Article

A Fast Projected Gradient Algorithm for Quaternion Hermitian Eigenvalue Problems

Shan-Qi Duan and Qing-Wen Wang [*]

Posted Date: 7 March 2025

doi: 10.20944/preprints202503.0469.v1

Keywords: GHR calculus; Principal eigenvalues; Quaternion Hermitian matrix; Nesterov's accelerated gradient method

Article

A Fast Projected Gradient Algorithm for Quaternion Hermitian Eigenvalue Problems

Shan-Qi Duan [1] **and Qing-Wen Wang** [1,2,*]

[1] Department of Mathematics and Newtouch Center for Mathematics, Shanghai University, Shanghai 200444, China; shanqiduan@shu.edu.cn
[2] Collaborative Innovation Center for the Marine Artificial Intelligence, Shanghai 200444, China
[*] Correspondence: wqw@t.shu.edu.cn

Abstract: In this paper, based on the novel generalized Hamilton-real (GHR) calculus, we propose for the first time a quaternion Nesterov's accelerated projected gradient algorithm for computing the dominant eigenvalue and eigenvector of quaternion Hermitian matrices. By introducing momentum terms and look-ahead updates, the algorithm achieves a faster convergence rate. We theoretically prove the convergence of the quaternion Nesterov's accelerated projected gradient algorithm. Numerical experiments show that the proposed method outperforms the quaternion projected gradient ascent method and the traditional algebraic methods in terms of computational accuracy and runtime efficiency.

Keywords: GHR calculus; principal eigenvalues; quaternion hermitian matrix; Nesterov's accelerated gradient method

MSC: 15A18; 15B57; 65F10; 65F15

1. Introduction

In 1843, Sir William Rowan Hamilton [9] introduced quaternions in an effort to expand the concept of complex numbers into spaces of higher dimensions. Quaternions and quaternion matrices play a critical role in many applications, such as quantum mechanics, computer graphics, quaternion principal component analysis (QPCA) and image processing [2,11,23,24]. Due to the non-commutative property of quaternion multiplication, the eigenvalues of quaternion matrices are distinguished into left and right types, with the right eigenvalue problem having garnered widespread attention[4,6,14,16,25].

In recent years, a series of numerical methods have been developed to compute the eigenvalues of quaternion matrices, particularly focusing on the eigenvalue problems of Hermitian matrices. These numerical methods can be broadly categorized into three classes: The first class involves direct quaternion arithmetic operations. For instance, Bunse-Gerstner proposed a quaternion QR algorithm for solving the right eigenvalue problem of quaternion matrices [1]. However, due to the complexity of quaternion arithmetic, this algorithm requires significant computational effort. The second class is based on the real or complex counterparts of quaternion matrices. By studying the real or complex counterpart structures and properties of quaternion matrices, and leveraging stable orthogonal transformations, real or complex structure-preserving methods have been developed to solve the right eigenvalue problem of quaternion Hermitian matrices [8,10]. The third class is based on the real counterparts of quaternion matrices, leading to the development of numerous structure-preserving iterative algorithms. Examples include the explicitly restarted quaternion Arnoldi Method (ERQAM) [18], designed to compute standard right eigenpairs of general quaternion matrices, and a novel quaternion power method introduced in [13] for computing the dominant standard right eigenvalue and its corresponding eigenvector. Structure-preserving methods exhibit significant advantages in terms of storage space and computational efficiency.

In the field of quaternion optimization, significant progress has been made with the generalized HR calculus (GHR) [20–22]. GHR leverages quaternion rotations within a general orthogonal system, offering a way to compute the derivatives and gradients of functions with quaternion variables, thereby providing a solid theoretical foundation for the development of quaternion optimization methods. Subsequently, based on the generalized HR calculus (GHR), Diao et al. [3] proposed a gradient projection algorithm for maximizing the quaternion Rayleigh quotient under unit constraints. This algorithm demonstrated good performance and contributed to the development of quaternion optimization algorithms.

In this paper, we first equivalently transform the principal eigenvalue problem of quaternion Hermitian matrices into a maximization optimization problem over the quaternion skew field. Leveraging generalized HR calculus, we propose a quaternion Nesterov's accelerated gradient projection algorithm to solve it. Subsequently, we conduct a convergence analysis of the quaternion Nesterov's accelerated gradient projection algorithm, proving that a real differentiable function with Lipschitz continuous gradient possesses a quadratic upper bound. Furthermore, we demonstrate that the algorithm possesses a convergence rate of $(O(1/t^2)$. Finally, we compare our algorithm with two other methods, and numerical experiments indicate that our algorithm exhibits superior performance in terms of both accuracy and time efficiency.

The rest of this paper is organized as follows. Section 2 introduces some basic notations and fundamental properties of quaternions, including definitions of quaternion modulus, similarity, and rotation, with a particular emphasis on reviewing the relevant definitions and properties of generalized HR integrals. In Section 3, we design a quaternion Nesterov accelerated gradient projection algorithm to solve the principal eigenvalue and corresponding eigenvector of quaternion Hermitian matrices. Section 4 provides a convergence analysis of the quaternion Nesterov accelerated gradient projection algorithm. In Section 5, we conduct numerical experiments to validate the proposed method. Finally, in Section 6, we summarize this paper.

2. Preliminaries

In this section, some quaternion notations and basic definitions are introduced, which will be used in the rest of the paper.

2.1. Notations

Throughout this paper, to distinguish scalars, vectors, real or complex matrices and quaternion matrices, scalars will be denoted by lower case Greek letters, e.g., α, β, quaternions will be denoted by lowercase letters, e.g., $\mathbf{p}, \mathbf{q}$, and quaternion vectors are denoted by $\mathbf{x}, \mathbf{y}$, real or complex matrices will be defined by uppercase letters, e.g., A, B, and quaternion matrices will be denoted by bold uppercase letters, e.g., $\mathbf{A}, \mathbf{B}$. $\mathbf{I}$ denotes the $n \times n$ identity quaternion matrix. The operators $(\cdot)^T$ and $(\cdot)^H$ represent transpose and conjugate transpose, respectively. The MATLAB function command will be denoted by typewriter letters, e.g., $[V, D] = \texttt{eig}(A)$.

2.2. Quaternions and Quaternion Matrices

Denote the set of quaternions as

$$\mathbb{Q} = \text{span}\{1, \mathbf{i}, \mathbf{j}, \mathbf{k}\} \triangleq \{\mathbf{q} = q_1 + q_2\mathbf{i} + q_3\mathbf{j} + q_4\mathbf{k} \mid q_1, q_2, q_3, q_4 \in \mathbb{R}\}, \tag{1}$$

where $\mathbf{i}, \mathbf{j}, \mathbf{k}$ are three imaginary units of quaternions, satisfying

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

The scalar (real) part of $\mathbf{q}$ is denoted by $\Re(\mathbf{q}) = q_1$. And the vector (imaginary) part of $\mathbf{q}$ is denoted by $\Im(\mathbf{q}) = q_2\mathbf{i} + q_3\mathbf{j} + q_4\mathbf{k}$. A quaternion is called imaginary when its real part is equal to zero. The multiplication of quaternions adheres to the distributive law but is noncommutative.

