$, *the equation* (37) *can be solved.*

Let $\mathbf{x}(t) = x_r(t)\mathbf{i} + x_g(t)\mathbf{j} + x_b(t)\mathbf{k}$ with $x_r(t)$, $x_g(t)$ and $x_b(t)$ being the solutions of (37). The input signal with the random noise has the following form

$$\mathbf{c}(t) = c_r(t-1)\mathbf{i} + c_g(t-1)\mathbf{j} + c_b(t-1)\mathbf{k} + \mathbf{n}(t),$$

where $\mathbf{n}(t)$ is a random noise. The quaternion linear systems (1) is then built, see [10].

(1) First, we demonstrate the superiority of QRGRK method by comparing it with GRK, RGRK, ME-RGRK and ME-MWRK. The numerical results are reported in Table 3. The convergence histories are depicted in Figure 4. We observe from Table 3 and Figure 4 that QRGRK method is more efficient than GRK, RGRK, ME-RGRK and ME-MWRK methods in dealing with the signal filter problem, because it can achieve lower relative residual norm in shorter time. In addition, the QRGRK method requires less running time to reach the same number of iterations.



(a) $\theta = 0.3$ (b) $\theta = 0.5$

(c) $\theta = 0.7$ (d) $\theta = 1$

Figure 3. The convergence behaviors of RRN versus IT given by the PmQRGRK method for Example 5.2 with $m = 150, n = 4000$.

(2) Second, we illustrate the effectiveness of the momentum technique. We take $(\alpha,\beta) = (1.4, 0.5)$ for the PmQRGRK method. The IT, CPU and RNN are listed in Table 4 and the convergence curves are plotted in Figure 5. From Table 4 and Figure 5, we see that the PmQRGRK method is superior to QRGRK and QRK method in terms of convergence speed and computation time. Besides, the calculation times of the PmQRGRK method under fixed precision are significantly less than those of the QRGRK and QRK methods, which shows the superiority of the momentum acceleration.

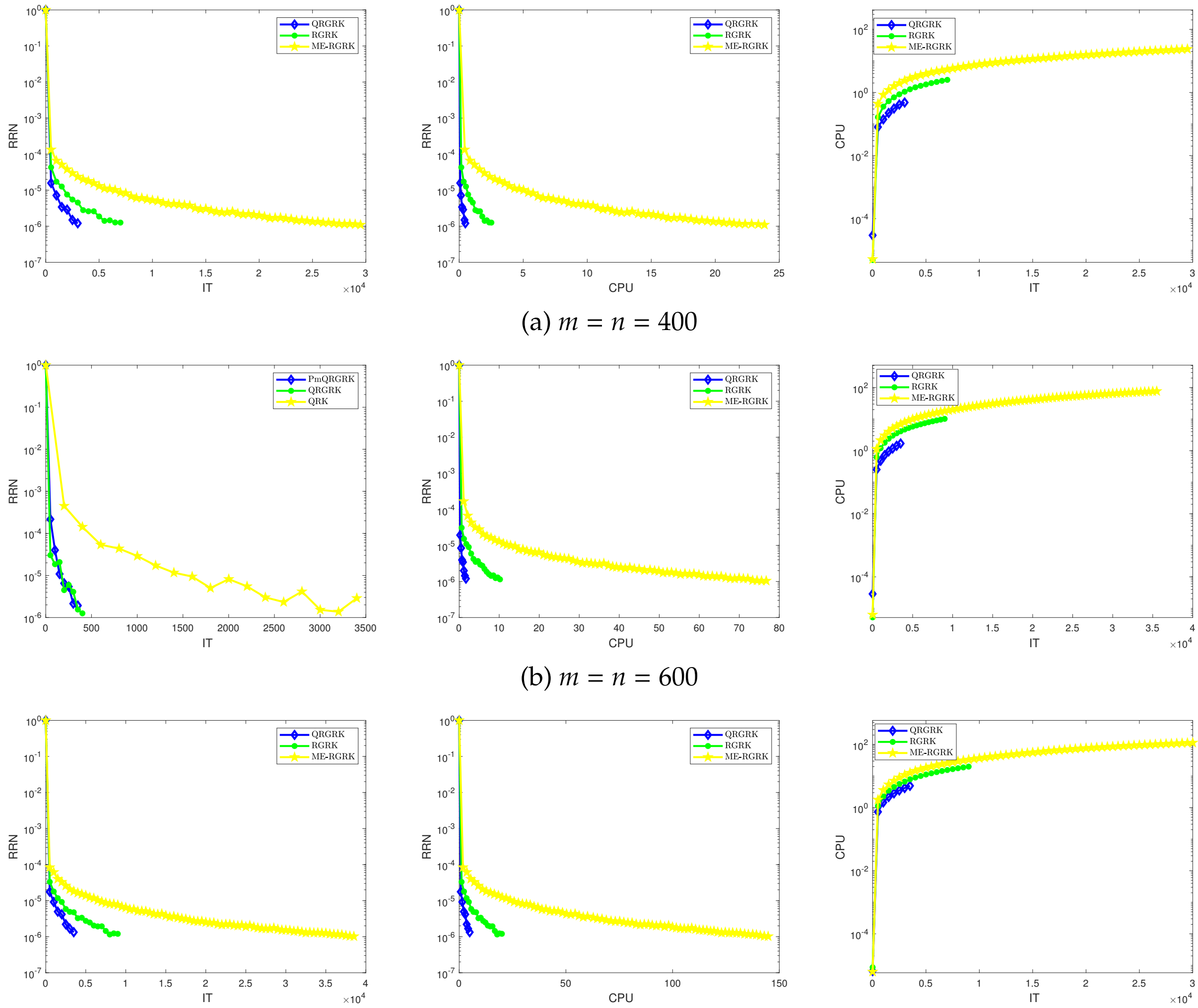
Table 3. Numerical comparison results for Example 5.3.

