Mathematics

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

AIMS Mathematics, 9(10): 26371–26402.
DOI: 10.3934/math.20241284
Received: 08 July 2024
Revised: 02 August 2024
Accepted: 21 August 2024
Published: 11 September 2024

Research article

Systems of quaternionic linear matrix equations: solution, computation, algorithm, and applications

Abdur Rehman[1], Muhammad Zia Ur Rahman[2], Asim Ghaffar[2], Carlos Martin-Barreiro[3], Cecilia Castro[4,*], Víctor Leiva[5,*] and Xavier Cabezas[6,7]

1 Department of Basic Sciences and Humanities, University of Engineering and Technology, Lahore, Faisalabad Campus, Faisalabad, Pakistan

2 Department of Mechanical, Mechatronics and Manufacturing Engineering, University of Engineering and Technology, Lahore, Faisalabad Campus, Faisalabad, Pakistan

3 Facultad de Ingeniería, Universidad Espíritu Santo, Samborondón, Ecuador

4 Centre of Mathematics, Universidade do Minho, Braga, Portugal

5 Escuela de Ingeniería Industrial, Pontificia Universidad Católica de Valparaíso, Valparaíso, Chile

6 Facultad de Ciencias Naturales y Matemáticas, Escuela Superior Politécnica del Litoral ESPOL, Guayaquil, Ecuador

7 Centro de Estudios e Investigaciones Estadísticas, Escuela Superior Politécnica del Litoral ESPOL, Guayaquil, Ecuador

* **Correspondence:** Email: cecilia@math.uminho.pt, victorleivasanchez@gmail.com.

Abstract: In applied and computational mathematics, quaternions are fundamental in representing three-dimensional rotations. However, specific types of quaternionic linear matrix equations remain few explored. This study introduces new quaternionic linear matrix equations and their necessary and sufficient conditions for solvability. We employ a methodology involving lemmas and ranks of coefficient matrices to develop a novel algorithm. This algorithm is validated through numerical examples, showing its applications in advanced fields. In control theory, these equations are used for analyzing control systems, particularly for spacecraft attitude control in aerospace engineering and for control of arms in robotics. In quantum computing, quaternionic equations model quantum gates and transformations, which are important for algorithms and error correction, contributing to the development of fault-tolerant quantum computers. In signal processing, these equations enhance multidimensional signal filtering and noise reduction, with applications in color image processing and radar signal analysis. We extend our study to include cases of η-Hermitian and i-Hermitian solutions. Our work represents an advancement in applied mathematics, providing computational methods for solving quaternionic matrix equations and expanding their practical applications.

Keywords: control theory; Hermitian solutions; quaternionic equations; solvability conditions
Mathematics Subject Classification: 62H25, 62H30

1. Introduction

Abstract mathematical constructs can play instrumental roles in addressing practical problems. For example, matrices are ubiquitous in diverse fields such as computer graphics and quantum mechanics, serving as computational backbones for complex systems [1]. Similarly, quaternions have carved out a unique niche for themselves, especially in representing three-dimensional rotations—a role traditionally taken on by three-dimensional square matrices.

Quaternions can be traced back to the work of Sir William Rowan Hamilton, a 19th-century Irish mathematician [2]. Unlike traditional real or complex numbers, quaternions belong to a peculiar category of noncommutative yet associative division algebras. This peculiarity is evident in the multiplication of such matrices, which share the non-commutativity. Extending beyond pure mathematics, quaternions have proven invaluable in domains like color image processing as well as quantum mechanics and physics [3–9]. In engineering, quaternions serve as vital tools for formulating quaternionic linear matrix equations, providing fresh perspectives into subjects such as neural networks, perturbation and control theories, and sensitivity analysis [10, 11].

Quaternionic linear systems are crucial in addressing specific engineering challenges such as singular system control [12], system design [13], and linear descriptor systems [14]. These systems often require solutions to different forms of linear quaternion matrix equations [15], which are frequently the transformed forms of constant-coefficient quaternion differential equations [16]. Recent studies have explored various applications and solutions of quaternionic matrix equations, including zeroing neural networks for color restoration of images [17], formulas on the Drazin inverse for the sum of two matrices and block matrices [18], singular value decomposition and generalized inverse of a commutative quaternion matrix [19], as well as novel quaternion linear matrix equation solvers for acoustic source tracking [20]. Additionally, studies presented in [21, 22] have contributed to the understanding and solving of these equations by addressing specific types of solutions such as η-anti-Hermitian and η-Hermicity properties.

Other studies [23] stated the solvability and algorithm for Sylvester-type quaternion matrix equations, using generalized inverses and providing an algorithm validated numerically. Furthermore, systems of quaternionic linear matrix equations are used both in the study of Pythagorean-hodograph curves and for theories of quaternionic functions such as slice regularity [24, 25]. However, a gap remains in the literature regarding the full comprehension and efficient solving methods for these complex systems related to quaternionic linear equations. To bridge this gap, one can extend the quaternionic matrix theory by introducing new conditions for the solvability of quaternionic linear systems and by developing a novel algorithmic approach to solving these systems.

The objectives of this study are: (i) to establish a more comprehensive understanding of quaternionic linear systems; (ii) to develop an efficient solving methodology for these systems providing a computational analysis and algorithmic solutions; and (iii) to show their practical applications in various advanced fields.

To achieve these objectives, we employ a multidisciplinary approach that combines mathematical rigor with computational techniques. This approach involves formulating and proving rigorous mathematical lemmas, developing a novel algorithm, and its subsequent validation through numerical computation, testing and illustrations. Our approach incorporates techniques from control theory and quantum computing, thereby broadening the scope and applicability of quaternionic equations in engineering and applied sciences. Therefore, the novelty of the present study lies in the introduction of new conditions for system solvability, its computation, and the development of a novel algorithmic approach, which addresses existing gaps and enhances the understanding and practical implementation of quaternionic linear systems.

The rest of this article is structured as follows. Section 2 provides the mathematical foundations and reviews the related literature on quaternion matrices. In Section 3, we present our core findings, substantiated by a series of instrumental lemmas. In Section 4, our approach to solving the equations is outlined, supplemented with an algorithm, flowchart, and practical numerical example. Section 5 focuses on the special case of the Hermitian solution, detailing its conditions and properties. In Section 6, we summarize our contributions and discuss their broader implications.

2. Preliminaries

In this section, we establish the mathematical foundations and prior works that are essential for our research on quaternionic matrices. We begin by introducing the basic notations and definitions related to quaternion algebra, including the unique properties of quaternions and their noncommutative multiplication. This is followed by an exploration of addition and multiplication rules for quaternionic matrices, and then we present the concepts of Moore-Penrose inverse, η-conjugate transpose, and i-conjugate transpose for quaternionic matrices.

2.1. Notation and definitions

Throughout this work, we denote the quaternion algebra by $\mathbb{H}$, which is defined as

$$\mathbb{H} \equiv \{a_0 + a_1\mathrm{i} + a_2\mathrm{j} + a_3\mathrm{k} \mid a_0, a_1, a_2, a_3 \in \mathbb{R}\}, \tag{2.1}$$

where $\mathrm{i}, \mathrm{j}, \mathrm{k}$ are three distinct imaginary units satisfying the multiplication rules stated as

$$\mathrm{i}^2 = \mathrm{j}^2 = \mathrm{k}^2 = -1, \mathrm{ij} = \mathrm{k}, \mathrm{ji} = -\mathrm{k}, \mathrm{jk} = \mathrm{i}, \mathrm{kj} = -\mathrm{i}, \mathrm{ki} = \mathrm{j}, \mathrm{ik} = -\mathrm{j}. \tag{2.2}$$

Quaternions extend traditional complex numbers by incorporating these three imaginary units, which reflect their noncommutative nature, in contrast to the commutative multiplication of traditional complex and real numbers. This quaternionic algebraic structure enables the representation of three-dimensional rotations and constitutes a four-dimensional construct.

2.2. Addition and multiplication of quaternionic matrices

Addition and multiplication for quaternionic matrices follow analogous rules to those for real or complex matrices, with the further consideration of the noncommutative nature of quaternion multiplication. For quaternionic matrices $A = (a_{ij})$ and $B = (b_{ij})$, both of size $m \times n$, their sum $C = (c_{ij}) = A + B$ is defined element-wise by $c_{ij} = a_{ij} + b_{ij}$, for all $i \in \{1, \ldots, m\}$ and $j \in \{1, \ldots, n\}$.

Multiplication, however, is affected by the noncommutative nature of quaternion multiplication. For multiplication, if $A = (a_{ij})$ is an $m \times p$ matrix and $B = (b_{ij})$ is a $p \times n$ matrix, their product $C = (c_{ij}) = AB$ is defined by $c_{ij} = \sum_{k=1}^{p} a_{ik} b_{kj}$, for all $i \in \{1, \ldots, m\}$ and $j \in \{1, \ldots, n\}$, where the multiplication of the individual quaternion elements adheres to the rules of quaternion multiplication stated in (2.2). An example of quaternion multiplication is given as follows. Consider two quaternions $q_1 = a_1 + b_1\mathrm{i} + c_1\mathrm{j} + d_1\mathrm{k}$ and $q_2 = a_2 + b_2\mathrm{i} + c_2\mathrm{j} + d_2\mathrm{k}$. Their product is given by $q_1 q_2 = (a_1 + b_1\mathrm{i} + c_1\mathrm{j} + d_1\mathrm{k})(a_2 + b_2\mathrm{i} + c_2\mathrm{j} + d_2\mathrm{k})$. Expanding this product and applying the rules of quaternion multiplication, we get $q_1 q_2 = a_1 a_2 - b_1 b_2 - c_1 c_2 - d_1 d_2 + (a_1 b_2 + b_1 a_2 + c_1 d_2 - d_1 c_2)\mathrm{i} + (a_1 c_2 - b_1 d_2 + c_1 a_2 + d_1 b_2)\mathrm{j} + (a_1 d_2 + b_1 c_2 - c_1 b_2 + d_1 a_2)\mathrm{k}$. This noncommutative multiplication must be applied element-wise when multiplying quaternionic matrices.

2.3. Key operations on quaternionic matrices

Having established the basic operations of addition and multiplication for quaternionic matrices, we now turn our attention to more advanced concepts essential for solving linear systems involving quaternionic matrices. These concepts include the Moore-Penrose inverse, η-conjugate transpose, and i-conjugate transpose.

In traditional linear algebra, the Moore-Penrose inverse is a powerful tool for finding generalized inverses of singular or non-square matrices [26], allowing us to solve linear equations that do not have unique solutions. For a quaternionic matrix A, the Moore-Penrose inverse, denoted by $A^\dagger$, is defined as the unique matrix that satisfies the conditions represented as $AA^\dagger A = A$, $A^\dagger A A^\dagger = A^\dagger$, $(AA^\dagger)^* = AA^\dagger$, and $(A^\dagger A)^* = A^\dagger A$, where $(\cdot)^*$ denotes the conjugate transpose. This inverse $A^\dagger$ allows us to handle cases where the matrix is not invertible in the traditional sense, providing a means to work with over-determined or under-determined systems.

The η-conjugate transpose of a quaternionic matrix A, denoted by A_η^*, extends the conjugate transpose of the complex matrix algebra. This involves to transpose A and replace each element with its η-conjugate. For example, a quaternionic number $q = a + b\mathrm{i} + c\mathrm{j} + d\mathrm{k}$ has its η-conjugate given by $q_\eta^* = a - \eta b\mathrm{i} - \eta c\mathrm{j} - \eta d\mathrm{k}$, where η is a real number. The η-conjugate transpose is helpful in the study of Hermitian quaternionic matrices, which find applications in fields such as quantum mechanics and signal processing.

The i-conjugate transpose of a quaternionic matrix A, denoted by A_i^*, is obtained by taking the transpose of A and replacing each element with its i-conjugate. The i-conjugate of a quaternionic number $q = a + b\mathrm{i} + c\mathrm{j} + d\mathrm{k}$ is given by $q_\mathrm{i}^* = a - b\mathrm{i} + c\mathrm{j} + d\mathrm{k}$, where only the coefficient of the i-part is conjugated (such as in traditional complex numbers), while the coefficients of the j-part and k-part remain unchanged. This form of conjugation is used in specific quaternionic matrix operations, providing additional flexibility in mathematical manipulations and problem-solving.

The set of all $m \times n$ matrices over $\mathbb{H}$ is denoted by $\mathbb{H}^{m\times n}$. To understand the structure and properties of quaternionic matrices, we introduce the concept of projection operators. For a quaternionic matrix A, they are given by $L_A = I - A^\dagger A$ and $R_A = I - AA^\dagger$, where I is the identity matrix. These projectors have properties formulated as $L_A = L_A^* = L_A^2 = L_A^\dagger$ and $R_A = R_A^2 = R_A^* = R_A^\dagger$, where $(\cdot)^*$ denotes the conjugate transpose, as mentioned; $(\cdot)^\dagger$ is the Moore-Penrose inverse; whereas L_A and R_A are the left and right projection operators applied to the matrix A, reflecting operations that act on A from the left and right sides, respectively. These projectors play a crucial role in decomposing and analyzing quaternionic matrices, similar to their role in traditional linear algebra.