The zero element in $\mathbb{Q}$ is $\mathbf{0} = 0 + 0\mathbf{i} + 0\mathbf{j} + 0\mathbf{k}$ and the unit element is $\mathbf{1} = 1 + 0\mathbf{i} + 0\mathbf{j} + 0\mathbf{k}$. For any $\mathbf{q} = q_0 + q_1\mathbf{i} + q_2\mathbf{j} + q_3\mathbf{k} \in \mathbb{Q}$, the conjugate of a quaternion is defined as

$$\mathbf{q}^* = q_0 - q_1\mathbf{i} - q_2\mathbf{j} - q_3\mathbf{k}.$$

The magnitude of $\mathbf{q}$ is $|\mathbf{q}| = \sqrt{\mathbf{q}^*\mathbf{q}} = \sqrt{q_0^2 + q_1^2 + q_2^2 + q_3^2}$, it follows that the inverse of a nonzero quaternion $\mathbf{q}$ is given by $\mathbf{q}^{-1} = \mathbf{q}^*/|\mathbf{q}|^2$.

Two quaternions $\mathbf{a}$ and $\mathbf{b}$ are said to be similar if there exists a nonzero quaternion $\mathbf{c}$ such that $\mathbf{c}^{-1}\mathbf{a}\mathbf{c} = \mathbf{b}$, this is written as $\mathbf{a} \sim \mathbf{b}$. Obviously, $\mathbf{a}$ and $\mathbf{b}$ are similar if and only if there is a unit quaternion $\mathbf{d}$ such that $\mathbf{d}^{-1}\mathbf{a}\mathbf{d} = \mathbf{b}$, and two similar quaternions have the same norm. It is routine to check that $\sim$ is an equivalence relation on the quaternions. We denote by $[\mathbf{a}]$ the equivalence class containing $\mathbf{a}$. If $\mathbf{q} = q_0 + q_1\mathbf{i} + q_2\mathbf{j} + q_3\mathbf{k} \in \mathbb{Q}$, then $\mathbf{q}$ and $q_0 + \sqrt{q_1^2 + q_2^2 + q_3^2}\mathbf{i}$ are similar, namely, $\mathbf{q} \in \left[q_0 + \sqrt{q_1^2 + q_2^2 + q_3^2}\mathbf{i}\right]$.

Quaternions can also be expressed in polar form as $\mathbf{q} = |\mathbf{q}|(\cos\theta + \hat{\mathbf{q}}\sin\theta)$, where $\hat{\mathbf{q}} = \Im(\mathbf{q})/|\Im(\mathbf{q})|$ is a pure unit quaternion and $\theta = \arccos(\Re(\mathbf{q})/|\mathbf{q}|) \in \mathbb{R}$ denotes the angle (or argument) of the quaternion. Next, we will introduce the quaternion rotation and involution operators.

Definition 1 (Quaternion rotation[19])**.** *For any quaternion* $\mathbf{q}$, *the transformation*

$$\mathbf{q}^{\mu} \triangleq \mu\mathbf{q}\mu^{-1} \tag{2}$$

geometrically describes a three-dimensional rotation of the vector part of $\mathbf{q}$ *by an angle* 2θ *about the vector part of* μ, *where* $\mu = |\mu|(\cos\theta + \hat{\mu}\sin\theta)$ *is any nonzero quaternion.*

Specifically, if μ in (2) is an imaginary unit, then the quaternion rotation (2) reduces to quaternion involution [5], defined by

$$\begin{aligned}\mathbf{q}^{\mathbf{i}} &= -\mathbf{i}\mathbf{q}\mathbf{i} = q_0 + \mathbf{i}q_1 - \mathbf{j}q_2 - \mathbf{k}q_3,\\ \mathbf{q}^{\mathbf{j}} &= -\mathbf{j}\mathbf{q}\mathbf{j} = q_0 - \mathbf{i}q_1 + \mathbf{j}q_2 - \mathbf{k}q_3,\\ \mathbf{q}^{\mathbf{k}} &= -\mathbf{k}\mathbf{q}\mathbf{k} = q_0 - \mathbf{i}q_1 - \mathbf{j}q_2 + \mathbf{k}q_3,\end{aligned}$$

where $\mathbf{q} = q_0 + \mathbf{i}q_1 + \mathbf{j}q_2 + \mathbf{k}q_3 \in \mathbb{Q}$. Below, we will list some properties of quaternion rotation, including

$$(\mathbf{p}\mathbf{q})^{\mu} = \mathbf{p}^{\mu}\mathbf{q}^{\mu}, \quad \mathbf{p}\mathbf{q} = \mathbf{q}^{\mathbf{p}}\mathbf{p} = \mathbf{q}\mathbf{p}^{(\mathbf{q}^*)}, \quad \forall \mathbf{p}, \mathbf{q} \in \mathbb{Q}$$

and

$$\mathbf{q}^{\mu\nu} = (\mathbf{q}^{\nu})^{\mu}, \quad \mathbf{q}^{\mu *} \triangleq (\mathbf{q}^*)^{\mu} = (\mathbf{q}^{\mu})^* \triangleq \mathbf{q}^{*\mu}, \quad \forall \nu, \mu \in \mathbb{Q}.$$

Note that the representation in (1) can be extended to a general orthogonal basis $\{1, \mathbf{i}^{\mu}, \mathbf{j}^{\mu}, \mathbf{k}^{\mu}\}$, where the following properties hold [19]:

$$\mathbf{i}^{\mu}\mathbf{i}^{\mu} = \mathbf{j}^{\mu}\mathbf{j}^{\mu} = \mathbf{k}^{\mu}\mathbf{k}^{\mu} = \mathbf{i}^{\mu}\mathbf{j}^{\mu}\mathbf{k}^{\mu} = -1.$$

Denote the set of quaternion matrices as

$$\mathbb{Q}^{m\times n} = \left\{\mathbf{A} = A_0 + A_1\mathbf{i} + A_2\mathbf{j} + A_3\mathbf{k} \mid A_0, A_1, A_2, A_3 \in \mathbb{R}^{m\times n}\right\}.$$

The conjugate transpose of $\mathbf{A}$ is $\mathbf{A}^* = A_0 - A_1\mathbf{i} - A_2\mathbf{j} - A_3\mathbf{k}$. We say that a square quaternion matrix $\mathbf{A} \in \mathbb{Q}^{n\times n}$ is normal if $\mathbf{A}^*\mathbf{A} = \mathbf{A}\mathbf{A}^*$; Hermitian if $\mathbf{A}^* = \mathbf{A}$, i.e. $A_0 = A_0^{\top}$ and $A_i = -A_i^{\top}, i = 1, 2, 3$; Unitary if $\mathbf{A}^*\mathbf{A} = \mathbf{I}$, where $\mathbf{I}$ is the identity matrix; Invertible (nonsingular) if there exists a matrix

$\mathbf{B} \in \mathbb{Q}^{n\times n}$ such that $\mathbf{AB} = \mathbf{BA} = \mathbf{I}$. In this case, we denote $\mathbf{A}^{-1} = \mathbf{B}$. We have $(\mathbf{AB})^{-1} = \mathbf{B}^{-1}\mathbf{A}^{-1}$ if $\mathbf{A}$ and $\mathbf{B}$ are invertible, and $(\mathbf{A}^*)^{-1} = (\mathbf{A}^{-1})^*$ if $\mathbf{A}$ is invertible.