(m,n)		(400,400)	(600,600)	(800,800)	(150,4000)	(150,5000)	(200,4000)	(200,5000)
$\theta = 0.3$								
QRGRK	IT	3199	3814	4221	519	513	601	604
	CPU	0.6008	1.9789	5.9054	0.8973	1.0956	1.3002	1.7828
	RRN	9.8184e-07	9.6911e-07	9.7485e-07	9.4518e-07	8.8198e-07	9.7133e-07	9.3872e-07
RGRK[4]	IT	7473	8488	8284	1364	1275	1471	1429
	CPU	2.7320	10.7485	60.3266	2.8373	3.1983	3.9460	5.3558
	RRN	9.8624e-07	9.9567e-07	9.4480e-07	9.6902e-07	9.5563e-07	9.6091e-07	9.9939e-07
ME-RGRK[33]	IT	31740	31771	34862	5262	5230	6043	6123
	CPU	24.2748	81.2881	303.4287	39.8761	27.7110	32.3563	45.1065
	RRN	9.9564e-07	9.9371e-07	9.9855e-07	9.9970e-07	9.9029e-07	9.8253e-07	9.8513e-07
$\theta = 0.5$								
QGRK	IT	3085	3693	3996	486	483	592	605
	CPU	0.6083	1.8341	5.0547	0.8274	1.0504	1.3371	1.7618
	RRN	9.8538e-07	9.8107e-07	9.9308e-07	9.9488e-07	9.3114e-07	9.2450e-07	8.9996e-07
GRK[3]	IT	7378	8310	9303	1272	1147	1427	1371
	CPU	2.8582	31.3428	41.8742	2.6255	2.9954	3.9049	4.6993
	RRN	9.9593e-07	9.8662-07	9.7479e-07	9.6261e-07	9.6044e-07	9.5906e-07	9.9590e-07
ME-RGRK[33]	IT	2917	36254	37345	5131	4953	5826	5362
	CPU	26.3773	289.3727	320.1233	21.5449	26.2200	31.6292	38.2307
	RRN	9.9816e-07	9.9980e-07	9.9987e-07	9.9930e-07	9.9726e-07	9.9988e-07	9.8006e-07
$\theta = 0.7$								
QRGRK	IT	3050	3575	3987	505	476	603	571
	CPU	0.6670	1.8352	5.3131	0.8690	0.9964	1.2829	1.6665
	RRN	9.7387e-07	9.9530e-07	9.8770e-07	9.0228e-07	9.9257e-07	9.2595e-07	9.7616e-07
RGRK[4]	IT	7328	9087	9492	1127	1153	1309	1259
	CPU	2.6754	11.4099	49.2162	2.4421	3.0025	3.5138	4.2124
	RRN	9.8519e-07	9.8567e-07	9.9942e-07	9.4977e-07	8.9189e-07	9.9545e-07	9.9447e-07
ME-RGRK[33]	IT	29625	35550	39516	4520	4445	5710	5329
	CPU	26.9594	83.1763	336.2669	19.2479	24.4543	30.7064	64.0556
	RRN	9.9376e-07	9.9670e-07	9.9649e-07	9.9840e-07	9.8488e-07	9.6102e-07	9.7262e-07
$\theta = 1$								
QRGRKIMAX	IT	3087	3528	4032	481	459	569	551
	CPU	0.5220	1.6595	5.3536	0.7292	0.9713	1.2168	1.5675
	RRN	9.8615e-07	9.7927e-07	9.9905e-07	8.6950e-07	9.2455e-07	9.9151e-07	9.8825e-07
RGRK[4]	IT	7349	7857	8942	1057	989	1215	1181
	CPU	2.7252	8.8195	20.1882	2.2184	2.5596	3.2823	4.1435
	RRN	9.9130e-07	9.9916e-07	9.9168e-07	9.7363e-07	9.6634e-07	9.9632e-07	9.8943e-07
ME-MWRK[33]	IT	26756	37660	35568	4420	3943	5318	4621
	CPU	24.2321	87.0046	153.6180	18.0416	19.9316	29.2785	32.2381
	RRN	9.9789e-07	9.9867e-07	9.9755e-07	9.9658e-07	9.9421e-07	9.9882e-07	9.7217e-07