To illustrate the concepts of the Moore-Penrose inverse, η-conjugate transpose, and i-conjugate transpose of a quaternionic matrix, consider a 2×2 quaternionic matrix A and its usual, η-conjugate, and i-conjugate transposes given by

$$A = \begin{pmatrix} 1+\mathrm{i} & 2+\mathrm{j} \\ 2+\mathrm{j} & 4+\mathrm{k} \end{pmatrix},\ A^\top = \begin{pmatrix} 1+\mathrm{i} & 2+\mathrm{j} \\ 2+\mathrm{j} & 4+\mathrm{k} \end{pmatrix},\ A_\eta^* = \begin{pmatrix} 1-\eta\mathrm{i} & 2-\eta\mathrm{j} \\ 2-\eta\mathrm{j} & 4-\eta\mathrm{k} \end{pmatrix},\ A_\mathrm{i}^* = \begin{pmatrix} 1-\mathrm{i} & 2+\mathrm{j} \\ 2+\mathrm{j} & 4+\mathrm{k} \end{pmatrix}.$$

Now, we compute $A^\top A$, considering the noncommutative nature of quaternionic multiplication, obtaining

$$A^\top A = \begin{pmatrix} (1+\mathrm{i})(1+\mathrm{i}) + (2+\mathrm{j})(2+\mathrm{j}) & (1+\mathrm{i})(2+\mathrm{j}) + (2+\mathrm{j})(4+\mathrm{k}) \\ (2+\mathrm{j})(1+\mathrm{i}) + (4+\mathrm{k})(2+\mathrm{j}) & (2+\mathrm{j})(2+\mathrm{j}) + (4+\mathrm{k})(4+\mathrm{k}) \end{pmatrix}.$$

The individual terms are calculated as $(1+\mathrm{i})(1+\mathrm{i}) = -1+2\mathrm{i}$, $(2+\mathrm{j})(2+\mathrm{j}) = -1+4\mathrm{j}$, $(1+\mathrm{i})(2+\mathrm{j}) = 2+2\mathrm{i}+\mathrm{j}+\mathrm{k}$, $(2+\mathrm{j})(4+\mathrm{k}) = 8+2\mathrm{k}+4\mathrm{j}-\mathrm{i}$, and so on for the remaining terms. After calculating all the products, we reach

$$A^\top A = \begin{pmatrix} 3+2\mathrm{i}+4\mathrm{j} & 10+3\mathrm{i}+5\mathrm{j}+3\mathrm{k} \\ 10+\mathrm{i}+5\mathrm{j}+\mathrm{k} & 18+4\mathrm{j}+8\mathrm{k} \end{pmatrix}.$$

Next, we compute the inverse of $A^\top A$ given by

$$(A^\top A)^{-1} = \begin{pmatrix} -0.221-0.133\mathrm{i}+0.201\mathrm{j}-0.020\mathrm{k} & 0.109+0.124\mathrm{i}-0.089\mathrm{j}+0.053\mathrm{k} \\ 0.120+0.125\mathrm{i}-0.059\mathrm{j}+0.019\mathrm{k} & -0.039-0.066\mathrm{i}+0.002\mathrm{j}-0.043\mathrm{k} \end{pmatrix}.$$

Therefore, we calculate the corresponding Moore-Penrose inverse as

$$A^\dagger = (A^\top A)^{-1}A^\top = \begin{pmatrix} 0.219-0.159\mathrm{i}+0.113\mathrm{j}+0.009\mathrm{k} & -0.260+0.163\mathrm{i}-0.298\mathrm{j}+0.148\mathrm{k} \\ -0.084+0.156\mathrm{i}-0.074\mathrm{j}-0.074\mathrm{k} & 0.187-0.032\mathrm{i}+0.077\mathrm{j}-0.047\mathrm{k} \end{pmatrix}.$$

This example illustrates the unique aspects of working with quaternionic matrices, including the computation of the Moore-Penrose inverse and conjugate transposes. The `Python` code to reproduce this example is provided in Appendix A.

2.4. *Background and related work*

Quaternions, first introduced by Sir W.R. Hamilton [2], are notable for their associative nature and noncommutative multiplication, properties that have garnered high interest. Recently, quaternion matrices have found applications in various fields, including quantum physics and mechanics, as well as color image processing [3–9]. Engineering disciplines, such as singular system control [12], system design [13], and linear descriptor systems [14], frequently encounter problems requiring solutions to various forms of linear quaternion matrix equations. For example, interactions with constant coefficient quaternion differential equations [16] can be transformed into linear quaternion matrix equations, underscoring the importance of investigating their solutions. Additional applications can be found in feedback systems, neural networks, perturbation theory, and sensitivity analysis [27–30]. Several researchers have addressed different types of linear quaternion matrix equations. Some notable examples based on quaternionic matrices include:

- The iterative solution of $A_1X + XB_1 = C_1$, examined in [31] and further explored in [32], where A_1, B_1, and C_1 are known quaternionic matrices, and X is the quaternionic matrix to be determined and similarly for the matrices mentioned below.

- The solvability condition of $A_1X + YA_2 = B$, analyzed in [33], with the general solution studied in [34], and a constrained solution presented in [35].
- In [36], the necessary and sufficient conditions for the system given by

$$A_1X_1 + Z_1B_1 = C_1, A_2X_2 + Z_1B_2 = C_2, \tag{2.3}$$

were detailed, along with its general solution. Further analysis of the solvability conditions for this system was conducted in [37], and its condition number was studied in [38]. A constrained solution for the structure stated in (2.3) was derived in [39].
- The general solution of the system

$$A_1X_1 + Y_1B_1 + C_1Z_1D_1 = E_1, A_2X_2 + Y_2B_2 + C_2Z_1D_2 = E_2, \tag{2.4}$$

was provided under solvability conditions in [40].
- The general solution for the system described as

$$\begin{aligned} &A_3X_1 = C_3, Y_1B_3 = C_5, F_1Z_1 = G_1, Z_1F_2 = G_2, A_4X_2 = C_4,\\ &Y_2B_4 = C_6, A_1X_1 + Y_1B_1 + C_1Z_1D_1 = E_1, A_2X_2 + Y_2B_2 + C_2Z_1D_2 = E_2, \end{aligned} \tag{2.5}$$

was presented in [10] under the assumption of system consistency.
- Similar models to the systems defined in (2.4) and (2.5) have been extensively studied in [1, 10, 22, 35, 40–43].

Further analysis was conducted in [44] when $X_2 = X_1$, examining the solvability conditions for the expression formulated in (2.3). The system defined in (2.3) was also explored in [45]. In [36], necessary and sufficient conditions were provided for the solvability of the system expressed as $A_1X_1 + Z_1B_1 = C_1$ and $A_2Z_1 + X_2B_2 = C_2$, with its general solution presented in [46]. General solutions for systems of generalized Sylvester matrix equations with four matrices were discussed in [47]. More recently, the general solution of the system given by $A_1X_1 + Z_1B_1 = C_1$, $A_2X_2 + Z_1B_2 = C_2$, $A_3X_2 + Z_2B_3 = C_3$, and $A_4X_3 + Z_2B_4 = C_4$, using rank equalities and generalized inverses, was derived in [48] for consistent systems. The general solution for certain systems of matrix equations can be found in [1, 22, 35, 41, 42]. Recently, the general solution of the system stated as

$$A_1X_1 + Y_1B_1 + C_1ZD_1 = E_1, A_2X_2 + Y_2B_2 + C_2ZD_2 = E_2, A_3X_3 + Y_3B_3 + C_3ZD_3 = E_3, \tag{2.6}$$

was delineated in [43] under consistent conditions.

From our bibliographical review, there is sparse research on generalizations of the systems presented in (2.3) to (2.6). Motivated by the relevant applications of linear quaternion matrix equations, in the present article, we explore the necessary and sufficient conditions of the system defined as

$$\begin{aligned} &A_1X_1 = F_1, A_2X_2 = F_2, A_3X_3 = F_3, A_4Z_1 = F_7, Y_1B_1 = F_4, Y_2B_2 = F_5, Y_3B_3 = F_6, ZB_4 = F_8,\\ &A_5X_1 + Y_1B_5 + C_1ZD_1 = E_1, A_6X_2 + Y_2B_6 + C_2ZD_2 = E_2, A_7X_3 + Y_3B_7 + C_3ZD_3 = E_3, \end{aligned} \tag{2.7}$$

and its general solution when solvable, where $\{A_i, B_i\}_{i=1}^{7}$, $\{C_j, D_j, E_j\}_{j=1}^{3}$, and $\{F_k\}_{k=1}^{8}$ are known quaternionic matrices, whereas $\{X_l, Y_l\}_{l=1}^{3}$ and Z are the quaternionic matrices to be determined. Moreover, we show into the η-Hermitian solution of the system given by

$$\begin{aligned} &A_1X_1 = F_1, A_2X_2 = F_2, A_3X_3 = F_3, Z = Z_\eta^*, A_4Z_1 = F_7, ZB_4 = F_8,\\ &A_5X_1 + (A_5X_1)_\eta^* + C_1ZC_{1_\eta}^* = E_1, A_6X_2 + (A_6X_2)_\eta^* + C_2ZC_{2_\eta}^* = E_2, A_7X_3 + (A_7X_3)_\eta^* + C_3ZC_{3_\eta}^* = E_3, \end{aligned} \tag{2.8}$$

and its general solution when consistent, where $\{A_i\}_{i=1}^{7}$, B_4, $\{C_j, E_j\}_{j=1}^{3}$, and $\{F_k\}_{k=1}^{8}$ are known quaternionic matrices, whereas $\{X_l\}_{l=1}^{3}$, Z_1, and Z are the quaternionic matrices to be determined, with Z^* denoting the conjugate transpose of Z and Z_η the η-conjugate transpose of Z, as mentioned.

Our primary objective is to establish solvability conditions for the system presented in (2.7), derive its general solution using generalized inverses and rank equalities of the provided coefficient matrices, and extend these results to the η-Hermitian solution of the system stated in (2.8) when solutions exist.

2.5. Conceptual applications of quaternionic linear matrix equations

The quaternionic linear matrix equations discussed in this article, such as those described in (2.7) and (2.8), have important practical applications in various advanced fields. In control theory, these equations are crucial for the design and analysis of advanced control systems, particularly in scenarios involving high-dimensional state variables and complex dynamic systems, such as aerospace engineering and robotics.

Quaternionic equations, for instance, are essential in the attitude control of spacecraft and unmanned aerial vehicles due to their ability to efficiently handle three-dimensional rotations and avoid singularities associated with Euler angles. These equations enable the development of robust control laws and stability analyses, which are vital for the reliable operation of the mentioned complex systems [49].

In quantum computing, the noncommutative nature of quaternions provides a powerful framework for representing and manipulating quantum states and operations. Quaternionic linear matrix equations are instrumental in modeling quantum gates and transformations, which are foundational for quantum algorithms and error correction methods. For instance, quaternionic representations can describe the evolution of quantum states in certain quantum walks, which play a critical role in developing quantum algorithms [50]. This framework is particularly relevant for designing and implementing fault-tolerant quantum computing systems, where precise control and correction of quantum states are essential.

In the field of signal processing, quaternionic equations are employed for advanced filtering and noise reduction in multidimensional signals. Specifically, in color image processing, quaternionic models treat color channels as a single entity rather than separate components, preserving the inherent correlations between channels and improving filtering performance [51]. Thus, quaternionic equations improve the accuracy and effectiveness of signal processing algorithms, leading to superior results in applications such as radar signal analysis and color image enhancement.

Additionally, quaternionic matrix equations are integral to the study of Pythagorean-hodograph curves and theories of quaternionic functions, such as slice regularity [24,25]. These studies underscore the broader mathematical and computational importance of quaternionic equations, extending their utility beyond traditional engineering fields.

By addressing specific solvability conditions and developing efficient algorithms for solving the quaternionic linear matrix equations detailed in (2.7) and (2.8), our work aims to provide robust tools for tackling real-world engineering and computational challenges, such as those found in control systems, quantum computing, and signal processing.

3. Principal findings

In this section, we present the core results of our research on linear quaternion matrix equations. Building on the foundational concepts and prior works introduced in Section 2, we provide a series of key lemmas that explore solvability conditions, general solutions, and η-Hermitian solutions of quaternionic matrix equations.

3.1. Key lemmas

In our exploration of linear quaternion matrix equations, we identify several key lemmas that are critical to understanding the structure and solutions of these equations. Each lemma provides a fundamental insight into the relationships between quaternionic matrices, which are crucial for establishing solvability and attaining general solutions.

The first lemma, based on [52], establishes fundamental relationships between three quaternionic matrices, denoted by K, P, and Q. Here, rank$[A]$ denotes the rank of a matrix A, which is defined as the number of linearly independent rows or columns in A. These rank equalities are used to establish relationships between different matrices, ensuring the consistency of solutions to the presented matrix equations.