2.3. GHR calculus

We now introduce the generalized HR derivatives which comprise both the product and chain rules, see [20,22] for more details.

Definition 2 (real-differentiability [20])**.** *Let $\mathbf{q} = q_a + q_b\mathbf{i} + q_c\mathbf{j} + q_d\mathbf{k} \in \mathbb{Q}$, then a function $f(\mathbf{q}) = f_a(q_a, q_b, q_c, q_d) + f_b(q_a, q_b, q_c, q_d)\mathbf{i} + f_c(q_a, q_b, q_c, q_d)\mathbf{j} + f_d(q_a, q_b, q_c, q_d)\mathbf{k}$ is called real differentiable when $f_a(q_a, q_b, q_c, q_d), f_b(q_a, q_b, q_c, q_d), f_c(q_a, q_b, q_c, q_d)$ and $f_d(q_a, q_b, q_c, q_d)$ are differentiable with respect to the real variables q_a, q_b, q_c and q_d, respectively.*

Definition 3 (GHR derivatives [20])**.** *If $f : \mathbb{Q} \to \mathbb{Q}$ is real differentiable, then the GHR derivatives of $f(\mathbf{q})$ with respect to $\mathbf{q}^\mu$ and $\mathbf{q}^{\mu*}$ $(0 \neq \mu \in \mathbb{Q})$ are defined as*

$$\frac{\partial f}{\partial \mathbf{q}^\mu} = \frac{1}{4}\left(\frac{\partial f}{\partial q_a} - \frac{\partial f}{\partial q_b}\mathbf{i}^\mu - \frac{\partial f}{\partial q_c}\mathbf{j}^\mu - \frac{\partial f}{\partial q_d}\mathbf{k}^\mu\right),$$

and

$$\frac{\partial f}{\partial \mathbf{q}^{\mu*}} = \frac{1}{4}\left(\frac{\partial f}{\partial q_a} + \frac{\partial f}{\partial q_b}\mathbf{i}^\mu + \frac{\partial f}{\partial q_c}\mathbf{j}^\mu + \frac{\partial f}{\partial q_d}\mathbf{k}^\mu\right),$$

where $\mathbf{q} = q_a + q_b\mathbf{i} + q_c\mathbf{j} + q_d\mathbf{k} \in \mathbb{Q}$, $q_a, q_b, q_c, q_d \in \mathbb{R}$, $\partial f/\partial q_a, \partial f/\partial q_b, \partial f/\partial q_c$ and $\partial f/\partial q_d$ are the partial derivatives of f with respect to q_a, q_b, q_c and q_d, while the set $\{1, \mathbf{i}^\mu, \mathbf{j}^\mu, \mathbf{k}^\mu\}$ is an orthogonal basis of $\mathbb{Q}$.

Definition 4 (Quaternion gradient[20])**.** *Let $f(\tilde{\mathbf{q}}) : \mathbb{Q}^{n\times 1} \to \mathbb{Q}$ and $\tilde{\mathbf{q}} = (\mathbf{q}_1, \mathbf{q}_2, \ldots, \mathbf{q}_n) \in \mathbb{Q}^{n\times 1}$, then the two quaternion gradients of f are defined as*

$$\nabla_{\tilde{\mathbf{q}}} f \triangleq \left(\frac{\partial f}{\partial \tilde{\mathbf{q}}}\right)^T = \left(\frac{\partial f}{\partial \mathbf{q}_1}, \ldots, \frac{\partial f}{\partial \mathbf{q}_n}\right)^T \in \mathbb{Q}^{n\times 1}$$

and

$$\nabla_{\tilde{\mathbf{q}}^*} f \triangleq \left(\frac{\partial f}{\partial \tilde{\mathbf{q}}^*}\right)^T = \left(\frac{\partial f}{\partial \mathbf{q}_1^*}, \ldots, \frac{\partial f}{\partial \mathbf{q}_n^*}\right)^T \in \mathbb{Q}^{n\times 1}.$$

Based on the definitions of GHR provided above, we consider a simple quadratic function $f(\mathbf{x}) = \mathbf{x}^H\mathbf{A}\mathbf{x}$, where $\mathbf{x} \in \mathbb{Q}^n$ and $\mathbf{A} \in \mathbb{Q}^{n\times n}$ is a quaternion Hermitian matrix, then the gradient of this function f is given by

$$\nabla_{\mathbf{x}} f(\mathbf{x}) = \frac{1}{2}(\mathbf{A}\mathbf{x})^*,\ \nabla_{\mathbf{x}^*} f(\mathbf{x}) = \frac{1}{2}\mathbf{A}\mathbf{x},$$

in which $\nabla_{\mathbf{x}^*} f$ is the steepest ascent direction [22].

3. Quaternion Nesterov's Accelerated Gradient (QNAG)

In this section, we will introduce the quaternion Nesterov accelerated projected gradient algorithm. To this end, we first review the definition related to the eigenvalue of quaternion Hermitian matrices.

Definition 5. *Let $\mathbf{A} \in \mathbb{Q}^{n\times n}$ be a quaternion Hermitian matrix. Then $\lambda \in \mathbb{R}$ is called an eigenvalue of $\mathbf{A}$ if there exists a nonzero vector $\mathbf{x} \in \mathbb{Q}^n$ such that*

$$\mathbf{A}\mathbf{x} = \mathbf{x}\lambda.$$

Here $\mathbf{x}$ is called the eigenvector corresponding to the eigenvalue λ.

Due to the non-commutativity of quaternion multiplication, general quaternion square matrices have distinct left and right eigenvalues. However, for a quaternion Hermitian matrix $\mathbf{A}$, if $\mathbf{A}$ has a right eigenvalue λ and its corresponding eigenvector $\mathbf{x}$, it is straightforward to show that

$$\mathbf{x}^H\mathbf{A}\mathbf{x} = \mathbf{x}^H\mathbf{x}\lambda,$$

by dividing both sides of the above equation by $\mathbf{x}^H\mathbf{x}$, we obtain

$$\lambda = \frac{\mathbf{x}^H\mathbf{A}\mathbf{x}}{\mathbf{x}^H\mathbf{x}} = \frac{\mathbf{x}^H\mathbf{A}\mathbf{x}}{\|\mathbf{x}\|^2} \in \mathbb{R}, \tag{3}$$

where (3) represents the Rayleigh quotient on the quaternion skew field. Therefore, the eigenvalues of quaternion Hermitian matrices are all real numbers, and thus there is no distinction between left and right eigenvalues.