5.3. Image restoration

In the following, we show an application of our results in the color image restoration problem.

Example 5.4. *Let $\mathbf{X} = X_1\mathbf{i} + X_2\mathbf{j} + X_3\mathbf{k} \in \mathbb{Q}^{s\times s}$ be the desired unknown true color image, where $X_1, X_2, X_3 \in \mathbb{R}^{s\times s}$ denote the red, green, and blue channels. Let $\mathbf{x} = \mathrm{vec}(\mathbf{X}) \in \mathbb{Q}^{s^2}$ be the quaternion vector obtained by stacking the columns of $\mathbf{X}$, and $\mathbf{A} \in \mathbb{Q}^{s^2\times s^2}$ denote the blurring matrix. Such color image is contaminated by $\mathbf{A}$ giving the blurred color image $\mathbf{C} = C_1\mathbf{i} + C_2\mathbf{j} + C_3\mathbf{k} \in \mathbb{Q}^{s\times s}$. Denote $\mathbf{c} = \mathrm{vec}(\mathbf{C}) \in \mathbb{Q}^{s^2}$. Then the model of color image formation can be modelled as the quaternion linear systems* (1).



(a) $m = n = 400$

(b) $m = n = 600$

(c) $m = n = 800$

Figure 4. The numerical results for Example 5.3 using QRGRK, RGRK and ME-RGRK methods with $\theta = 0.7$.

Let $\mathbf{A} = A_1\mathrm{i} + A_2\mathrm{j} + A_3\mathrm{k} \in \mathbb{Q}^{s^2 \times s^2}$ with A_1, A_2, A_3 being real Gaussian Toeplitz matrix whose entries are given by

$$t_{ij} = \begin{cases} \dfrac{1}{\sqrt{2\pi}\sigma} \exp(-\dfrac{(i-j)^2}{2\sigma^2}), & |i-j| \leq r, \\ 0, & \text{otherwise}. \end{cases} \tag{38}$$

We run QRGRK method to restore the blurred color images including "House", "Female" and "Peppers" from the USC-SIPI image database[1)]. Figure 6 shows the restored results of three color images. The original color images, blurred color images and restored color images obtained by QRGRK method are located in the first, second and third columns of Figure 6, respectively. It is seen that the restoration quality is acceptable.

[1)] http://sipi.usc.edu/database/

Table 4. Numerical comparison results for Example 5.3.