Lemma 3.1. *Let $A \in \mathbb{H}^{m\times n}$, $B \in \mathbb{H}^{m\times t}$, and $C \in \mathbb{H}^{l\times n}$ be quaternionic matrices. Then, we have*

$$\operatorname{rank}\begin{bmatrix} A \\ C \end{bmatrix} - \operatorname{rank}[A] = \operatorname{rank}[CL_A], \operatorname{rank}\begin{bmatrix} A & B \end{bmatrix} - \operatorname{rank}[B] = \operatorname{rank}[R_BA],$$

$$\operatorname{rank}\begin{bmatrix} A & B \\ C & 0 \end{bmatrix} - \operatorname{rank}[R_BAL_C] = \operatorname{rank}[B] + \operatorname{rank}[C].$$

Next, we examine the solvability conditions of matrix equations involving quaternionic matrices A_1 and A_2. The following lemma, adapted from [53], elucidates these conditions.

Lemma 3.2. *Let $A_1 \in \mathbb{H}^{m\times n}$ and $A_2 \in \mathbb{H}^{m\times p}$ be quaternionic matrices. Then, the equation $A_1X = A_2$ is solvable if and only if $A_2 = A_1A_1^{\dagger}A_2$. The general solution of this equation is given by $X = A_1^{\dagger}A_2 + L_{A_1}U_1$, where U_1 is any quaternionic matrix in $\mathbb{H}^{n\times p}$.*

Building on the previous lemma, we extend our understanding of consistency conditions, this time considering quaternionic matrices B_{11} and D_{11}. The following lemma, adapted from [53], outlines the necessary and sufficient conditions for the consistency of the equation $YB_{11} = D_{11}$.

Lemma 3.3. *Let $B_{11} \in \mathbb{H}^{m\times n}$ and $D_{11} \in \mathbb{H}^{p\times n}$ be quaternionic matrices. Then, the equation $YB_{11} = D_{11}$ is consistent if and only if $D_{11} = D_{11}B_{11}^{\dagger}B_{11}$. Under this condition, the general solution of the equation is $Y = D_{11}B_{11}^{\dagger} + W_1R_{B_{11}}$, where W_1 is any quaternionic matrix in $\mathbb{H}^{p\times m}$.*

The following lemma, based on [54], provides consistency conditions for a system involving various quaternionic matrices.

Lemma 3.4. *Let $A_1 \in \mathbb{H}^{m\times n}$, $B_1 \in \mathbb{H}^{p\times q}$, $C_1 \in \mathbb{H}^{m\times p}$, and $C_2 \in \mathbb{H}^{n\times q}$ be quaternionic matrices. Then, the system given by $A_1X_1 = C_1$ and $X_1B_1 = C_2$ is consistent if and only if $R_{A_1}C_1 = 0$, $C_2L_{B_1} = 0$, and $A_1C_2 = C_1B_1$. Under these conditions, the general solution of the system is expressed as $X_1 = A_1^{\dagger}C_1 + L_{A_1}C_2B_1^{\dagger} + L_{A_1}U_1R_{B_1}$, where U_1 is any quaternionic matrix in $\mathbb{H}^{n\times p}$.*

Our exploration continues with a more intricate lemma derived from [55], which connects several quaternionic matrices and establishes a set of equivalent statements that are critical for the analysis of the matrix equation system.

Lemma 3.5. *Let $A_1 \in \mathbb{H}^{m\times n}$, $B_1 \in \mathbb{H}^{p\times q}$, $C_3 \in \mathbb{H}^{m\times r}$, $D_3 \in \mathbb{H}^{u\times q}$, $C_4 \in \mathbb{H}^{m\times r}$, $D_4 \in \mathbb{H}^{u\times q}$, and $E_1 \in \mathbb{H}^{m\times q}$ be quaternionic matrices. Define auxiliary quaternionic matrices as $A = R_{A_1}C_3$, $B = D_3L_{B_1}$, $C = R_{A_1}C_4$, $D = D_4L_{B_1}$, $E = R_{A_1}E_1L_{B_1}$, $F = R_AC$, $G = DL_B$, and $H = CL_F$. Then, the following statements are equivalent:*

(i) The quaternionic matrix equation

$$A_1U + VB_1 + C_3WD_3 + C_4ZD_4 = E_1 \tag{3.1}$$

has a solution.

(ii) It holds that $R_FR_AE = 0$, $EL_BL_G = 0$, $R_AEL_D = 0$, and $R_CEL_B = 0$.

(iii) It holds that the rank equalities are satisfied as

$$\operatorname{rank}\begin{bmatrix} E_1 & C_4 & C_3 & A_1 \\ B_1 & 0 & 0 & 0 \end{bmatrix} = \operatorname{rank}[B_1] + \operatorname{rank}\left[C_4\, C_3\, A_1\right], \operatorname{rank}\begin{bmatrix} E_1 & A_1 \\ D_3 & 0 \\ D_4 & 0 \\ B_1 & 0 \end{bmatrix} = \operatorname{rank}\begin{bmatrix} D_3 \\ D_4 \\ B_1 \end{bmatrix} + \operatorname{rank}[A_1],$$

$$\operatorname{rank}\begin{bmatrix} E_1 & C_3 & A_1 \\ D_4 & 0 & 0 \\ B_1 & 0 & 0 \end{bmatrix} = \operatorname{rank}\left[A_1\, C_3\right] + \operatorname{rank}\begin{bmatrix} D_4 \\ B_1 \end{bmatrix}, \operatorname{rank}\begin{bmatrix} E_1 & C_4 & A_1 \\ D_3 & 0 & 0 \\ B_1 & 0 & 0 \end{bmatrix} = \operatorname{rank}\left[A_1\, C_4\right] + \operatorname{rank}\begin{bmatrix} D_3 \\ B_1 \end{bmatrix}.$$

Under these statements (i)–(iii), the general solution to the equation given in (3.1) *is expressed as:*

$$\begin{aligned}
U &= A_1^{\dagger}(E_1 - C_3WD_3 - C_4ZD_4) - A_1^{\dagger}S_7B_1 + L_{A_1}S_6,\\
V &= R_{A_1}(E_1 - C_3WD_3 - C_4ZD_4)B_1^{\dagger} + A_1A_1^{\dagger}S_7 + S_8R_{B_1},\\
W &= A^{\dagger}EB^{\dagger} - A^{\dagger}CF^{\dagger}EB^{\dagger} - A^{\dagger}HC^{\dagger}EG^{\dagger}DB^{\dagger} - A^{\dagger}HS_2R_GDB^{\dagger} + L_AS_4 + S_5R_B,\\
Z &= F^{\dagger}ED^{\dagger} + H^{\dagger}HC^{\dagger}EG^{\dagger} + L_FL_HS_1 + L_FS_2R_G + S_3R_D,
\end{aligned}$$

where $S_1, \dots, S_8$ are any matrices of appropriate dimensions over $\mathbb{H}$.

As previously established, the lemmas we have explored provide the necessary groundwork for our main findings. In particular, Lemma 3.5 sets the stage for our central theorem. With this foundation in place, we now present the core theorem of this investigation, which synthesizes the results discussed and offers a comprehensive characterization of the underlying quaternionic matrix relationships.

3.2. Main theorem and comprehensive analysis

Next, we present the principal theorem of our investigation, which consolidates the insights garnered from the previous lemmas and provides a comprehensive analysis of the consistency conditions and solutions for a broad class of matrix equations. The theorem relies on a diverse set of matrices and the mathematical framework established in the earlier sections.

Theorem 3.1. *Let $\{A_i, B_i\}_{i=1}^{7}$, $\{C_j, D_j\}_{j=1}^{3}$, and $\{E_k\}_{k=1}^{6}$ be quaternionic matrices of conformable sizes over $\mathbb{H}$ and consider the following relationships:*

$$\begin{aligned}
A_8 &= A_5L_{A_1}, B_8 = R_{B_1}B_5, C_4 = C_1L_{A_4}, D_4 = R_{B_4}D_1,\\
F_1 &= A_1^{\dagger}E_1, \widehat{X}_{01} = A_1^{\dagger}F_1, F_4 = E_4B_1^{\dagger}, \widehat{Y}_{01} = F_4B_1^{\dagger},\\
A_9 &= A_6L_{A_2}, B_9 = R_{B_2}B_6, C_5 = C_2L_{A_5}, D_5 = R_{B_5}D_2,\\
F_2 &= A_2^{\dagger}E_2, \widehat{X}_{02} = A_2^{\dagger}F_2, F_5 = E_5B_2^{\dagger}, \widehat{Y}_{02} = F_5B_2^{\dagger},\\
A_{10} &= A_7L_{A_3}, B_{10} = R_{B_3}B_7, C_6 = C_3L_{A_6}, D_6 = R_{B_6}D_3, F_3 = A_3^{\dagger}E_3,\\
\widehat{X}_{03} &= A_3^{\dagger}F_3, \widehat{Y}_{03} = F_6B_3^{\dagger}, F_6 = E_6B_3^{\dagger}, A_{11} = R_{A_8}C_4, B_{11} = D_4L_{B_8},\\
E_7 &= R_{A_8}E_4L_{B_8}, F_7 = A_4^{\dagger}E_7, A_{12} = R_{A_9}C_5, B_{12} = D_5L_{B_9}, E_8 = R_{A_9}E_5L_{B_9}, F_8 = E_8B_4^{\dagger},\\
\widehat{Z}_{01} &= A_4^{\dagger}F_7 + L_{A_4}F_8B_4^{\dagger}, A_{13} = R_{A_{10}}C_6, B_{13} = D_6L_{B_{10}}, E_9 = R_{A_{10}}E_6L_{B_{10}},\\
A_{14} &= A_{12}L_{A_{11}}, B_{14} = R_{B_{11}}B_{12}, E_{10} = E_8 - A_{12}A_{11}^{\dagger}E_7B_{11}^{\dagger}B_{12},\\
A_{15} &= \begin{bmatrix} L_{A_{11}}L_{A_{14}} & -L_{A_{13}} \end{bmatrix}, C_8 = L_{A_{11}}, C_9 = L_{A_{12}}, D_8 = R_{B_{12}}, D_9 = R_{B_{11}},\\
B_{15} &= \begin{bmatrix} R_{B_{14}}R_{B_{11}} \\ -R_{B_{13}} \end{bmatrix}, A = R_{A_{15}}C_8, B = D_8L_{B_{15}}, C = R_{A_{15}}E_{11}L_{B_{15}},\\
D &= D_9L_{B_{15}}, E = R_{A_{15}}E_{11}L_{B_{15}}, M = R_AC, N = DL_B, S = CL_M,
\end{aligned} \tag{3.2}$$

where L_A and R_A represent the left and right projection operators associated with the matrix A. The matrices $\widehat{X}_{01}, \widehat{X}_{02}, \widehat{X}_{03}$, $\widehat{Y}_{01}, \widehat{Y}_{02}, \widehat{Y}_{03}$, and $\widehat{Z}_{01}$ denote specific solutions obtained from generalized inverses. Then, the following statements are equivalent, wherein, as mentioned, rank$[A]$ *represents the rank of the matrix A: (i) The system stated in* (2.7) *is solvable; (ii) the system satisfies to*

$$\begin{aligned}
&R_{A_1}F_1 = 0, R_{A_2}F_2 = 0, R_{A_3}F_3 = 0, F_4L_{B_1} = 0, F_5L_{B_2} = 0, F_6L_{B_3} = 0, R_{A_4}F_7 = 0,\\
&F_8L_{B_4} = 0, A_4F_8 = F_7B_4, R_{A_{1j}}E_{j+6} = 0, E_{j+6}L_{B_{1j}} = 0, j \in \{1, 2, 3\},\\
&R_{A_{14}}E_{10}L_{B_{14}} = 0, R_AEL_D = 0, R_CEL_B = 0, R_MR_AE = 0, EL_BL_N = 0;
\end{aligned} \tag{3.3}$$