Our goal is to compute the principal eigenvalue of a given quaternion Hermitian matrix. By defining the objective function as $f(\mathbf{x}) = \mathbf{x}^H\mathbf{A}\mathbf{x}$ and imposing the normalization constraint $\|\mathbf{x}\|^2 = 1$, we can equivalently transform the problem of finding the principal eigenvalue of a given quaternion Hermitian matrix into the following maximization optimization problem on the quaternion skew-field

$$\begin{aligned} \max_{\mathbf{x}\in\mathbb{Q}^n} & f(\mathbf{x}) = \mathbf{x}^H\mathbf{A}\mathbf{x} \\ \text{s.t. } & \|\mathbf{x}\|^2 = 1. \end{aligned} \tag{4}$$

The above problem (4) can be addressed using the quaternion gradient projection algorithm. Since the introduction of the Nesterov's Accelerated Gradient (NAG) method [15], the incorporation of momentum has become a conventional approach to overcome the shortsighted issue in gradient algorithms [7,12]. To tackle the problem (4), we propose a quaternion Nesterov's accelerated gradient projection algorithm (QNAG). Given an initial point $\mathbf{x}_0$, and set $\mathbf{x}_1 = \mathbf{x}_0$, the QNAG method repeats, for $t \geq 0$,

$$\begin{aligned} \mathbf{y}_t &= \mathbf{x}_t + \beta(\mathbf{x}_t - \mathbf{x}_{t-1}), \\ \mathbf{z}_{k+1} &= \mathbf{y}_t + \alpha\nabla_{\mathbf{y}^*} f(\mathbf{y}_t), \\ \mathbf{x}_{t+1} &= \mathbf{z}_{t+1}/\|\mathbf{z}_{t+1}\|, \end{aligned} \tag{5}$$

where α and β are the step-size and momentum parameters, respectively. When the momentum parameter $\beta = 0$, QNAG simplifies to standard gradient ascent (GA). When $\beta > 0$ it is possible to achieve accelerated rates of convergence for certain combinations of α and β in the deterministic setting. The framework of the proposed algorithm is detailed below.

Remark 1. *After obtaining the principal eigenvalue and its corresponding eigenvector of the quaternion Hermitian matrix, we can employ a deflation technique by updating $A = A - \lambda\mathbf{x}\mathbf{x}^H$ and continue to apply the QNAG algorithm. By repeating this process, all eigenvalues and their corresponding eigenvectors can be obtained.*

4. Convergence Analysis of QNAG

In this section, we will theoretically prove the convergence properties of the quaternion Nesterov's accelerated gradient projection algorithm. We begin our analysis with the following Lemma 1.

Lemma 1. *If $f : \mathbb{Q}^n \to \mathbb{R}$ is a real differentiable and gradient Lipschitz continuous function with constant $L > 0$,*

$$\|\nabla_{\mathbf{x}^*} f(\mathbf{x} + \Delta\mathbf{x}) - \nabla_{\mathbf{x}^*} f(\mathbf{x})\| \leq L\|\Delta\mathbf{x}\|, \ \forall \mathbf{x}, \Delta\mathbf{x} \in \mathbb{Q}^n.$$

Algorithm 1 Quaternion Nesterov's Accelerated Gradient (QNAG).

```
Input: Given a quaternion Hermitian matrix $\mathbf{A} \in \mathbb{Q}^{n\times n}$, step size $0 < \alpha \in \mathbb{R}$, momentum coefficient
    $0 < \beta \in \mathbb{R}$ and the maximum iteration number $I_{\max}$.
Output: The principle eigenvalue $\lambda$ and its corresponding eigenvector $\mathbf{x}_{t+1}$.
Initialize: a unit quaternion vector $\mathbf{x}_0 \in \mathbb{Q}^n$.
$\mathbf{x}_1 = \mathbf{x}_0$.
for $t = 1$ to $I_{\max}$ do
    Momentum extrapolation: $\mathbf{y}_t = \mathbf{x}_t + \beta(\mathbf{x}_t - \mathbf{x}_{t-1})$.
    Compute the gradient at the extrapolation point: $\nabla_{\mathbf{y}^*} f(\mathbf{y}_t) = \frac{1}{2}\mathbf{A}\mathbf{y}_t$.
    Gradient ascent: $\mathbf{z}_{k+1} = \mathbf{y}_t + \alpha \nabla_{\mathbf{y}^*} f(\mathbf{y}_t)$.
    Normalization: $\mathbf{x}_{t+1} = \mathbf{z}_{t+1} / \|\mathbf{z}_{t+1}\|$.
    $\lambda = \mathbf{x}_{t+1}^H \mathbf{A}\mathbf{x}_{t+1}$.
    if $\|A\mathbf{x}_{t+1} - \mathbf{x}_{t+1}\lambda\| < 10^{-10}$ then
        Break
    end if
end for
```

Then, f has the following quadratic upper bound

$$\left| f(\mathbf{y}) - f(\mathbf{x}) - 4\Re\left\{ \nabla_{\mathbf{x}^*} f(\mathbf{x})^H (\mathbf{y} - \mathbf{x}) \right\} \right| \leq 2L\|\mathbf{y} - \mathbf{x}\|^2, \ \forall \mathbf{x}, \mathbf{y} \in \mathbb{Q}^n. \tag{6}$$

Proof. For any $\mathbf{x}, \mathbf{y} \in \mathbb{Q}^n$, let $\Delta\mathbf{x} = \mathbf{y} - \mathbf{x}$. Define the parameterized path $g(t) = f(\mathbf{x} + t\Delta\mathbf{x})$ for $t \in [0, 1]$. Then the difference in the function can be expressed as

$$f(\mathbf{y}) - f(\mathbf{x}) = \int_0^1 \frac{\mathrm{d}}{\mathrm{d}t} g(t) \mathrm{d}t.$$

Using the chain rule, the derivative $g'(t)$ corresponds to the directional derivative of f along $\Delta\mathbf{x}$. In terms of the quaternion gradient [22], this is given by

$$g'(t) = 4\Re\left\{ \nabla_{\mathbf{x}^*} f(\mathbf{x} + t\Delta\mathbf{x})^H \Delta\mathbf{x} \right\}.$$

Thus, the function difference becomes

$$f(\mathbf{y}) - f(\mathbf{x}) = \int_0^1 4\Re\left\{ \nabla_{\mathbf{x}^*} f(\mathbf{x} + t\Delta\mathbf{x})^H \Delta\mathbf{x} \right\} \mathrm{d}t.$$

Subtracting the linear term $4\Re\{\nabla_{\mathbf{x}^*} f(\mathbf{x})^H \Delta\mathbf{x}\}$ from both sides

$$f(\mathbf{y}) - f(\mathbf{x}) - 4\Re\left\{ \nabla_{\mathbf{x}^*} f(\mathbf{x})^H \Delta\mathbf{x} \right\} = \int_0^1 4\Re\left\{ (\nabla_{\mathbf{x}^*} f(\mathbf{x} + t\Delta\mathbf{x}) - \nabla_{\mathbf{x}^*} f(\mathbf{x}))^H \Delta\mathbf{x} \right\} \mathrm{d}t,$$

then using the absolute value inequality and the Cauchy-Schwarz inequality, we have

$$\left| f(\mathbf{y}) - f(\mathbf{x}) - 4\Re\left\{ \nabla_{\mathbf{x}^*} f(\mathbf{x})^H \Delta\mathbf{x} \right\} \right| \leq 4 \int_0^1 \|\nabla_{\mathbf{x}^*} f(\mathbf{x} + t\Delta\mathbf{x}) - \nabla_{\mathbf{x}^*} f(\mathbf{x})\| \cdot \|\Delta\mathbf{x}\| \mathrm{d}t. \tag{7}$$