(m, n)		(400,400)	(600,600)	(800,800)	(150,4000)	(150,5000)	(200,4000)	(200,5000)
QRK	IT	7975	11041	12417	1732	1839	3204	2690
	CPU	1.1673	5.1081	15.5996	2.7021	3.8879	6.4181	7.0641
	RRN	9.7490e-07	9.8519e-07	9.9954e-07	9.7730e-07	9.8741e-07	9.9630e-07	9.9905e-07
$\theta = 0.3$								
PmQRGRK	IT	2769	3001	3674	415	383	520	519
	CPU	0.5470	1.5436	4.9573	0.7255	0.8424	1.9899	1.5248
	RRN	9.5784e-07	9.5348e-07	9.4351e-07	9.4466e-07	8.1935e-07	9.4935e-07	7.7198e-07
QRGRK	IT	3791	3810	4073	509	479	638	587
	CPU	0.6014	1.9770	5.3295	0.8817	1.1074	1.4014	1.6365
	RRN	9.8868e-07	9.3621e-07	9.9770e-07	9.9716e-07	9.9815e-07	9.7358e-07	9.9192e-07
$\theta = 0.5$								
PmQGRK	IT	2492	3068	3351	395	372	491	481
	CPU	0.3801	1.5960	4.3417	0.06502	0.8234	1.0677	1.4005
	RRN	9.8330e-07	9.3908e-07	9.2074e-07	9.6984e-07	9.5082e-07	9.5502e-07	9.1900e-07
QGRK	IT	3083	3757	3940	491	483	632	560
	CPU	0.4449	1.9591	5.0573	0.7775	0.9951	1.3581	1.5658
	RRN	9.9225e-07	9.9783-07	9.6708e-07	9.8289e-07	9.8492e-07	9.6393e-07	9,9678e-07
$\theta = 0.7$								
PmQRGRK	IT	2547	2960	3767	396	376	457	474
	CPU	0.5671	1.4062	4.8722	0.6559	0.7724	1.0267	1.3802
	RRN	9.8271e-07	9.7309e-07	9.9916e-07	9.7074e-07	9.9353e-07	8.9072e-07	9.0640e-07
QRGRK	IT	3240	3800	3903	489	468	611	570
	CPU	0.5671	1.4062	4.8722	0.7783	0.9889	1.2817	1.6237
	RRN	9.8725e-07	9.8119e-07	9.7315e-07	9.6421e-07	9.9135e-07	9.2956e-07	9.4574e-07
$\theta = 1$								
PmQRGRKIMAX	IT	2620	3055	3370	365	368	435	449
	CPU	0.4689	1.3410	4.3001	0.6449	0.7160	0.9483	1.2842
	RRN	9.8270e-07	9.9103e-07	9.6539e-07	9.3200e-07	8.7255e-07	9.9182e-07	9.9493e-07
QRGRKIMAX	IT	3137	3415	3800	514	459	587	533
	CPU	0.5261	1.5030	4.8748	0.7732	0.9346	1.2774	1.5141
	RRN	9.9028e-07	9.9253e-07	9.8171e-07	9.8799e-07	9.6562e-07	9.1252e-07	9.8280e-07

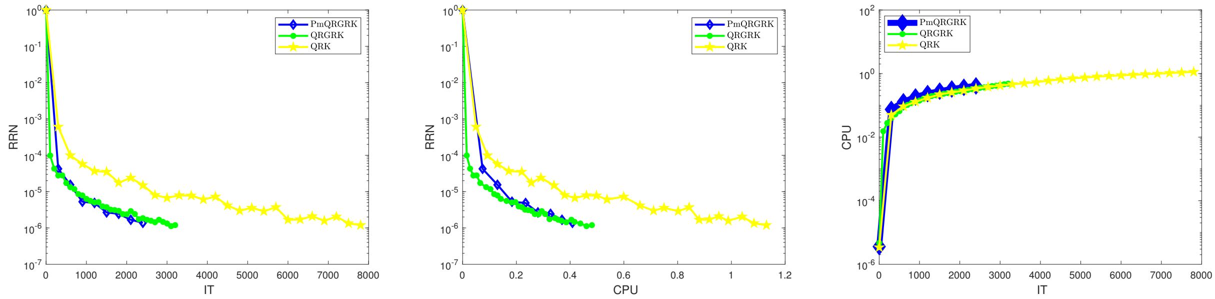
We also measure the recovered quality by the relative error (RE) and peak signal-to-noise ratio (PSNR), which are given by