(iii) the framework yields:

$$\begin{aligned}
&\text{rank}\begin{bmatrix} A_1 & F_1 \end{bmatrix} = \text{rank}[A_1], \text{rank}\begin{bmatrix} A_2 & F_2 \end{bmatrix} = \text{rank}[A_2], \text{rank}\begin{bmatrix} A_3 & F_3 \end{bmatrix} = \text{rank}[A_3],\\
&\text{rank}\begin{bmatrix} B_1 \\ F_4 \end{bmatrix} = \text{rank}[B_1], \text{rank}\begin{bmatrix} B_2 \\ F_5 \end{bmatrix} = \text{rank}[B_2], \text{rank}\begin{bmatrix} B_3 \\ F_6 \end{bmatrix} = \text{rank}[B_3], \text{rank}\begin{bmatrix} A_4 & F_7 \end{bmatrix} = \text{rank}[A_4],\\
&\text{rank}\begin{bmatrix} B_4 \\ F_8 \end{bmatrix} = \text{rank}[B_4], A_4F_8 = F_7B_4,\\
&\text{rank}\begin{bmatrix} E_i & A_{i+4} & C_i & F_{i+3} \\ B_{i+4} & 0 & 0 & B_i \\ F_7D_i & 0 & A_4 & 0 \\ F_i & A_i & 0 & 0 \end{bmatrix} = \text{rank}\begin{bmatrix} C_i & A_{i+4} \\ A_4 & 0 \\ 0 & A_i \end{bmatrix} + \text{rank}\begin{bmatrix} B_{i+4} & B_i \end{bmatrix}, i \in \{1, 2, 3\},\\
&\text{rank}\begin{bmatrix} E_i & C_iF_8 & A_{i+4} & F_{i+3} \\ D_i & B_4 & 0 & 0 \\ B_{i+4} & 0 & 0 & B_i \\ F_i & 0 & A_i & 0 \end{bmatrix} = \text{rank}\begin{bmatrix} D_i & B_4 & 0 \\ B_{i+4} & 0 & B_i \end{bmatrix} + \text{rank}\begin{bmatrix} A_{i+4} \\ A_i \end{bmatrix}, i \in \{1, 2, 3\},
\end{aligned}$$

$$\operatorname{rank}\begin{bmatrix} E_1 & C_1 & 0 & A_5 & 0 & C_1F_8 & F_4 & 0 \\ D_1 & 0 & D_2 & 0 & 0 & B_4 & 0 & 0 \\ 0 & C_2 & -E_2 & 0 & A_6 & 0 & 0 & -F_5 \\ B_5 & 0 & 0 & 0 & 0 & 0 & B_1 & 0 \\ 0 & 0 & B_6 & 0 & 0 & 0 & 0 & B_2 \\ 0 & A_4 & -F_7D_2 & 0 & 0 & 0 & 0 & 0 \\ F_1 & 0 & 0 & A_1 & 0 & 0 & 0 & 0 \\ 0 & 0 & -F_2 & 0 & A_2 & 0 & 0 & 0 \end{bmatrix} = \operatorname{rank}\begin{bmatrix} D_1 & D_2 & B_4 & 0 & 0 \\ B_5 & 0 & 0 & B_1 & 0 \\ 0 & B_6 & 0 & 0 & B_2 \end{bmatrix} + \operatorname{rank}\begin{bmatrix} C_1 & A_5 & 0 \\ C_2 & 0 & A_6 \\ A_4 & 0 & 0 \\ 0 & A_1 & 0 \\ 0 & 0 & A_2 \end{bmatrix},$$

$$\operatorname{rank}\begin{bmatrix} 0 & D_1 & D_3 & 0 & 0 & B_4 & 0 & 0 \\ C_1 & -E_1 & 0 & A_5 & 0 & 0 & -F_4 & 0 \\ C_3 & 0 & E_3 & 0 & A_7 & C_3F_8 & 0 & F_6 \\ 0 & B_5 & 0 & 0 & 0 & 0 & B_1 & 0 \\ 0 & 0 & B_7 & 0 & 0 & 0 & 0 & B_3 \\ A_4 & -F_7D_1 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & -F_1 & 0 & A_1 & 0 & 0 & 0 & 0 \\ 0 & 0 & F_3 & 0 & A_3 & 0 & 0 & 0 \end{bmatrix} = \operatorname{rank}\begin{bmatrix} C_3 & A_7 & 0 \\ C_1 & 0 & A_5 \\ A_4 & 0 & 0 \\ 0 & A_3 & 0 \\ 0 & 0 & A_1 \end{bmatrix} + \operatorname{rank}\begin{bmatrix} 0 & D_2 & 0 & D_1 & B_4 & 0 & 0 & 0 & 0 & 0 \\ D_1 & 0 & D_3 & 0 & 0 & B_4 & 0 & 0 & 0 & 0 \\ B_5 & 0 & 0 & 0 & 0 & 0 & B_1 & 0 & 0 & 0 \\ 0 & B_6 & 0 & 0 & 0 & 0 & 0 & B_2 & 0 & 0 \\ 0 & 0 & B_7 & 0 & 0 & 0 & 0 & 0 & B_3 & 0 \\ 0 & 0 & 0 & B_5 & 0 & 0 & 0 & 0 & 0 & B_1 \end{bmatrix},$$

$$\operatorname{rank}\begin{bmatrix} 0 & 0 & 0 & D_2 & 0 & D_1 & 0 & 0 & 0 & 0 & B_4 & 0 & 0 & 0 & 0 \\ 0 & 0 & D_2 & 0 & D_3 & 0 & 0 & 0 & 0 & 0 & 0 & B_4 & 0 & 0 & 0 \\ 0 & C_2 & E_2 & 0 & 0 & 0 & A_6 & 0 & 0 & 0 & 0 & 0 & F_5 & 0 & 0 \\ C_2 & 0 & 0 & 0 & 0 & 0 & 0 & A_6 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & C_3 & 0 & 0 & E_3 & 0 & 0 & 0 & A_7 & 0 & 0 & C_3F_8 & 0 & 0 & F_6 \\ C_1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & A_5 & 0 & 0 & B_2 & 0 & 0 \\ 0 & 0 & B_6 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & B_2 & 0 & 0 \\ 0 & 0 & 0 & B_6 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & B_2 & 0 \\ 0 & 0 & 0 & 0 & B_7 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & B_3 \\ 0 & 0 & 0 & 0 & 0 & B_5 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & B_1 \\ A_4 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & A_4 & F_7D_2 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & F_2 & 0 & 0 & 0 & A_2 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & A_2 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & F_3 & 0 & 0 & 0 & A_3 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & A_1 & 0 & 0 & 0 & 0 & 0 \end{bmatrix}$$

$$=\operatorname{rank}\begin{bmatrix} 0 & C_2 & A_6 & 0 & 0 & 0 \\ C_2 & 0 & 0 & A_6 & 0 & 0 \\ 0 & C_3 & 0 & 0 & A_7 & 0 \\ C_1 & 0 & 0 & 0 & 0 & A_5 \\ A_4 & 0 & 0 & 0 & 0 & 0 \\ 0 & A_4 & 0 & 0 & 0 & 0 \\ 0 & 0 & A_2 & 0 & 0 & 0 \\ 0 & 0 & 0 & A_2 & 0 & 0 \\ 0 & 0 & 0 & 0 & A_3 & 0 \\ 0 & 0 & 0 & 0 & 0 & A_1 \end{bmatrix} + \operatorname{rank}\begin{bmatrix} 0 & D_2 & 0 & D_1 & B_4 & 0 & 0 & 0 & 0 \\ D_2 & 0 & D_3 & 0 & 0 & B_4 & 0 & 0 & 0 \\ B_6 & 0 & 0 & 0 & 0 & 0 & B_2 & 0 & 0 \\ 0 & B_6 0 & 0 & 0 & 0 & 0 & B_2 & 0 & \\ 0 & 0 & 0 & B_5 & 0 & 0 & 0 & 0 & B_1 \end{bmatrix},$$

$$\operatorname{rank}\begin{bmatrix} 0 & 0 & D_2 & D_3 & D_1 & 0 & 0 & 0 & B_4 & 0 & 0 \\ -C_2 & C_2 & E_2 & 0 & 0 & A_6 & 0 & 0 & F_5 & 0 & 0 \\ C_1 & 0 & 0 & 0 & 0 & 0 & A_5 & 0 & 0 & 0 & 0 \\ 0 & C_3 & 0 & 0 & 0 & 0 & 0 & A_7 & 0 & F_6 & 0 \\ 0 & 0 & B_6 & 0 & 0 & 0 & 0 & 0 & B_2 & 0 & 0 \\ 0 & 0 & 0 & B_7 & 0 & 0 & 0 & 0 & 0 & B_3 & 0 \\ 0 & 0 & 0 & 0 & B_5 & 0 & 0 & 0 & 0 & 0 & B_1 \\ A_4 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & A_4 & 0 & F_7D_3 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & F_2 & 0 & 0 & A_2 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & A_1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & F_3 & 0 & 0 & 0 & A_3 & 0 & 0 & 0 \end{bmatrix} = \operatorname{rank}\begin{bmatrix} D_2 & D_3 & D_1 & B_4 & 0 & 0 & 0 \\ B_6 & 0 & 0 & 0 & B_2 & 0 & 0 \\ 0 & B_7 & 0 & 0 & 0 & B_3 & 0 \\ 0 & 0 & B_5 & 0 & 0 & 0 & B_1 \end{bmatrix} + \operatorname{rank}\begin{bmatrix} C_3 & A_7 & 0 \\ C_1 & 0 & A_5 \\ A_4 & 0 & 0 \\ 0 & A_3 & 0 \\ 0 & 0 & A_1 \end{bmatrix},$$

$$\operatorname{rank}\begin{bmatrix} 0 & D_1 & D_2 & 0 & 0 & 0 & 0 & B_4 & 0 & 0 & 0 & 0 \\ 0 & D_1 & 0 & D_3 & 0 & 0 & 0 & 0 & B_4 & 0 & 0 & 0 \\ C_2 & 0 & E_2 & 0 & A_6 & 0 & 0 & C_2F_8 & C_2F_8 & 0 & F_5 & 0 \\ C_3 & 0 & 0 & E_3 & 0 & A_7 & 0 & 0 & 0 & 0 & 0 & F_6 \\ C_1 & E_1 & 0 & 0 & 0 & 0 & A_5 & 0 & 0 & F_4 & 0 & 0 \\ 0 & B_5 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & B_1 & 0 & 0 \\ 0 & 0 & B_6 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & B_2 & 0 \\ 0 & 0 & 0 & B_7 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & B_3 \\ A_4 & F_7D_1 & 0 & F_7D_3 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & \\ 0 & 0 & F_2 & 0 & A_2 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & F_3 & 0 & A_3 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & F_1 & 0 & 0 & 0 & 0 & A_1 & 0 & 0 & 0 & 0 & 0 \end{bmatrix}$$

$$=\text{rank}\begin{bmatrix} C_2 & A_6 & 0 & 0 \\ C_3 & 0 & A_7 & 0 \\ C_1 & 0 & 0 & A_5 \\ A_4 & 0 & 0 & 0 \\ 0 & A_2 & 0 & 0 \\ 0 & 0 & A_3 & 0 \\ 0 & 0 & 0 & A_1 \end{bmatrix} + \text{rank}\begin{bmatrix} D_1 & D_2 & 0 & B_4 & 0 & 0 & 0 & 0 \\ D_1 & 0 & D_3 & 0 & B_4 & 0 & 0 & 0 \\ B_5 & 0 & 0 & 0 & 0 & B_1 & 0 & 0 \\ 0 & B_6 & 0 & 0 & 0 & 0 & B_2 & 0 \\ 0 & 0 & B_7 & 0 & 0 & 0 & 0 & B_3 \end{bmatrix}. \tag{3.4}$$

In this case, the general solution to the system stated in (2.7) *can be expressed as*

$$X_1 = A_1^{\dagger}F_1 + L_{A_1}X_{11}, X_{11} = A_8^{\dagger}(E_4 - C_4Z_{11}D_4) - A_8^{\dagger}T_1B_8^{\dagger} + L_{A_8}T_2, \tag{3.5}$$
$$X_2 = A_2^{\dagger}F_2 + L_{A_2}X_{22}, X_{22} = A_9^{\dagger}(E_5 - C_5Z_{11}D_5) - A_9^{\dagger}T_4B_9^{\dagger} + L_{A_9}T_5,$$
$$X_3 = A_3^{\dagger}F_3 + L_{A_3}X_{33}, X_{33} = A_{10}^{\dagger}(E_6 - C_6Z_{11}D_6) - A_{10}^{\dagger}T_7B_{10}^{\dagger} + L_{A_{10}}T_6,$$
$$Y_1 = F_4B_1^{\dagger} + Y_{11}R_{B_1}, Y_{11} = R_{A_8}(E_4 - C_4Z_{11}D_4)B_8^{\dagger} + A_8A_8^{\dagger}T_1 + T_8R_{B_8},$$
$$Y_2 = F_5B_2^{\dagger} + Y_{22}R_{B_2}, Y_{22} = R_{A_9}(E_5 - C_5Z_{11}D_5)B_9^{\dagger} + A_9A_9^{\dagger}T_4 + T_9R_{B_9},$$
$$Y_3 = F_6B_3^{\dagger} + Y_{33}R_{B_3}, Y_{33} = R_{A_{10}}(E_6 - C_6Z_{11}D_6)B_{10}^{\dagger} + A_{10}A_{10}^{\dagger}T_7 + T_{10}R_{B_{10}},$$
$$Z = A_4^{\dagger}F_7 + L_{A_4}F_8B_4^{\dagger} + L_{A_4}Z_{11}R_{B_4}, \tag{3.6}$$
$$Z_{11} = Z_a + L_{A_{11}}L_{A_{14}}Z_1 + Z_2R_{B_{14}}R_{B_{11}} + L_{A_{11}}Z_3R_{B_{12}} + L_{A_{12}}Z_4R_{B_{11}}, \tag{3.7}$$
$$Z_{11} = Z_b + L_{A_{13}}Z_5 + Z_6R_{B_{13}}, \tag{3.8}$$

where Z_a and Z_b are particular solutions derived through projections and generalized inverses, where the indices a and b correspond to specific parameters or dimensions relevant to the solution. Additionally, we have that

$$\begin{bmatrix} Z_1 \\ Z_5 \end{bmatrix} = A_{15}^{\dagger}(E_{11} - C_8Z_3D_8 - C_9Z_4D_9) - A_{15}^{\dagger}Z_7B_{15} + L_{A_{15}}Z_8, \tag{3.9}$$
$$\begin{bmatrix} Z_2 & Z_6 \end{bmatrix} = R_{A_{15}}(E_{11} - C_8Z_3D_8 - C_9Z_4D_9)B_{15}^{\dagger} + A_{15}A_{15}^{\dagger}Z_7 + Z_9R_{B_{15}},$$
$$Z_3 = A^{\dagger}EB^{\dagger} - A^{\dagger}CM^{\dagger}EB^{\dagger} - A^{\dagger}SC^{\dagger}EN^{\dagger}DB^{\dagger} - A^{\dagger}SZ_{10}R_NDB^{\dagger} + L_AZ_{15} + Z_{12}R_B,$$
$$Z_4 = M^{\dagger}ED^{\dagger} + S^{\dagger}SC^{\dagger}EN^{\dagger} + L_ML_SZ_{13} + L_MZ_{10}R_N + Z_{14}R_D, \tag{3.10}$$

where $T_1, T_2, T_4, \ldots, T_{10}, Z_7, \ldots, Z_{10}, Z_{12}, \ldots, Z_{15}$ are suitable matrices over $\mathbb{H}$.