By the gradient Lipschitz condition, we have

$$\|\nabla_{\mathbf{x}^*} f(\mathbf{x} + t\Delta\mathbf{x}) - \nabla_{\mathbf{x}^*} f(\mathbf{x})\| \leq L\|t\Delta\mathbf{x}\| = Lt\|\Delta\mathbf{x}\|,$$

substituting the above inequality into the integral (7), we get

$$\text{Right-hand side } \leq 4L\|\Delta\mathbf{x}\|^2 \int_0^1 t\mathrm{d}t = 4L\|\Delta\mathbf{x}\|^2 \cdot \frac{1}{2} = 2L\|\Delta\mathbf{x}\|^2.$$

Thus, for all $\mathbf{x}, \mathbf{y} \in \mathbb{Q}^n$, we have

$$\left| f(\mathbf{y}) - f(\mathbf{x}) - 4\Re\left\{ \nabla_{\mathbf{x}^*} f(\mathbf{x})^H (\mathbf{y} - \mathbf{x}) \right\} \right| \leq 2L\|\mathbf{y} - \mathbf{x}\|^2.$$

This completes the proof of the quadratic upper bound.

□

Theorem 1. *Let $f : \mathbb{Q}^n \to \mathbb{R}$ is a real differentiable and gradient Lipschitz continuous function with constant $L > 0$. If the step size $\alpha = \frac{1}{2L}$ and momentum parameter $\beta_t = \frac{t-1}{t+2}$, then*

$$f(\mathbf{x}_t^*) - f(\mathbf{x}_t) \leq C/(t+1)^2$$

is true for any $t \geq 0$, where $f(\mathbf{x}_t^*) = \lambda_{\max}$ and C is a constant related to the initial conditions.

Proof. From Lemma 1, for the update $\mathbf{z}_{t+1} = \mathbf{y}_t + \alpha \nabla_{\mathbf{y}^*} f(\mathbf{y}_t)$, we have

$$f(\mathbf{z}_{t+1}) \geq f(\mathbf{y}_t) + 4\Re\left\{ \nabla_{\mathbf{y}^*} f(\mathbf{y}_t)^H (\mathbf{z}_{t+1} - \mathbf{y}_t) \right\} - 2L\|\mathbf{z}_{t+1} - \mathbf{y}_t\|^2$$

Substituting $\mathbf{z}_{t+1} - \mathbf{y}_t = \alpha \nabla_{\mathbf{y}^*} f(\mathbf{y}_t)$, we derive

$$f(\mathbf{z}_{t+1}) \geq f(\mathbf{y}_t) + 4\alpha\|\nabla f(\mathbf{y}_t)\|^2 - 2L\alpha^2\|\nabla f(\mathbf{y}_t)\|^2.$$

Choosing $\alpha = \frac{1}{2L}$, this simplifies to

$$f(\mathbf{z}_{t+1}) \geq f(\mathbf{y}_t) + \frac{3}{2L}\|\nabla f(\mathbf{y}_t)\|^2. \tag{8}$$

After projecting $\mathbf{z}_{t+1}$ onto the unit sphere, $\mathbf{x}_{t+1} = \mathbf{z}_{t+1}/\|\mathbf{z}_{t+1}\|$. Applying Lemma 1 again to $\mathbf{x}_{t+1}$ and $\mathbf{z}_{t+1}$, we get

$$f(\mathbf{x}_{t+1}) \geq f(\mathbf{z}_{t+1}) + 4\Re\left\{ \nabla_{\mathbf{z}_{t+1}^*} f(\mathbf{z}_{t+1})^H (\mathbf{x}_{t+1} - \mathbf{z}_{t+1}) \right\} - 2L\|\mathbf{x}_{t+1} - \mathbf{z}_{t+1}\|^2.$$

By the optimality condition of the projection, we have

$$\Re\left\{ \nabla_{\mathbf{z}_{t+1}^*} f(\mathbf{z}_{t+1})^H (\mathbf{x}_{t+1} - \mathbf{z}_{t+1}) \right\} \geq 0$$

which implies

$$f(\mathbf{x}_{t+1}) \geq f(\mathbf{z}_{t+1}) - 2L\|\mathbf{x}_{t+1} - \mathbf{z}_{t+1}\|^2. \tag{9}$$

Next, we consider the bound of the projection error

$$\begin{aligned}
\|\mathbf{x}_{t+1} - \mathbf{z}_{t+1}\| &= \left\| \frac{\mathbf{z}_{t+1}}{\|\mathbf{z}_{t+1}\|} - \mathbf{z}_{t+1} \right\| \\
&= \left\| \frac{\mathbf{z}_{t+1}(1 - \|\mathbf{z}_{t+1}\|)}{\|\mathbf{z}_{t+1}\|} \right\| \\
&= \frac{|1 - \|\mathbf{z}_{t+1}\||}{\|\mathbf{z}_{t+1}\|}\|\mathbf{z}_{t+1}\| \\
&= |1 - \|\mathbf{z}_{t+1}\||.
\end{aligned}$$

Combining $\mathbf{z}_{t+1} = \mathbf{y}_t + \alpha \nabla_{\mathbf{y}^*} f(\mathbf{y}_t)$ and normalizing such that $\|\mathbf{y}_t\| = 1$, we obtain

$$\|\mathbf{z}_{t+1}\| \leq \|\mathbf{y}_t\| + \alpha\|\nabla_{\mathbf{y}^*} f(\mathbf{y}_t)\|,$$

Preprints.org (www.preprints.org) | NOT PEER-REVIEWED | Posted: 7 March 2025 doi:10.20944/preprints202503.0469.v1

which implies that

$$\|\mathbf{x}_{t+1} - \mathbf{z}_{t+1}\| \leq \alpha \|\nabla_{\mathbf{y}^*} f(\mathbf{y}_t)\|.$$

Thus, the projection error in (9) is a higher-order term, we get

$$f(\mathbf{x}_{t+1}) \geq f(\mathbf{z}_{t+1}) - 2L\alpha^2 \|\nabla f(\mathbf{y}_t)\|^2.$$

The formulas (8) and (9) give a lower bound

$$f(\mathbf{x}_{t+1}) \geq f(\mathbf{y}_t) + \frac{3}{2L}\|\nabla f(\mathbf{y}_t)\|^2 - \frac{1}{2L}\|\nabla f(\mathbf{y}_t)\|^2 = f(\mathbf{y}_t) + \frac{1}{L}\|\nabla f(\mathbf{y}_t)\|^2. \tag{10}$$