$$\mathrm{RE}(\mathbf{X}_{\text{restored}}, \mathbf{X}_{\text{true}}) = \frac{\|\mathbf{X}_{\text{restored}} - \mathbf{X}_{\text{true}}\|}{\|\mathbf{X}_{\text{true}}\|}$$

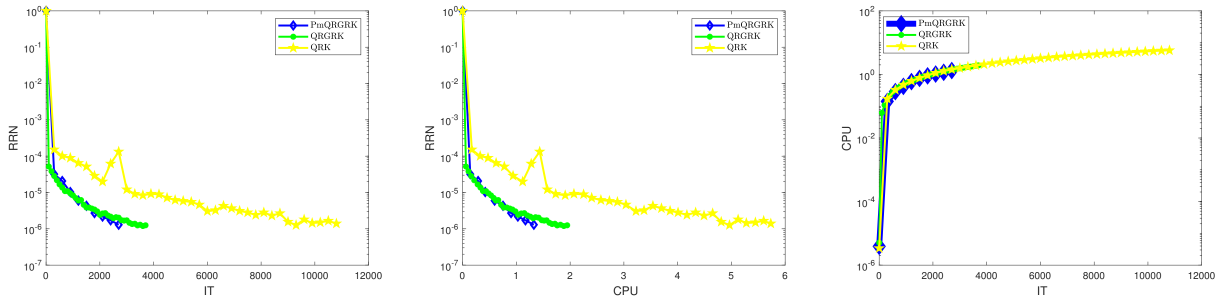
and

$$\mathrm{PSNR}(\mathbf{X}_{\text{restored}}, \mathbf{X}_{\text{true}}) = 10 \log_{10} \frac{3s^2 \times 255^2}{\|\mathbf{X}_{\text{restored}} - \mathbf{X}_{\text{true}}\|^2}.$$

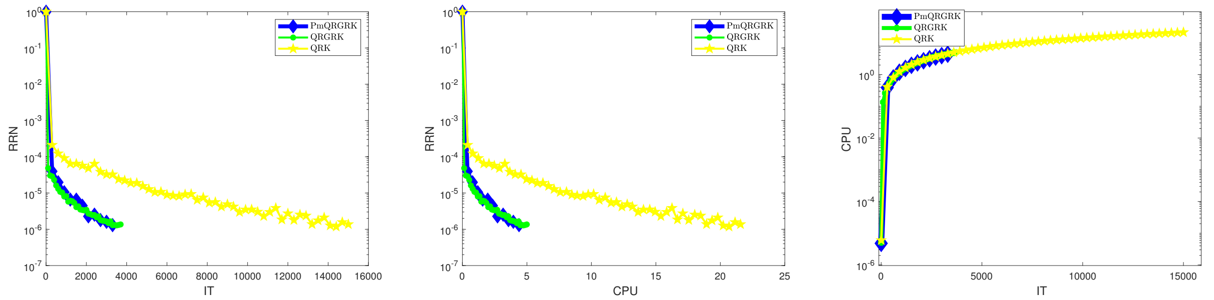
The corresponding results are reported in Table 5. We see from Table 5 that QRGRK method is feasible and effective for the color image restorations.



(a) $m = n = 400$



(b) $m = n = 600$



(c) $m = n = 800$

Figure 5. The numerical results for Example 5.3 using PmQRGRK, QRGRK and QRK methods with $\theta = 0.7$.

Table 5. Numerical results for Example 5.4.

Image	IT	Time	RRN	PSNR	RE
House	51732	295.1598	9.9987e-06	25.4816	0.0495
Female	67438	381.4083	9.9939e-06	29.4407	0.0646
Peppers	59559	349.7263	9.9903e-06	26.0788	0.0527

(a) Original (b) Blurred (c) QRGRK

Figure 6. The restoration results for Example 5.4.