Remark 3.1. *The variable Z_{11} is defined in two different forms to reflect its derivation under different conditions of solvability. On one hand, the expression stated in* (3.7) *represents Z_{11} as derived from the solvability conditions of the systems given in* (3.13)*,* (3.15)*, and* (3.17)*, where it incorporates projections and generalized inversion operations involving matrices $A_{11}, A_{12}, A_{14}, B_{11}$, and B_{14}. On the other hand, the equation presented in* (3.8) *provides an alternative expression for Z_{11} based on the consistency conditions of the system described in* (2.8)*, involving matrices A_{13} and B_{13}. These two forms ensure the comprehensive solution and consistency of the matrix equations under different scenarios.*

The flowchart presented in Figure 1 summarizes the steps involved in the proof of Theorem 3.1.

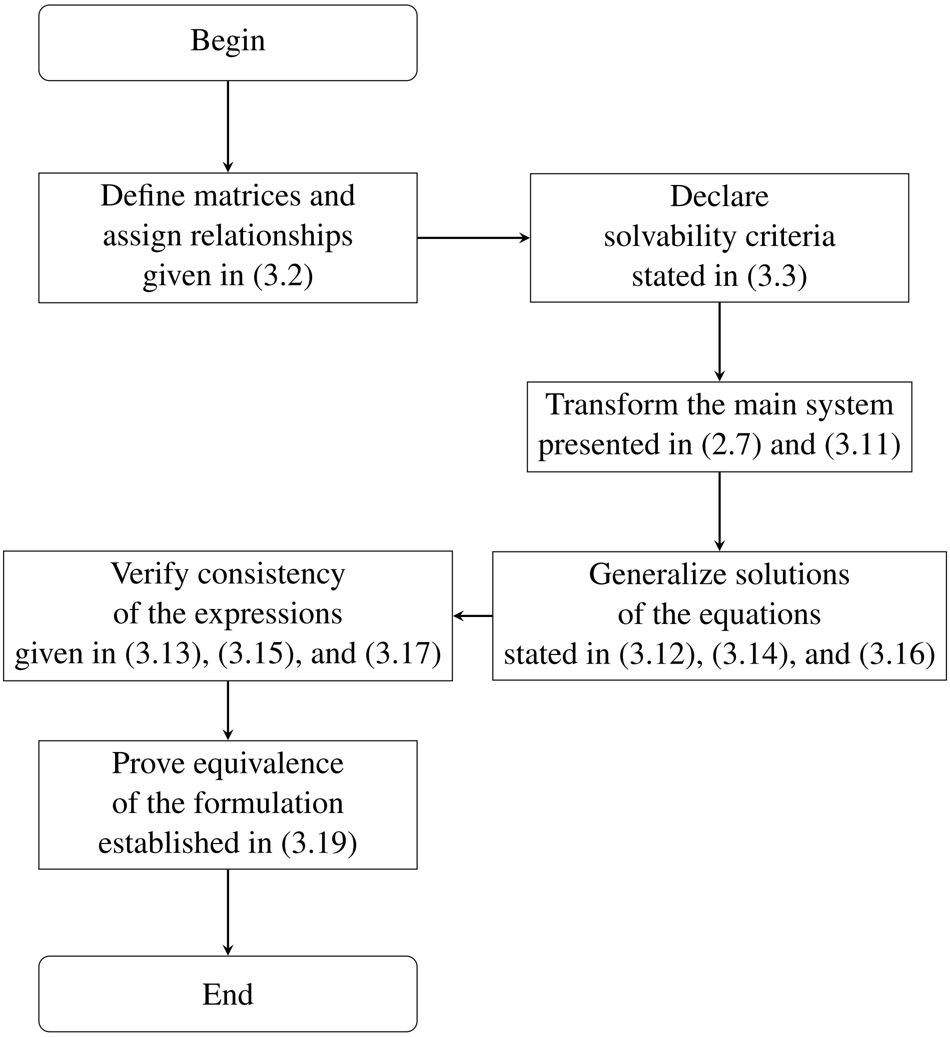

Figure 1. Flowchart illustrating the proof steps of Theorem 3.1.

Proof of Theorem 3.1. To prove the relationship between statements (i) and (ii) of Theorem 3.1, we state the expressions given in (2.7) in their alternative forms represented as

$$\begin{aligned} &A_1X_1 = F_1, Y_1B_1 = F_4, A_4Z_1 = F_7, ZB_4 = F_8, A_5X_1 + Y_1B_5 + C_1ZD_1 = E_1,\\ &A_2X_2 = F_2, Y_2B_2 = F_5, A_4Z = F_7, ZB_4 = F_8, A_6X_2 + Y_2B_6 + C_2ZD_2 = E_2,\\ &A_3X_3 = F_3, Y_3B_3 = F_6, A_4Z_1 = F_7, ZB_4 = F_8, A_7X_3 + Y_3B_3 + C_3ZD_3 = E_3. \end{aligned} \tag{3.11}$$

By using Lemmas 3.2–3.4, the general solution to $A_1X_1 = F_1, Y_1B_1 = F_4$ and $A_4Z = F_7, ZB_4 = F_8$ is given by

$$X_1 = A_1^{\dagger}F_1 + L_{A_1}X_{11}, Y_1 = F_4B_1^{\dagger} + Y_{11}R_{B_1}, Z = A_4^{\dagger}F_7 + L_{A_4}F_8B_4^{\dagger} + L_{A_4}Z_{11}R_{B_4}, \tag{3.12}$$

where X_{11}, Y_{11} are arbitrary quaternionic matrices of suitable dimensions, and Z_{11} is defined in (3.7). Substituting the expression formulated in (3.12) in the corresponding equation of the system presented (3.11), we obtain

$$A_8X_{11} + Y_{11}B_8 + C_4Z_{11}D_4 = E_4. \tag{3.13}$$

Similarly, the general solution to $A_2X_2 = F_2$ and $Y_2B_2 = F_5$, by utilizing Lemmas 3.2 and 3.3, with Z_{11} as defined in (3.7), is stated as

$$X_2 = A_2^\dagger F_2 + L_{A_2}X_{22}, Y_2 = F_5B_2^\dagger + Y_{22}R_{B_2}, \tag{3.14}$$

where X_{22} and Y_{22} are any quaternionic matrices of suitable dimensions. Incorporating the terms presented in (3.12) and (3.14) into the corresponding component of the system defined in (3.11), with Z_{11} as defined in (3.7), we deduce

$$A_9X_2 + Y_{22}B_9 + C_5Z_{11}D_5 = E_5. \tag{3.15}$$

Again, on the same line, the general solution to $A_3X_3 = F_3$ and $Y_3B_3 = F_6$, by employing Lemmas 3.2 and 3.3, with Z_{11} as defined in (3.8), is established as

$$X_3 = A_3^\dagger F_3 + L_{A_3}X_{33}, Y_3 = F_6B_3^\dagger + Y_{33}R_{B_3}, \tag{3.16}$$

where X_{33} and Y_{33} are any quaternionic matrices of suitable dimensions. Substituting the expressions stated in (3.12) and (3.16) in the corresponding equation of the formula given in (3.11), with Z_{11} as defined in (3.8), we reach

$$A_{10}X_{33} + Y_{33}B_{10} + C_6Z_{11}D_6 = E_6. \tag{3.17}$$

The solvability conditions of the systems established in (3.11) are equivalent to the solvability conditions of the system given in (3.13), (3.15), and (3.17). The system formulated from (3.13), (3.15), and (3.17) is consistent if and only if we have consistency in the system defined as

$$A_{11}Z_{11}B_{11} = E_7, A_{12}Z_{11}B_{12} = E_8, A_{13}Z_{11}B_{13} = E_9. \tag{3.18}$$

Now, we investigate the system described in (3.18). The first two equations of this system are consistent if and only if all the rank equalities from the third line of the expression defined in (3.3) are met with $R_{A_{13}}E_9 = 0$ and $E_9L_{B_{13}} = 0$. In this case, the common solution of the first two equations of the system formulated in (3.18) can be stated as in (3.7). In addition, the last equation of the expression given in (3.18) is consistent if and only if $R_{A_{13}}E_9 = 0$ and $E_9L_{B_{13}} = 0$. In this case, the general solution to the last equation of the system presented in (3.18) is given by the expression defined in (3.8). From the formulations given in (3.7) and (3.8), we get

$$A_{15}\begin{bmatrix} Z_1 \\ Z_5 \end{bmatrix} + \begin{bmatrix} Z_2 & Z_6 \end{bmatrix} + C_8Z_3D_8 + C_9Z_4D_9 = 0. \tag{3.19}$$

If the equation presented in (3.19) is consistent, all rank equalities specified in the corresponding equation of the system established in (3.3) must be satisfied. Under this condition, the general solution for the system identified in (3.19) is outlined in the expressions given by (3.9) and (3.10). To demonstrate the equivalence of statements (ii) and (iii) of Theorem 3.1, we examine the conditions stated in the system presented in (3.3) and compare them with the rank equalities of the system read in (3.4). As an illustration, we highlight how the third-last equality of the system formulated in (3.3) corresponds to its counterpart in the system defined in (3.4). Similar logic applies to establish the equivalence of other conditions. By applying Lemma 3.1 and using the elementary row operation, we have

$$\begin{aligned}\operatorname{rank}[R_C E L_B] =&\operatorname{rank}\begin{bmatrix} E & C \\ B & 0 \end{bmatrix} - \operatorname{rank}[B] - \operatorname{rank}[C] = \operatorname{rank}\begin{bmatrix} R_{A_{15}}E_{11}L_{B_{15}} & R_{A_{15}}C_9 \\ D_8 L_{B_{15}} & 0 \end{bmatrix} - \operatorname{rank}[R_{A_{15}}C_9] - \operatorname{rank}[D_8 L_{B_{15}}] \\ =&\operatorname{rank}\begin{bmatrix} E_{11} & C_9 & A_{15} \\ D_8 & 0 & 0 \\ B_{15} & 0 & 0 \end{bmatrix} - \operatorname{rank}\begin{bmatrix} C_9 & A_{15} \end{bmatrix} - \operatorname{rank}\begin{bmatrix} D_8 \\ B_{15} \end{bmatrix} \\ =&\operatorname{rank}\begin{bmatrix} Z_b - Z_a & L_{A_{12}} & L_{A_{11}}L_{A_{14}} & -L_{A_{13}} \\ R_{B_{12}} & 0 & 0 & 0 \\ R_{B_{14}}B_{11} & 0 & 0 & 0 \\ -R_{B_{13}} & 0 & 0 & 0 \end{bmatrix} - \operatorname{rank}\begin{bmatrix} L_{A_{12}} & L_{A_{11}}L_{A_{14}} & -L_{A_{13}} \end{bmatrix} - \operatorname{rank}\begin{bmatrix} R_{B_{12}} \\ R_{B_{14}}R_{B_{11}} \\ R_{B_{13}} \end{bmatrix}.\end{aligned}$$