We then define the Lyapunov function as

$$\Phi_t = \theta_t^2 (f(\mathbf{x}^*) - f(\mathbf{x}_t)) + \frac{1}{2}\|\mathbf{v}_t - \mathbf{x}^*\|^2,$$

where the auxiliary sequence $\theta_t = \frac{t+1}{2}$ satisfies $\beta_t = \frac{\theta_t - 1}{\theta_{t+1}}$, and $\mathbf{v}_t$ follows the update rule

$$\mathbf{v}_{t+1} = \mathbf{v}_t + \theta_t \alpha \nabla f(\mathbf{y}_t).$$

Our goal is to prove that $\Phi_{t+1} \leq \Phi_t$. Compute $\Phi_{t+1} - \Phi_t$, we obtain

$$\Phi_{t+1} - \Phi_t = \theta_{t+1}^2 (f(\mathbf{x}^*) - f(\mathbf{x}_{t+1})) - \theta_t^2 (f(\mathbf{x}^*) - f(\mathbf{x}_t)) + \frac{1}{2}\Big(\|\mathbf{v}_{t+1} - \mathbf{x}^*\|^2 - \|\mathbf{v}_t - \mathbf{x}^*\|^2\Big). \tag{11}$$

For the term $\|\mathbf{v}_{t+1} - \mathbf{x}^*\|^2$, we have

$$\begin{aligned}\|\mathbf{v}_{t+1} - \mathbf{x}^*\|^2 &= \|\mathbf{v}_t - \mathbf{x}^* + \theta_t \alpha \nabla f(\mathbf{y}_t)\|^2 \\ &= \|\mathbf{v}_t - \mathbf{x}^*\|^2 + 2\theta_t \alpha \Re\Big\{\nabla f(\mathbf{y}_t)^H (\mathbf{v}_t - \mathbf{x}^*)\Big\} + \theta_t^2 \alpha^2 \|\nabla f(\mathbf{y}_t)\|^2.\end{aligned}$$

Thus we have

$$\begin{aligned}\Phi_{t+1} - \Phi_t = &\Big(\theta_{t+1}^2 - \theta_t^2\Big)(f(\mathbf{x}^*) - f(\mathbf{x}_{t+1})) + \theta_t^2 (f(\mathbf{x}_{t+1}) - f(\mathbf{x}_t)) \\ &\theta_t \alpha \Re\Big\{\nabla f(\mathbf{y}_t)^H (\mathbf{v}_t - \mathbf{x}^*)\Big\} + \frac{\theta_t^2 + \alpha^2}{2}\|\nabla f(\mathbf{y}_t)\|^2.\end{aligned}$$

Suppose that

$$f(\mathbf{x}^*) \leq f(\mathbf{y}_t) + \Re\Big\{\nabla f(\mathbf{y}_t)^H (\mathbf{x}^* - \mathbf{y}_t)\Big\}.$$

Substituting this into the Lyapunov difference and combining with (10), we arrive at

$$\begin{aligned}\Phi_{t+1} - \Phi_t \leq &\Big(\theta_{t+1}^2 - \theta_t^2 - \theta_t^2\Big)(f(\mathbf{x}^*) - f(\mathbf{y}_t)) \\ &+ \theta_t \alpha \Re\Big\{\nabla f(\mathbf{y}_t)^H (\mathbf{v}_t - \mathbf{x}^*)\Big\} + \frac{\theta_t^2 \alpha^2}{2}\|\nabla f(\mathbf{y}_t)\|^2.\end{aligned}$$

Using $\theta_{t+1} = \theta_t + \frac{1}{2}$ and $\alpha = \frac{1}{2L}$, we simplify

$$\theta_{t+1}^2 - \theta_t^2 = \theta_t + \frac{1}{4} \quad \text{and} \quad \frac{\theta_t^2}{L} \geq \theta_t^2 \alpha^2 L.$$

After algebraic manipulation, we show that

$$\Phi_{t+1} \leq \Phi_t - \frac{\theta_t^2}{4L}\|\nabla f(y_t)\|^2 \leq \Phi_t.$$

From $\Phi_t \leq \Phi_0$ and $\theta_t = \frac{t+1}{2}$, we have

$$\frac{(t+1)^2}{4}(f(\mathbf{x}^*) - f(\mathbf{x}_t)) \leq \Phi_0 = \frac{1}{4}(f(\mathbf{x}^*) - f(x_0)) + \frac{1}{2}\|\mathbf{v}_0 - \mathbf{x}^*\|^2$$

Letting $\mathbf{v}_0 = \mathbf{x}_0$, we conclude

$$f(\mathbf{x}^*) - f(\mathbf{x}_t) \leq \frac{C}{(t+1)^2},$$

where $C = f(\mathbf{x}^*) - f(\mathbf{x}_0) + 2\|\mathbf{v}_0 - \mathbf{x}^*\|^2$ is a constant related to the initial setup. Algorithm 1 achieves an $O(1/t^2)$ convergence rate. This completes the proof.

□

5. Numerical Experiments

In this section, we provide numerical examples to demonstrate the feasibility and effectiveness of the quaternion Nesterov's accelerated gradient projection algorithm for the eigenvalue problem of quaternion Hermitian matrices. In the specific implementation of Algorithm 1, we set the constant step size and momentum parameter to $\alpha = 0.05$ and $\beta = 0.9$, respectively.

All the experiments are performed under Windows 11 and MATLAB version 23.2.0.2365128 (R2023b) with an AMD Ryzen 7 5800H with Radeon Graphics CPU at 3.20 GHz and 16 GB of memory.

Example 1. *Given quaternion matrix* $\mathbf{A} = A_0 + A_1\mathbf{i} + A_2\mathbf{j} + A_3\mathbf{k}$ *with*

$$A_0 = \begin{pmatrix} 17.6331 & -1.6420 & -1.2730 \\ -1.6420 & 8.3929 & -1.7952 \\ -1.2730 & -1.7952 & 15.1089 \end{pmatrix}, \ A_1 = \begin{pmatrix} 0 & 1.2315 & 1.5751 \\ -1.2315 & 0 & 2.5700 \\ -1.5751 & -2.5700 & 0 \end{pmatrix}$$

and

$$A_2 = \begin{pmatrix} 0 & 0.6530 & 3.2730 \\ -0.6530 & 0 & 1.2301 \\ -3.2730 & -1.2301 & 0 \end{pmatrix}, \ A_3 = \begin{pmatrix} 0 & -4.3909 & -9.2817 \\ 4.3909 & 0 & 1.9585 \\ 9.2817 & -1.9585 & 0 \end{pmatrix}.$$