6. Conclusions

In this paper, by using a relaxed greedy probability selection criterion, we propose a structure preserving QRGRK algorithm for the quaternion linear systems (1) and establish its asymptotic convergence theory. In order to further improve the convergence performance of the structure preserving QRGRK method, we employ Polyak's momentum acceleration technique and present the structure preserving PmQRGRK method. Our convergence results provide a theoretical guarantee for the convergence of the PmQRGRK method with constant stepsizes and momentum parameters. Numerical experiments show that the structure preserving QRGRK algorithm has more advantages than the traditional RGRK and ME-RGRK in terms of storage and computing time. In addition, numerical results demonstrate that PmQRGRK is superior to QRGRK method in terms of iteration counts and computing time.

Acknowledgment

We would like to thank the editor and anonymous referees for their valuable comments and suggestions which have considerably improved this paper.

References

[1] C. Brezinski, *Projection Methods for Systems of Equations*, Elsevier Science B.V., 1997.
[2] Z. Z. Bai, X. G. Liu, *On the Meany inequality with applications to convergence analysis of several row-action iteration methods*, Numer. Math. **124** (2013), 215–236.
[3] 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.
[4] Z. Z. Bai, W. T. Wu, *On relaxed greedy randomized Kaczmarz methods for solving large sparse linear systems*, Appl. Math. Lett. **83** (2018), 21–26.
[5] S. De Leo, G. Scolarici, *Right eigenvalue equation in quaternionic quantum mechanics*, J. Phys. A Math. Gen. **33** (2000), 2971–2995.
[6] X. L. Fan, Y. Li, Z. H. Liu, J.L. Zhao, *Solving quaternion linear system based on semi-tensor product of quaternion matrices*, Symmetry-Basel **14** (2022), 1359–1379.
[7] R. Ghiloni, V. Moretti, A. Perotti, *Continuous slice functional calculus in quaternionic Hilbert spaces*, Rev. Math. Phys. **25** (2013), 1350006.
[8] L. Huang, Q. W. Wang, Y. Zhang, *The Moore-Penrose inverses of matrices over quaternion polynomial rings*, Linear Algebra Appl. **475** (2015), 45–61.
[9] Z. G. Jia, *The Eigenvalue Problem of Quaternion Matrix: Structure-Preserving Algorithms and Applications*, Science Press, Beijing, 2019.
[10] Z. G. Jia, M. K. Ng, *Structure preserving quaternion generalized minimal residual method*, SIAM J. Matrix Anal. A. **42** (2021), 616–634.
[11] Z. G. Jia, M. K. Ng , G. J. Song, *Robust quaternion matrix completion with applications to image inpainting*, Numer. Linear Algebr. **26** (2019), e2245.
[12] Z. G. Jia, M. S. Wei, S. T. Ling, *A new structure-preserving method for quaternion Hermitian eigenvalue problems*, J. Comput. Appl. Math. **239** (2013), 12–24.
[13] Z. G. Jia, M. S. Wei, M. X. Zhao, Y. Chen, *A new real structure-preserving quaternion QR algorithm*, J. Comput. Appl. Math. **343** (2018), 26–48.
[14] S. Kaczmarz, *Angenäherte Auflösung von Systemen linearer Gleichungen*, Bull. Int. Acad. Pol. Sci. Lett. A **35** (1937), 355–357.
[15] C. Lanczos, *Solution of systems of linear equations by minimized iterations*, J. Res. Nat. Bur. Standards **49** (1952), 33–53.
[16] Y. Liu, C. Q. Gu, *On greedy randomized block Kaczmarz method for consistent linear systems*, Linear Algebra Appl. **616** (2021), 178–200.
[17] N. Loizou, P. Richtárik, *Momentum and stochastic momentum for stochastic gradient*, newton, proximal point and subspace descent methods, Comput. Optim. Appl. **77** (2020), 653–710.