Recalling that Z_b is a special solution to $A_{13}Z_{11}B_{13}$ and Z_a is a special solution to the pair of equations $A_{11}Z_{11}B_{11} = E_7$ and $A_{12}Z_{11}B_{12} = E_8$ —where Z_{11} is defined in (3.8)—, we use these facts to attain

$$\begin{aligned}\operatorname{rank}[R_C E L_B] =&\operatorname{rank}\begin{bmatrix}
0 & 0 & 0 & D_2 & 0 & D_1 & 0 & 0 & 0 & 0 & B_4 & 0 & 0 & 0 & 0 \\
0 & 0 & D_2 & 0 & D_3 & 0 & 0 & 0 & 0 & 0 & 0 & B_4 & 0 & 0 & 0 \\
0 & C_2 & E_2 & 0 & 0 & 0 & A_6 & 0 & 0 & 0 & 0 & 0 & F_5 & 0 & 0 \\
C_2 & 0 & 0 & 0 & 0 & 0 & 0 & A_6 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & C_3 & 0 & 0 & E_3 & 0 & 0 & 0 & A_7 & 0 & 0 & C_3F_8 & 0 & 0 & F_6 \\
C_1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & A_5 & 0 & 0 & B_2 & 0 & 0 \\
0 & 0 & B_6 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & B_2 & 0 & 0 \\
0 & 0 & 0 & B_6 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & B_2 & 0 \\
0 & 0 & 0 & 0 & B_7 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & B_3 \\
0 & 0 & 0 & 0 & 0 & B_5 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & B_1 \\
A_4 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & A_4 & F_7D_2 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & F_2 & 0 & 0 & 0 & A_2 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & A_2 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & F_3 & 0 & 0 & 0 & A_3 & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & A_1 & 0 & 0 & 0 & 0 & 0
\end{bmatrix} \\
&- \operatorname{rank}\begin{bmatrix}
0 & C_2 & A_6 & 0 & 0 & 0 \\
C_2 & 0 & 0 & A_6 & 0 & 0 \\
0 & C_3 & 0 & 0 & A_7 & 0 \\
C_1 & 0 & 0 & 0 & 0 & A_5 \\
A_4 & 0 & 0 & 0 & 0 & 0 \\
0 & A_4 & 0 & 0 & 0 & 0 \\
0 & 0 & A_2 & 0 & 0 & 0 \\
0 & 0 & 0 & A_2 & 0 & 0 \\
0 & 0 & 0 & 0 & A_3 & 0 \\
0 & 0 & 0 & 0 & 0 & A_1
\end{bmatrix} - \operatorname{rank}\begin{bmatrix}
0 & D_2 & 0 & D_1 & B_4 & 0 & 0 & 0 & 0 \\
D_2 & 0 & D_3 & 0 & 0 & B_4 & 0 & 0 & 0 \\
B_6 & 0 & 0 & 0 & 0 & 0 & B_2 & 0 & 0 \\
0 & B_6 & 0 & 0 & 0 & 0 & B_2 & 0 & \\
0 & 0 & 0 & B_5 & 0 & 0 & 0 & 0 & B_1
\end{bmatrix}.\end{aligned}$$

Consequently, the condition $\operatorname{rank}[R_C E L_B] = 0$ aligns with the third-last rank equality stated in (3.4), where Z_{11} is defined in (3.7). Using a similar approach, we can verify the remaining rank equalities. Thus, the proof of Theorem 3.1 is completed. □

Remark 3.2. *By setting each of the specific matrices in our system presented in* (2.7) *to zero, we can successively derive the principal results ranging from the systems defined in* (2.3) *to* (2.6).

4. Solving the system: algorithm, flowchart, and illustration

In this section, we elucidate the steps for solving the system described in (2.7) based on the principles established in Theorem 3.1. These steps are illustrated both algorithmically and visually through a flowchart. A numerical illustration is also provided to offer a practical perspective on its application.

4.1. Procedure and schematic overview

Algorithm 1 is a structured approach that offers both efficiency and precision. The flowchart in Figure 2 provides a visual and textual representation of Algorithm 1, ensuring that the process is easily understandable and implementable, aiding in its comprehension.

Algorithm 1 Solution the system stated in (2.7).

Require: Matrices $\{A_i, B_i\}_{i=1}^{7}$, $\{C_j, D_j\}_{j=1}^{3}$, and $\{E_k\}_{k=1}^{6}$ over $\mathbb{H}$.

Ensure: $X_1, X_2, X_3, Y_1, Y_2, Y_3$, and Z or a message of inconsistency.

Calculate $A_8 = A_5 L_{A_1}$, $B_8 = R_{B_1} B_5$, $C_4 = C_1 L_{A_4}$, $D_4 = R_{B_4} D_1$
Compute $F_1 = A_1^{\dagger} E_1$, $\widehat{X}_{01} = A_1^{\dagger} F_1$
Determine $F_4 = E_4 B_1^{\dagger}$, $\widehat{Y}_{01} = F_4 B_1^{\dagger}$
State $A_9 = A_6 L_{A_2}$, $B_9 = R_{B_2} B_6$, $C_5 = C_2 L_{A_5}$, $D_5 = R_{B_5} D_2$
Establish $F_2 = A_2^{\dagger} E_2$, $\widehat{X}_{02} = A_2^{\dagger} F_2$
Obtain $F_5 = E_5 B_2^{\dagger}$, $\widehat{Y}_{02} = F_5 B_2^{\dagger}$
Reach $A_{10} = A_7 L_{A_3}$, $B_{10} = R_{B_3} B_7$, $C_6 = C_3 L_{A_6}$, $D_6 = R_{B_6} D_3$
Attain $F_3 = A_3^{\dagger} E_3$, $\widehat{X}_{03} = A_3^{\dagger} F_3$
Get $F_6 = E_6 B_3^{\dagger}$, $\widehat{Y}_{03} = F_6 B_3^{\dagger}$
Generate $A_{11} = R_{A_8} C_4$, $B_{11} = D_4 L_{B_8}$, $E_7 = R_{A_8} E_4 L_{B_8}$
Retrieve $F_7 = A_4^{\dagger} E_7$, $F_8 = E_8 B_4^{\dagger}$
Produce $\widehat{Z}_{01} = A_4^{\dagger} F_7 + L_{A_4} F_8 B_4^{\dagger}$
Gauge $\{A_i, B_i\}_{i=12}^{15}$, $\{C_j, D_j\}_{j=8}^{9}$, $\{E_k\}_{k=8}^{11}$, A, B, C, D, E, M, N, and S using the system stated in (3.2).
if the conditions stated in the systems presented in (3.3) or (3.4) are met **then**
    Compute $X_1, X_2, X_3, Y_1, Y_2, Y_3$, and Z using the equations formulated in (3.5)-(3.6).
else
    Output: "inconsistent system".
return The computed matrices or a message of inconsistent system.

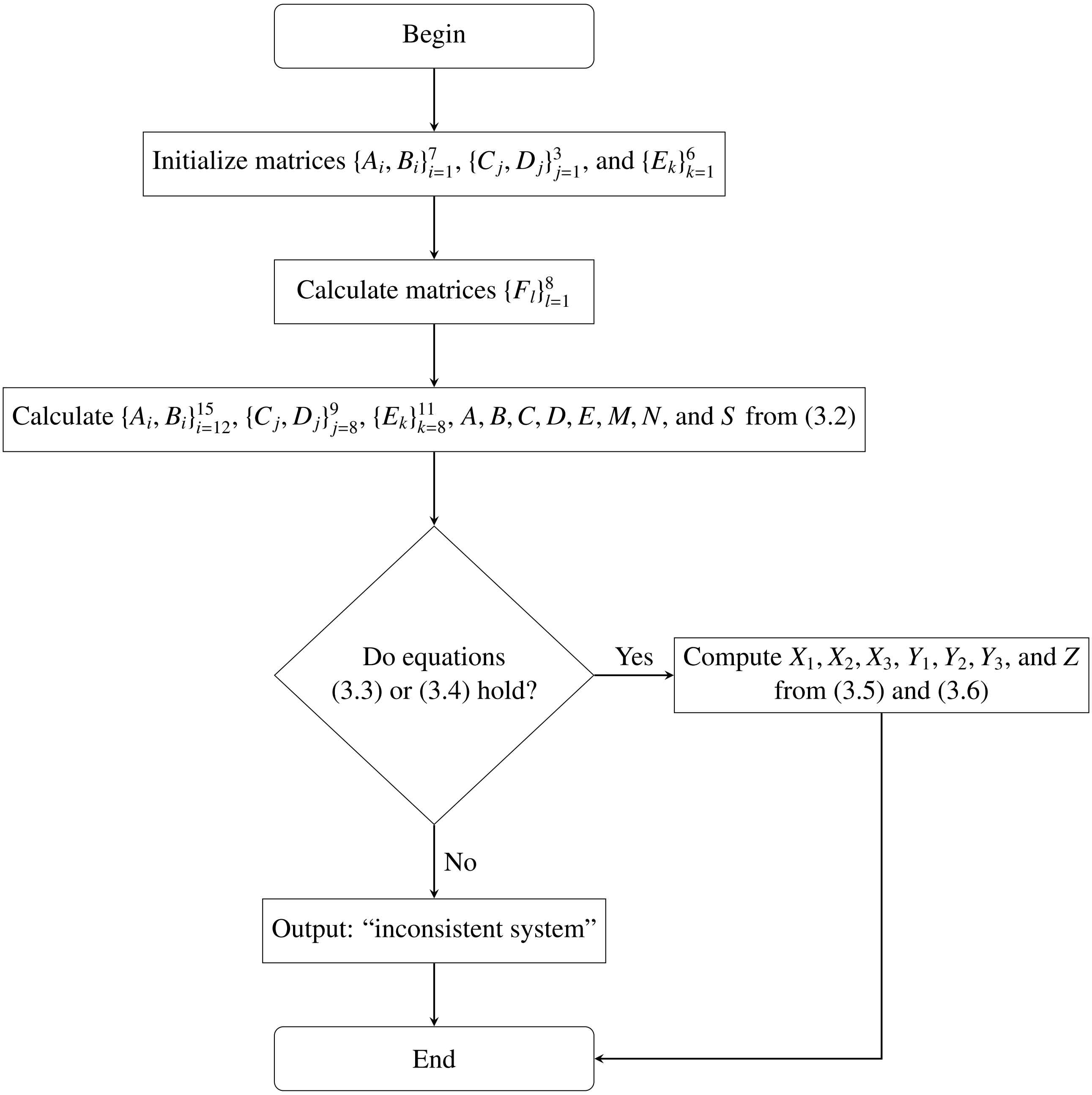

Figure 2. Flowchart of Algorithmic 1 for the solution of the system presented in (2.7).

4.2. Numerical example

To show the application of Algorithm 1 and the relationships defined in Theorem 3.1, we present the following numerical example. The matrices provided here were carefully chosen to illustrate the steps in the algorithm and their connection to the theoretical results. For readers interested in replicating the results or understanding the computational aspects in more detail, we include a `Python` script in Appendix B. This script implements Algorithm 1 step-by-step, following the exact sequence described in Theorem 3.1 and applied to the matrices in this example. The `Python` code allows for verification of the computations and provides a framework for further exploration of similar matrix systems.

The numerical example is as follows. We start by initializing the matrices $\{A_i, B_i\}_{i=1}^{7}$, $\{C_j, D_j\}_{j=1}^{3}$, and $\{E_k\}_{k=1}^{6}$ presented as

$$A_1 = \begin{bmatrix} 1 & 12\mathrm{j}+1 & \mathrm{i} \\ 5+\mathrm{j} & \mathrm{i}+\mathrm{k} & 2+\mathrm{j} \end{bmatrix}, A_2 = \begin{bmatrix} 2+\mathrm{k} & \mathrm{i} \\ 1+\mathrm{j} & \mathrm{i}+\mathrm{k} \\ 1+\mathrm{j} & 7+\mathrm{k} \end{bmatrix}, A_3 = \begin{bmatrix} \mathrm{i}+5 \\ \mathrm{j}+\mathrm{k} \\ 1+\mathrm{i} \end{bmatrix}, A_4 = \begin{bmatrix} 1+\mathrm{k} & \mathrm{i}+\mathrm{j} & 2+5\mathrm{k} \\ \mathrm{i}+\mathrm{j} & 4+6\mathrm{k} & 3+\mathrm{i} \end{bmatrix},$$

$$A_5 = \begin{bmatrix} 1+\mathrm{j} & \mathrm{i}+\mathrm{k} & 5+7\mathrm{j} \\ 2+5\mathrm{k} & 7+\mathrm{i} & \mathrm{j}+\mathrm{k} \end{bmatrix}, A_6 = \begin{bmatrix} \mathrm{i}+5 & \mathrm{j}+\mathrm{k} \\ 2-7\mathrm{k} & 10 \end{bmatrix}, A_7 = \begin{bmatrix} \mathrm{i}+\mathrm{j} \\ \mathrm{k}+1 \end{bmatrix}, B_1 = \begin{bmatrix} 4\mathrm{j} & 2+\mathrm{i}+\mathrm{k} & 7+\mathrm{i}+\mathrm{j} \\ 1+\mathrm{k} & 2\mathrm{j} & \mathrm{i}+\mathrm{j} \end{bmatrix},$$