In this experiment, we employ the quaternion Nesterov's accelerated gradient (QNAG) method (Algorithm 1) to compute all eigenvalues of $\mathbf{A}$, which are $\lambda_1 = 27.0543, \lambda_2 = 12.4577, \lambda_3 = 1.6229$, with their corresponding eigenvectors being

$$\mathbf{x}_1 = \begin{pmatrix} 0.4485 + 0.4371\mathbf{i} - 0.1443\mathbf{j} - 0.3829\mathbf{k} \\ -0.0365 + 0.0228\mathbf{i} + 0.0147\mathbf{j} + 0.1706\mathbf{k} \\ 0.3077 + 0.1006\mathbf{i} + 0.2161\mathbf{j} + 0.5076\mathbf{k} \end{pmatrix},$$
$$\mathbf{x}_2 = \begin{pmatrix} 0.1208 - 0.0357\mathbf{i} + 0.3112\mathbf{j} - 0.2267\mathbf{k} \\ 0.0627 - 0.7303\mathbf{i} - 0.0928\mathbf{j} - 0.0756\mathbf{k} \\ -0.2492 + 0.4640\mathbf{i} + 0.0583\mathbf{j} - 0.0588\mathbf{k} \end{pmatrix},$$
$$\mathbf{x}_3 = \begin{pmatrix} -0.0011 - 0.1557\mathbf{i} + 0.3809\mathbf{j} + 0.3270\mathbf{k} \\ 0.5029 + 0.3354\mathbf{i} + 0.0994\mathbf{j} + 0.2046\mathbf{k} \\ 0.1149 + 0.4714\mathbf{i} + 0.1735\mathbf{j} + 0.2025\mathbf{k} \end{pmatrix}.$$

Additionally, we obtain the following three residuals:

$$\|\mathbf{A}\mathbf{x}_1 - \mathbf{x}_1\lambda_1\| \leq 6.7432e - 12,$$
$$\|\mathbf{A}\mathbf{x}_2 - \mathbf{x}_2\lambda_2\| \leq 5.1990e - 12,$$
$$\|\mathbf{A}\mathbf{x}_3 - \mathbf{x}_3\lambda_3\| \leq 4.3378e - 12.$$

doi:10.20944/preprints202503.0469.v1

It is evident that the residuals are controlled within an ideal range, demonstrating the feasibility and effectiveness of Algorithm 1 in computing the eigenvalues of quaternion Hermitian matrices.

Example 2. *In this experiment, we utilize MATLAB's built-in functions to randomly generate three quaternion Hermitian matrices of different sizes and compare Algorithm 1 with the QPGA algorithm [3] and the* `eig` *function in the Quaternion Toolbox for MATLAB (QTFM)[17].*

We first test the performance of Algorithm 1, QPGA, and the eig function in computing the principal eigenvalues of three different types and sizes of quaternion Hermitian matrices. The numerical experimental results are presented in Table 1, which includes three evaluation metrics: the number of iterations, residuals, and runtime. Superior results are highlighted in bold. It can be observed that Algorithm 1 outperforms the other algorithms in terms of the number of iterations, problem residuals, and runtime, demonstrating significant advantages when computing large-scale quaternion Hermitian matrices.

Table 1. Numerical comparison results of the QNAG algorithm, QPGA algorithm, and eig function for computing the principal eigenvalue and its corresponding eigenvector of three types of quaternion Hermitian matrices.

Matrix type	n	Ours			QPGA [3]			`eig` [17]	
		IT	Residual	CPU(s)	IT	Residual	CPU(s)	Residual	CPU(s)
General Hermitian	300	**453**	**4.5453e-12**	**0.6667**	1764	9.9457e-12	2.4857	9.6329e-12	3.4420
	500	**432**	**9.4172e-12**	**2.6465**	1368	9.8076e-12	8.2786	3.3852e-11	24.6503
	1000	**461**	**8.2506e-12**	**10.4795**	1592	9.8945e-12	36.1470	2.1115e-10	373.4246
Positive Semi-definite	300	**434**	**1.6844e-12**	**0.6420**	1141	9.8121e-12	1.7283	1.5353e-11	3.3504
	500	**454**	**9.2637e-12**	**2.6836**	2020	9.9177e-12	11.7135	1.9891e-11	24.6779
	1000	**485**	**8.2055e-12**	**10.6467**	1943	9.9733e-12	42.3868	1.7807e-10	362.1100
Positive Definite	300	**419**	**1.5814e-12**	**0.6079**	1615	9.8821e-12	2.2746	3.9411e-11	3.3670
	500	**457**	**9.9405e-12**	**2.7210**	1530	9.5587e-12	9.1385	2.3945e-11	24.3842
	1000	**503**	**7.8888e-12**	**11.2300**	2636	9.9721e-12	56.6146	7.2911e-11	379.0733

Subsequently, we plotted the variation curves of the objective function values for the first 50 iterations of Algorithm 1 and the QNGA algorithm, as shown in Figure 1. It is evident that our algorithm achieves a faster increase in the objective function, demonstrating higher efficiency in obtaining the maximum eigenvalue.

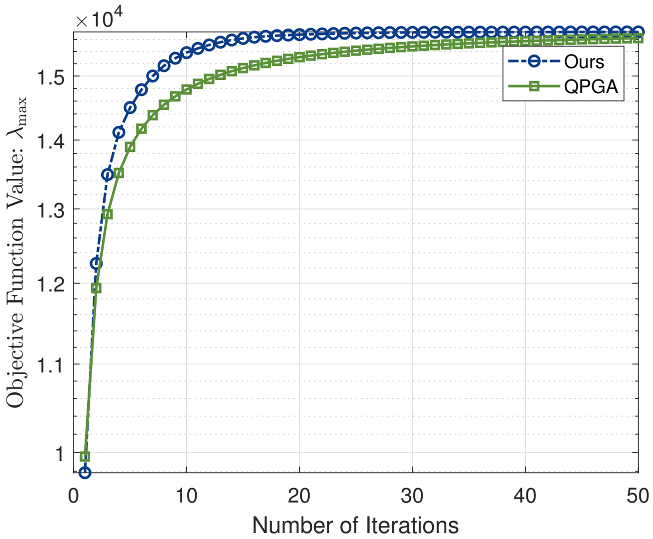

Figure 1. Variation curve of the objective function value with respect to the number of iterations for Algorithm 1 on a 1000-order quaternion Hermitian matrix.

Figure 2 illustrates the residual variation curves generated by Algorithm 1 and the QNGA algorithm with respect to the number of iterations. Across the tested matrix dimensions, our algorithm consistently achieves higher accuracy and efficiency.