[18] Z. Liu, F. Zhang, C. Ferreira, Y. Zhang, *On circulant and skew-circulant splitting algorithms for continuous Sylvester equations*, Comput. Math. Appl. **109** (2022), 30–43.
[19] Y. Li, Y. Z. Zhang, M. S. Wei, H. Zhao, *Real structure-preserving algorithm for quaternion equality constrained least squares problem*, Math. Method. Appl. Sci. **43** (2020), 4558–4566.
[20] A. Ma, D. Needell, A. Ramdas, *Convergence properties of the randomized extended Gauss-Seidel and Kaczmarz methods*, SIAM J. Matrix Anal. Appl. **36** (2015), 1590–1604.
[21] I. Necoara, *Faster randomized block Kaczmarz algorithms*, SIAM J. Matrix Anal. A. **40** (2019), 1425–1452.
[22] Y. Q. Niu, B. Zheng, *On the randomized block Kaczmarz algorithms for solving matrix equation $AXB = C$*, J. Comput. Appl. Math. **460** (2025), 116421.
[23] B. T. Polyak, *Some methods of speeding up the convergence of iteration methods*, USSR Computational Mathematics and Mathematical Physics **4** (1964), 1–17.
[24] T. Parcollet, M. Morchid, G. Linarès, *A survey of quaternion neural networks*, Artif. Intell. Rev. **53** (2020), 2957–2982.
[25] L. Qi, Z. Luo, Q. Wang, X. Zhang, *Quaternion matrix optimization and the underlying calculus*, 2020. arXiv: 2009.13884.
[26] S. H. Strogatz, *Nolinear dynamics and chaos: with applications to physics, Biology, Chemistry and Engineering*, Westview Press, Boulder, Co, 2001.
[27] Y. Saad, *Iterative Methods for Sparse Linear Systems*, SIAM, Philadelphia, 2003.
[28] S. G. Shafiei, M. Hajarian M, *Developing Kaczmarz method for solving Sylvester matrix equations*, J. Franklin I. **359** (2022), 8991–9005.
[29] T. Sogabe, M. Sugihara, S. L. Zhang, *An extension of conjugate residual method to nonsymmetric linear systems*, J. Comput. Appl. Math. **226** (2009), 103–113.
[30] T. Strohmer, R. Vershynin, *A randomized Kaczmarz algorithm with exponential convergence*, J. Fourier. Anal. Appl. **15** (2009), 262–278.
[31] C. C. Took, D. P. Mandic, *The quaternion LMS algorithm for adaptive filtering of hypercomplex processes*, IEEE T. Signal Proces. **57** (2009), 1316–1327.
[32] Z. L. Tian, M. Y. Tian, Z. Y. Liu, T. Y. Liu, *The Jacobi and Gauss-Seidel-type iteration methods for the matrix equation $AXB = C$*, Appl. Math. Comput. **292** (2017), 63–75.
[33] N. C. Wu, C. Z. Liu, Q. Zuo, *On the Kaczmarz methods based on relaxed greedy selection for solving matrix equation $AXB = C$*, J. Comput. Appl. Math. **413** (2022), 114374.
[34] M. S. Wei, Y. Li, F. X. Zhang, J. L. Zhao, *Quaternion matrix computations*, Nova Science Publishers, 2018.
[35] L. Wen, F. Yin, Y. M. Liao, G. X. Huang, *A greedy average block Kaczmarz method for the large scaled consistent system of linear equations*. AIMS Math. **7** (2022), 6792–6806.
[36] N. C. Wu, Y. Zhou, Z. L. Tian, *On the relaxed greedy randomized Kaczmarz methods with momentum acceleration for solving matrix equation $AXB = C$*, 2023. arXiv: 2301.12753.
[37] S. F. Yuan, Q. W. Wang, X. F. Duan, *On solutions of the quaternion matrix equation $AX = B$ and their applications in color image restoration*, Appl. Math. Comput. **221** (2013), 10–20.
[38] F. Zhang, *Quaternions and matrices of quaternions*, Linear Algebra Appl. **21** (1997), 21–57.
[39] A. Zouzias, N. M. Freris, *Randomized extended Kaczmarz for solving least squares*, SIAM J. Matrix Anal. A. **34** (2013), 773–793.
[40] Y. Zeng, D. R. Han, Y. S. Su, J. X. Xie, *On adaptive stochastic heavy ball momentum for solving linear systems*, SIAM J. Matrix Anal. A. **45** (2024), 1259–1286.