$$B_2 = \begin{bmatrix} 1+5\mathrm{k} & \mathrm{i}+\mathrm{j} \\ \mathrm{i}+8 & 2+\mathrm{k} \end{bmatrix}, B_3 = \begin{bmatrix} \mathrm{i}+\mathrm{j} & 2+\mathrm{k} & 3+\mathrm{i} \\ 1+2\mathrm{k} & 7 & \mathrm{i}+5\mathrm{j} \end{bmatrix}, B_4 = \begin{bmatrix} 1+\mathrm{k} & \mathrm{i}+\mathrm{j} & 2\mathrm{k} \\ 7\mathrm{i}+5\mathrm{j} & 1+\mathrm{k} & 2+7\mathrm{j} \end{bmatrix},$$

$$B_5 = \begin{bmatrix} 1+\mathrm{i} & 2+\mathrm{j} & 7\mathrm{k} \\ \mathrm{j} & \mathrm{k} & \mathrm{i}+\mathrm{k} \end{bmatrix}, B_6 = \begin{bmatrix} \mathrm{k} & \mathrm{i} & 2+\mathrm{j} \\ \mathrm{j} & 2-\mathrm{k} & 2+\mathrm{i}+\mathrm{j} \\ \mathrm{i}+7 & \mathrm{j} & 2-\mathrm{k} \end{bmatrix}, B_7 = \begin{bmatrix} 1+\mathrm{j} & 2 & \mathrm{k} \\ \mathrm{j}-\mathrm{k} & 5+8\mathrm{j} & \mathrm{i} \end{bmatrix},$$

$$C_1 = \begin{bmatrix} 1+\mathrm{k} & \mathrm{i}+\mathrm{j} & 1 \\ \mathrm{j} & 2+\mathrm{i} & 5+\mathrm{k} \end{bmatrix}, C_2 = \begin{bmatrix} 1+\mathrm{j} & \mathrm{i}-\mathrm{k} & 5 \\ \mathrm{i}-9 & \mathrm{j}+\mathrm{k} & 7+\mathrm{i} \end{bmatrix}, C_3 = \begin{bmatrix} 2\mathrm{i} & \mathrm{k}+7 & 5-\mathrm{j} \\ 2 & \mathrm{j} & 3\mathrm{k} \end{bmatrix},$$

$$D_1 = \begin{bmatrix} 1+\mathrm{j} & \mathrm{k}+3 & \mathrm{i}+\mathrm{j} \\ \mathrm{i} & \mathrm{j}+2+\mathrm{k} & 7+\mathrm{i} \end{bmatrix}, D_2 = \begin{bmatrix} 1+\mathrm{j} & \mathrm{i}+\mathrm{k} & 15 \\ 7\mathrm{k} & 3+7\mathrm{j} & 1+\mathrm{i} \end{bmatrix}, D_3 = \begin{bmatrix} 1 & \mathrm{k} & 5+\mathrm{j} \\ \mathrm{j} & 2 & \mathrm{i} \end{bmatrix}.$$

Next, based on the steps of Algorithm 1. we compute the auxiliary quaternionic matrices as

$$A_8 = A_5L_{A_1}, B_8 = R_{B_1}B_5, C_4 = C_1L_{A_4}, D_4 = R_{B_4}D_1, F_1 = A_1^{\dagger}E_1, \widehat{X}_{01} = A_1^{\dagger}F_1, F_4 = E_4B_1^{\dagger}, \widehat{Y}_{01} = F_4B_1^{\dagger},$$
$$A_9 = A_6L_{A_2}, B_9 = R_{B_2}B_6, C_5 = C_2L_{A_5}, D_5 = R_{B_5}D_2, F_2 = A_2^{\dagger}E_2, \widehat{X}_{02} = A_2^{\dagger}F_2, F_5 = E_5B_2^{\dagger}, \widehat{Y}_{02} = F_5B_2^{\dagger},$$
$$A_{10} = A_7L_{A_3}, B_{10} = R_{B_3}B_7, C_6 = C_3L_{A_6}, D_6 = R_{B_6}D_3, F_3 = A_3^{\dagger}E_3, \widehat{X}_{03} = A_3^{\dagger}F_3, F_6 = E_6B_3^{\dagger}, \widehat{Y}_{03} = F_6B_3^{\dagger}.$$

After calculating these matrices, we verify if the conditions of Theorem 3.1 presented in (3.3) or (3.4) hold. As the conditions hold, we compute the solution matrices $X_1, X_2, X_3, Y_1, Y_2, Y_3$, and Z as

$$X_1 = \begin{bmatrix} \mathrm{i} & 2 & \mathrm{k}+7 \\ \mathrm{i}+\mathrm{j} & 3 & \mathrm{i}+\mathrm{j}+\mathrm{k} \\ 1+\mathrm{i} & 5\mathrm{k} & 5 \end{bmatrix}, X_2 = \begin{bmatrix} 7+\mathrm{i} & 6k & 10 \\ \mathrm{j}+\mathrm{k} & \mathrm{i}+\mathrm{j} & 5+\mathrm{k} \end{bmatrix}, X_3 = \begin{bmatrix} \mathrm{i}+\mathrm{j}+\mathrm{k}, 2+5\mathrm{j}+7\mathrm{k}, 15 \end{bmatrix},$$

$$Y_1 = \begin{bmatrix} \mathrm{k} & 1+\mathrm{i} \\ 2+\mathrm{i} & \mathrm{j}-1+\mathrm{k} \end{bmatrix}, Y_2 = \begin{bmatrix} \mathrm{i} & 1 & \mathrm{j}+\mathrm{k} \\ \mathrm{k}+\mathrm{j} & 5+\mathrm{j} & \mathrm{i}+\mathrm{j} \end{bmatrix}, Y_3 = \begin{bmatrix} \mathrm{k}+\mathrm{i} & 1 \\ \mathrm{j} & 2+\mathrm{i}+\mathrm{j} \end{bmatrix}, Z = \begin{bmatrix} 1+\mathrm{j} & 5+\mathrm{j}+\mathrm{k} \\ \mathrm{i}+7 & 2+7\mathrm{k} \\ +5 & \mathrm{i}+\mathrm{j}+\mathrm{k} \end{bmatrix}.$$

This example follows the steps of Algorithm 1, showing its applicability and of Theorem 3.1. By processing these matrices, Algorithm 1 yields a consistent system with the provided solution matrices.

5. Hermitian solution and illustrative example

Theorem 3.1 provided a general framework for solving quaternionic matrix equations without imposing additional constraints. In contrast, this section focuses on a specific type of solution: the η-Hermitian one. This solution is important in contexts where the solutions of quaternionic matrix must satisfy certain symmetry properties, which are crucial in fields like quantum mechanics and control theory. In this context, we state the necessary and sufficient conditions for the system presented in (2.8) to have an η-Hermitian solution. We also provide the general solution when this system is consistent.

5.1. Solvability equivalence

To clarify the conditions under which the quaternionic matrix equations have an η-Hermitian solution, we present Theorem 5.1. An η-Hermitian solution is one where the resulting matrix X satisfies a generalized Hermitian property. Specifically, a matrix X is considered η-Hermitian if it equals its own η-conjugate transpose, that is, $X = X_\eta^*$, exhibiting this specific symmetry under the η-conjugate operation.

Theorem 5.1. *Let $\{A_i\}_{i=1}^7$, $\{B_j\}_{j=1}^4$, and $\{C_k, D_k, E_k\}_{k=1}^3$ be quaternionic matrices of suitable dimension over $\mathbb{H}$. The relationships defining the conditions for the system to have an η-Hermitian solution are stated as*

$$
\begin{aligned}
&F_1 = A_1^\dagger E_1, F_2 = A_2^\dagger E_2, F_3 = A_3^\dagger E_3, \widehat{X}_{01} = A_1^\dagger F_1, \widehat{X}_{02} = A_2^\dagger F_2, \widehat{X}_{03} = A_3^\dagger F_3,\\
&E_4 = E_1 - (A_5\widehat{X}_{01} + (A_5\widehat{X}_{01})_\eta^* + C_1\widehat{Z}_{01}(C_1)_\eta^*), E_5 = E_2 - (A_6\widehat{X}_{02} + (A_6\widehat{X}_{02})_\eta^* + C_2\widehat{Z}_{01}(C_2)_\eta^*),\\
&E_6 = E_3 - (A_7\widehat{X}_{03} + (A_7\widehat{X}_{03})_\eta^* + C_3\widehat{Z}_{01}(C_3)_\eta^*), F_7 = C_4^\dagger D_4, F_8 = L_{C_4}(C_4^\dagger D_4)_\eta^*,\\
&C_4 = \begin{bmatrix} A_4 \\ (B_4)_\eta^* \end{bmatrix}, D_4 = \begin{bmatrix} F_7 \\ (F_8)_\eta^* \end{bmatrix}, \widehat{Z}_{01} = C_4^\dagger D_4 + L_{C_4}(C_4^\dagger D_4)_\eta^*,\\
&A_8 = A_5 L_{A_1}, C_5 = C_1 L_{A_4}, A_9 = A_6 L_{A_2}, C_6 = C_2 L_{A_4}, A_{10} = A_7 L_{A_3}, C_7 = C_3 L_{A_4},\\
&A_{11} = R_{A_8} C_5, A_{12} = R_{A_9} C_6, E_7 = R_{A_8} E_4 (R_{A_8})_\eta^*, E_8 = R_{A_9} E_5 (R_{A_9})_\eta^*,\\
&A_{13} = R_{A_{10}} C_7, A_{14} = A_{12} L_{A_{11}}, A_{15} = [L_{A_{11}} L_{A_{14}}, -L_{A_{13}}], E_9 = R_{A_{10}} E_6 (R_{A_{10}})_\eta^*,\\
&E_{10} = E_8 - A_{12} A_{11}^\dagger E_7 (A_{11}^\dagger)_\eta^* (A_{12})_\eta^*, D_7 = R_{A_{14}} A_{12}, E_{11} = Z_b - \Psi, Z_b = A_{13}^\dagger E_9 B_{13}^\dagger, C_8 = L_{A_{11}}, C_9 = L_{A_{12}},\\
&\Psi = A_{11}^\dagger E_7 (A_{11}^\dagger)_\eta^* + L_{A_{11}} A_{14}^\dagger E_{10} (A_{12}^\dagger)_\eta^* - L_{A_{11}} A_{14}^\dagger A_{12} D_7^\dagger R_{A_{14}} E_{10} (A_{12}^\dagger)_\eta^* + D_7^\dagger R_{A_{14}} E_{10} (A_{14}^\dagger)_\eta^* (L_{A_{11}})_\eta^*,\\
&A = R_{A_{15}} C_8, C = R_{A_{15}} C_9, E = R_{A_{15}} E_{11} (R_{A_{15}})_\eta^*, M = R_A C, N = A_\eta^* (R_C)_\eta^*, S = C L_M,
\end{aligned}
$$

where L_A and R_A represent the left and right projectors of the matrix A, η represents the conjugate transpose operation, $\widehat{X}_{01}, \widehat{X}_{02}, \widehat{X}_{03}$ are particular solutions of the equations $A_1X_1 = F_1$, $A_2X_2 = F_2$, and $A_3X_3 = F_3$ respectively, $\widehat{Z}_{01}$ is a particular solution of the equation $C_4Z = D_4$, and Ψ is a specific matrix defined in terms of A_{11}, A_{12}, A_{14}, D_7, E_7, and E_{10}, with A_η^* denoting the conjugate transpose of the quaternionic matrix A. Then, the equivalence conditions are: (i) the system stated in (2.7) *is solvable; (ii) the relationships $R_{A_1}F_1 = 0, R_{A_2}F_2 = 0$, $R_{A_3}F_3 = 0$, $R_{C_4}D_4 = 0$, $C_4(D_4)_\eta^* = D_4(C_4)_\eta^*$, $R_{A_{1j}}E_{j+6}(R_{A_{1j}})_\eta^* = 0$, for $j \in \{1, 2, 3\}$, $R_{A_{14}}E_{10}(R_{A_{14}})_\eta^* = 0$, $R_AE(R_A)_\eta^* = 0$, and $R_MR_AE = 0$ hold; (iii) the ranks of the matrix concatenations are specified as*

$$
\begin{aligned}
&\mathrm{rank}[A_1\ F_1] = \mathrm{rank}[A_1],\\
&\mathrm{rank}[A_2\ F_2] = \mathrm{rank}[A_2],\\
&\mathrm{rank}[A_3\ F_3] = \mathrm{rank}[A_3],\\
&C_4(D_4)_\eta^* = D_4(C_4)_\eta^*,
\end{aligned}
$$

$$\operatorname{rank}\begin{bmatrix} F_7 & A_4 \\ (F_8)^*_\eta & (B_4)^*_\eta \end{bmatrix} = \operatorname{rank}\begin{bmatrix} E_i & A_{i+4} & C_i & 0 & (F_i)^*_\eta \\ (C_i)^*_\eta & 0 & 0 & (A_4)^*_\eta & 0 \\ F_7(C_i)^*_\eta & 0 & A_4 & 0 & 0 \\ (A_{i+4})^*_\eta & 0 & 0 & 0 & (A_i)^*_\eta \\ F_i & A_i & 0 & 0 & 0 \end{bmatrix} = 2\operatorname{rank}\begin{bmatrix} C_i & A_{i+4} \\ A_4 & 0 \\ 0 & A_i \end{bmatrix}, i \in \{1,2,3\},$$