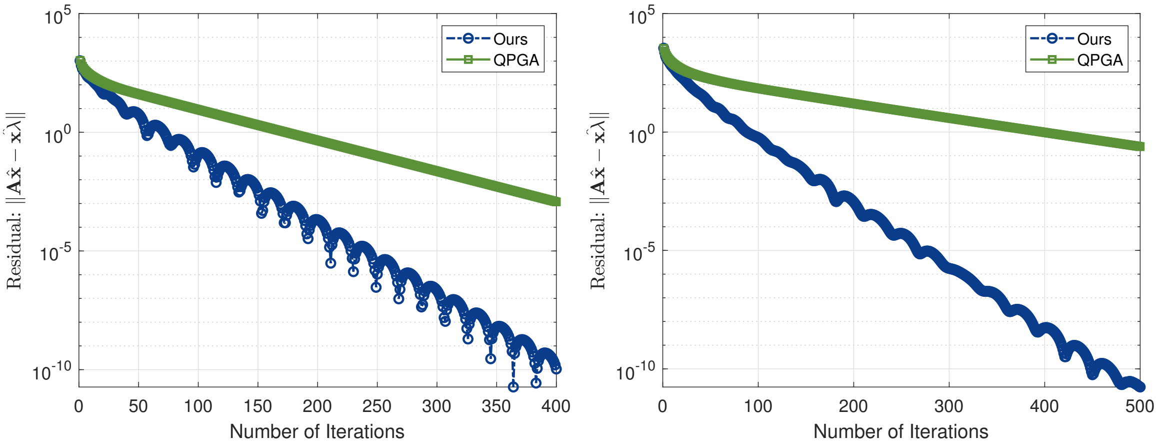

Figure 2. Comparison of fixed iteration steps and residual variation curves between Algorithm 1 and QPGA for solving Problem (4), with quaternion Hermitian matrix sizes of order 300 and 1000 on the left and right, respectively.

6. Conclusions

In this paper, leveraging the innovative generalized Hamilton-real (GHR) calculus, we have introduced a novel quaternion Nesterov's accelerated projected gradient algorithm designed to compute the dominant eigenvalue and corresponding eigenvector of quaternion Hermitian matrices. The incorporation of momentum terms and look-ahead updates has enabled the algorithm to attain an accelerated convergence rate. Theoretical analysis has confirmed the convergence of the quaternion Nesterov's accelerated projected gradient algorithm. Empirical results from numerical experiments indicate that the proposed method surpasses both the Quaternion Projected Gradient Ascent (QPGA) and conventional algebraic approaches in terms of both computational precision and efficiency in runtime.

Author Contributions: Shan-Qi Duan wrote the main manuscript text and performed the experiment. Qing-Wen Wang contributed to the conception of the study and helped to improve this manuscript with constructive suggestions. Xue-Feng Duan made a lot of useful suggestions. All authors reviewed the manuscript.

Funding: This work is supported by the National Natural Science Foundation of China under Grant 12371023.

Data Availability Statement: Data is contained within the article.

Conflicts of Interest: The authors declare no conflict of interest.

References

1. Bunse-Gerstner, A., Byers, R., Mehrmann, V. A quaternion QR algorithm. *Numer. Math.* **1989**, *55*, 83–95.
2. De Leo, S., Scolarici, G. Right eigenvalue equation in quaternionic quantum mechanics. *J. Phys. A Math. Gen.* **2000**, *33*, 2971.
3. Diao, Q., Liu, J., Zhang, N., Xu, D. An iterative algorithm for quaternion eigenvalue problems in signal processing. *IEEE Signal Process. Lett.* **2024**, *31*, 2505–2509.
4. Duan, S.Q., Wang, Q.W., Duan, X.F. On Rayleigh quotient iteration for the dual quaternion Hermitian eigenvalue problem. *Mathematics* **2024**, *12*.
5. Ell, T.A., Sangwine, S.J. Quaternion involutions and anti-involutions. *Comput. Math. Appl.* **2007**, *53*, 137–143.
6. Farid, F., Wang, Q.W., Zhang, F. On the eigenvalues of quaternion matrices. *Lin. Multilin. Alg.* **2011**, *59*, 451–473.
7. Ghadimi, S., Lan, G. Accelerated gradient methods for nonconvex nonlinear and stochastic programming. *Math. Program.* **2016**, *156*, 59–99.
8. Guo, Z., Jiang, T., Vasil'ev, V., Wang, G. Complex structure-preserving method for schrödinger equations in quaternionic quantum mechanics. *Numer. Algorithms* **2024**, *97*, 271–287.

doi:10.20944/preprints202503.0469.v1

9. Hamilton, W.R. Elements of Quaternions. Longmans, Green (1866)
10. Jia, Z., Wei, M., Ling, S. A new structure-preserving method for quaternion Hermitian eigenvalue problems. *J. Comput. Appl. Math.* **2013**, *239*, 12–24.
11. Jiang, T., Chen, L. An algebraic method for schrödinger equations in quaternionic quantum mechanics. *Comput. Phys. Commun.* **2008**, *178*, 795–799.
12. Li, H., Peng, Z., Pan, C., Zhao, D. Fast gradient method for low-rank matrix estimation. *J. Sci. Comput.* **2023**, *96*, 41.
13. Li, Y., Wei, M., Zhang, F., Zhao, J. On the power method for quaternion right eigenvalue problem. *J. Comput. Appl. Math.* **2019**, *345*, 59–69.
14. Macías-Virgós, E., Pereira-Sáez, M., Tarrío-Tobar, A.D. Rayleigh quotient and left eigenvalues of quaternionic matrices. *Linear Multilinear Algebra* **2023**, *71*, 2163–2179.
15. Nesterov, Y. A method for unconstrained convex minimization problem with the rate of convergence $O(1/k^2)$. *Dokl. Akad. Nauk. SSSR* **1983**, *269*, 543.
16. Rodman, L. Topics in quaternion linear algebra. Princeton University Press (2014)
17. Sangwine, S.J., Bihan, N.L. Quaternion toolbox for matlab, version 2 with support for octonions. Software library available at: http: //qtfm.sourceforge.net/ (2013)
18. Wang, Q.W., Wang, X.X. Arnoldi method for large quaternion right eigenvalue problem. *J. Sci. Comput.* **2020**, *82*, 58.
19. Ward, J.P. Quaternions and Cayley numbers: Algebra and applications, vol. 403. Springer Science & Business Media (2012)
20. Xu, D., Jahanchahi, C., Took, C.C., Mandic, D.P. Enabling quaternion derivatives: The generalized hr calculus. *R. Soc. Open Sci.* **2015**, *2*, 150255.
21. Xu, D., Mandic, D.P. The theory of quaternion matrix derivatives. *IEEE Trans. Signal Process.* **2015**, *63*, 1543–1556.
22. Xu, D., Xia, Y., Mandic, D.P. Optimization in quaternion dynamic systems: Gradient, Hessian, and learning algorithms. *IEEE Trans. Neural Netw. Learn. Syst.* **2015**, *27*, 249–261.
23. Yu, C.E., Liu, X., Zhang, Y. A new complex structure-preserving method for QSVD. *J. Sci. Comput.* **2024**, *99*, 37.
24. Zeng, R., Wu, J., Shao, Z., Chen, Y., Chen, B., Senhadji, L., Shu, H. Color image classification via quaternion principal component analysis network. *Neurocomputing* **2016**, *216*, 416–428.
25. Zhang, F. Quaternions and matrices of quaternions. *Linear Algebra Its Appl.* **1997**, *251*, 21–57.

Disclaimer/Publisher's Note: The statements, opinions and data contained in all publications are solely those of the individual author(s) and contributor(s) and not of MDPI and/or the editor(s). MDPI and/or the editor(s) disclaim responsibility for any injury to people or property resulting from any ideas, methods, instructions or products referred to in the content.