$$\operatorname{rank}\begin{bmatrix} E_2 & C_2 & 0 & A_6 & 0 & 0 & (F_2)^*_\eta & 0 \\ (C_2)^*_\eta & 0 & (C_1)^*_\eta & 0 & 0 & (A_4)^*_\eta & 0 & 0 \\ 0 & C_1 & -E_1 & 0 & A_5 & -C_1(F_7)^*_\eta & 0 & -(F_1)^*_\eta \\ (A_6)^*_\eta & 0 & 0 & 0 & 0 & 0 & (A_2)^*_\eta & 0 \\ 0 & 0 & (A_5)^*_\eta & 0 & 0 & 0 & 0 & (A_1)^*_\eta \\ 0 & A_4 & -F_7D_2 & 0 & 0 & 0 & 0 & 0 \\ F_7(C_2)^*_\eta & A_4 & 0 & 0 & 0 & 0 & 0 & 0 \\ F_2 & 0 & 0 & A_2 & 0 & 0 & 0 & 0 \\ 0 & 0 & -F_1 & 0 & A_1 & 0 & 0 & 0 \end{bmatrix} = 2\operatorname{rank}\begin{bmatrix} C_2 & A_6 & 0 \\ C_1 & 0 & A_5 \\ A_4 & 0 & 0 \\ 0 & A_2 & 0 \\ 0 & 0 & A_1 \end{bmatrix},$$

$$\operatorname{rank}\begin{bmatrix} 0 & 0 & (C_2)^*_\eta & 0 & (C_1)^*_\eta & 0 & 0 & (A_4)^*_\eta & 0 & 0 & 0 & 0 & 0 \\ 0 & (C_2)^*_\eta & 0 & (C_3)^*_\eta & 0 & 0 & 0 & 0 & (A_4)^*_\eta & 0 & 0 & 0 & 0 \\ C_1 & 0 & 0 & 0 & 0 & A_5 & 0 & 0 & -C_1(F_7)^*_\eta & 0 & 0 & 0 & 0 \\ C_3 & 0 & 0 & E_3 & 0 & 0 & A_7 & 0 & 0 & 0 & 0 & (F_3)^*_\eta & 0 \\ 0 & (A_6)^*_\eta & 0 & 0 & 0 & 0 & 0 & 0 & 0 & (A_2)^*_\eta & 0 & 0 & 0 \\ 0 & 0 & & (A_6)^*_\eta 0 & 0 & 0 & 0 & 0 & 0 & 0 & (A_2)^*_\eta & 0 & 0 \\ 0 & 0 & 0 & (A_7)^*_\eta & 0 & 0 & 0 & 0 & 0 & 0 & 0 & (A_3)^*_\eta & 0 \\ 0 & 0 & 0 & 0 & (A_5)^*_\eta & 0 & 0 & 0 & 0 & 0 & 0 & 0 & (A_1)^*_\eta \\ A_4 & 0 & 0 & F_7(C_3)^*_\eta & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & A_1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & F_3 & 0 & 0 & A_3 & 0 & 0 & 0 & 0 & 0 & 0 \end{bmatrix}$$

$$=\operatorname{rank}\begin{bmatrix} C_1 & 0 & A_5 & 0 & 0 \\ 0 & C_2 & 0 & A_6 & 0 \\ 0 & C_3 & 0 & 0 & A_7 \\ A_1 & 0 & 0 & 0 & 0 \\ 0 & A_4 & 0 & 0 & 0 \\ 0 & 0 & A_1 & 0 & 0 \\ 0 & 0 & 0 & A_2 & 0 \\ 0 & 0 & 0 & 0 & A_3 \end{bmatrix} + \operatorname{rank}\begin{bmatrix} C_2 & A_6 & 0 & 0 \\ -C_3 & 0 & A_7 & 0 \\ C_1 & 0 & 0 & A_5 \\ A_4 & 0 & 0 & 0 \\ 0 & A_2 & 0 & 0 \\ 0 & 0 & A_3 & 0 \\ 0 & 0 & 0 & A_1 \end{bmatrix},$$

$$\operatorname{rank}\begin{bmatrix} 0 & (C_1)^*_\eta & (C_3)^*_\eta & 0 & 0 & (A_4)^*_\eta & 0 & 0 \\ C_1 & E_1 & 0 & A_5 & 0 & 0 & (F_1)^*_\eta & 0 \\ C_3 & 0 & E_2 & 0 & A_7 & 0 & 0 & F_3 \\ 0 & (A_5)^*_\eta & 0 & 0 & 0 & 0 & (A_1)^*_\eta & 0 \\ 0 & 0 & (A_7)^*_\eta & 0 & 0 & 0 & 0 & (A_3)^*_\eta \\ A_4 & 0 & F_7(C_2)^*_\eta & 0 & 0 & 0 & 0 & 0 \\ 0 & F_1 & 0 & A_1 & 0 & 0 & 0 & 0 \\ 0 & 0 & F_3 & 0 & A_3 & 0 & 0 & 0 \end{bmatrix} = 2\operatorname{rank}\begin{bmatrix} C_3 & A_7 & 0 \\ C_1 & 0 & A_5 \\ A_4 & 0 & 0 \\ 0 & A_3 & 0 \\ 0 & 0 & A_1 \end{bmatrix}.$$

If the above conditions (i)–(iii) are fulfilled, the general solution to the system defined in (2.8) *is* *given by*

$$X_1 = (P_1 + (Q_1)^*_\eta)/2, X_2 = (P_2 + (Q_2)^*_\eta)/2, \quad X_3 = (P_3 + (Q_3)^*_\eta)/2, \quad Z_{11} = (Z_1 + (Z_1)^*_\eta)/2,$$

or $Z = (Z_2 + (Z_2)^*_\eta)/2$, *where*

$$\begin{aligned}
X_1 &= A_1^\dagger F_1 + L_{A_1}[A_8^\dagger(E_4 - C_4 Z_{11}(C_4)^*_\eta) - A_8^\dagger T_1 (A_8^\dagger)^*_\eta + L_{A_8} T_2],\\
X_2 &= A_2^\dagger F_2 + L_{A_2}[A_9^\dagger(E_5 - C_5 Z_{11}(C_5)^*_\eta) - A_9^\dagger T_4 (A_9^\dagger)^*_\eta + L_{A_9} T_5],\\
X_3 &= A_3^\dagger F_3 + L_{A_3}[A_{10}^\dagger(E_6 - C_6 Z_{11}(C_6)^*_\eta) - A_{10}^\dagger T_7 (A_{10}^\dagger)^*_\eta + L_{A_{10}} T_6],\\
Z &= C_4^\dagger D_4 + L_{C_4}(C_4^\dagger D_4)^*_\eta + R_{C_4} Z_{11} (R_{C_4})^*_\eta,\\
Z_{12} &= \psi + L_{A_{11}} L_{A_{14}} Z_1 + Z_2 (R_{A_{14}})^*_\eta (R_{A_{11}})^*_\eta + L_{A_{11}} Z_3 (R_{A_{12}})^*_\eta + L_{A_{12}} Z_4 (R_{A_{11}})^*_\eta, \qquad (5.1)\\
Z_{12} &= Z_b + L_{A_13} Z_5 + Z_6 R_{(A_{13})^*_\eta}, \qquad (5.2)
\end{aligned}$$

with

$$\begin{aligned}
Z_3 &= A^\dagger E (C^\dagger)^*_\eta - A^\dagger C M^\dagger E (C^\dagger)^*_\eta - A^\dagger S C^\dagger E L_{C^*_\eta} N^\dagger A^*_\eta (C^\dagger)^*_\eta - A^\dagger S Z_{10} R_N A^*_\eta (C^\dagger)^*_\eta + L_A Z_{11} + Z_{12} R_{C^*_\eta},\\
Z_4 &= M^\dagger E (A^\dagger)^*_\eta + S^\dagger S C^\dagger E L_{C^*_\eta} N^\dagger (A^\dagger)^*_\eta (C^\dagger)^*_\eta + L_M L_S Z_{13} + L_M Z_{10} R_N + Z_{14} R_{A^*_\eta},\\
[Z_2\, Z_6] &= R_{A_{15}}(E_{11} - C_8 Z_3 (C_9)^*_\eta - C_9 Z_4 (C_8)^*_\eta)(A_{15}^\dagger)^*_\eta + A_{15} A_{15}^\dagger Z_7 + Z_9 R_{(A_{15})^*_\eta},\\
\begin{bmatrix} Z_1 \\ Z_5 \end{bmatrix} &= A_{15}^\dagger (E_{11} - C_8 Z_3 (C_9)^*_\eta - C_9 Z_4 (C_8)^*_\eta) - A_{15}^\dagger Z_7 (A_{15})^*_\eta + L_{A_{15}} Z_8,
\end{aligned}$$

where $T_1, T_2, T_4, \ldots T_7, Z_7, \ldots, Z_9, Z_{10} = (Z_{10})^*_\eta, Z_{12}, \ldots, Z_{15}$ *are quaternionic matrices of suitable dimensions over* $\mathbb{H}$.

Remark 5.1. *Note that* Z_{12} *is defined in two different contexts within Theorem 5.1. In the first definition* (5.1)*,* Z_{12} *is used as an arbitrary matrix to express the general solution, involving projections and generalized inversion operations with matrices* $A_{11}, A_{12}, A_{14}, B_{11}$*, and* B_{14}*. In the second definition stated in* (5.2)*,* Z_{12} *is redefined in terms of a particular solution* Z_b *and of* Z_5, Z_6*, based on the consistency conditions of the systems.*

Proof of Theorem 5.1. The aim of this proof is to establish the equivalence in solvability between systems. Consider the system stated as

$$\begin{aligned}
&A_1 P_1 = F_1, A_2 P_2 = F_2, A_3 P_3 = F_3, A_4 Z_1 = F_7, Z_1 B_4 = F_8, A_5 P_1 + Q_1 (A_5)^*_\eta + C_1 Z_1 (C_1)^*_\eta = E_1,\\
&Z_1 = (Z_1)^*_\eta, A_6 P_2 + Q_2 (A_6)^*_\eta + C_2 Z_1 (C_2)^*_\eta = E_2, A_7 P_3 + Q_3 (A_7)^*_\eta + C_3 Z_1 (C_3)^*_\eta = E_3.
\end{aligned} \qquad (5.3)$$

We aim to prove the relationship between the equations stated in (5.3) and those presented in the system proposed in (2.8). First, we begin with the assumption that the system established in (2.8) has a solution, denoted as (X_1, X_2, X_3, Z) and under this assumption, the set $(P_1, Q_1, P_2, Q_2, P_3, Q_3, Z_1) = (X_1, (X_1)^*_\eta, X_2, (X_2)^*_\eta, X_3, (X_3)^*_\eta, Z)$ is a valid solution for the system given in (5.3). Second, if we have a solution for the system defined in (2.8), we can construct a solution to the system formulated in (5.3). Third, conversely, if the system presented in (5.3) has a solution given by $(P_1, Q_1, P_2, Q_2, P_3, Q_3, Z_1)$, then it immediately follows that the system represented in (2.8) also possesses a solution expressed as $(X_1, X_2, X_3, Z) = ((P_1 + (Q_1)^*_\eta)/2, (P_2 + (Q_2)^*_\eta)/2, (P_3 + (Q_3)^*_\eta)/2, Z_1 + (Z_1)^*_\eta)/2)$. Fourth and lastly, utilizing the insights of Theorem 3.1, we can derive the solvability conditions and deduce a general solution to the system formulated in (5.3). □

5.2. Illustrative example

In this example, we explore the calculation of the i-Hermitian solution for the system described in (2.8), utilizing values selected to meet the necessary rank conditions. This example illustrates the practical application of Theorem 5.1. For those interested in replicating the calculations or exploring the computational details of this example further, the `Python` code used to derive these solutions is provided in Appendix C. The numerical example is as follows. We start by initializing the matrices

$$A_1 = \begin{bmatrix} \mathrm{i}+\mathrm{j} & 2+\mathrm{k} \\ 3 & 7+\mathrm{i} \\ 12\mathrm{j} & \mathrm{i}+\mathrm{k} \end{bmatrix}, A_2 = \begin{bmatrix} \mathrm{i}+\mathrm{j}+\mathrm{k} \\ 2+5\mathrm{j} \\ 3+4\mathrm{i} \end{bmatrix}, A_3 = \begin{bmatrix} 2 & \mathrm{k} \\ \mathrm{i} & \mathrm{j} \\ \mathrm{i}+\mathrm{j} & 6+\mathrm{k} \end{bmatrix}, A_4 = \begin{bmatrix} 2\mathrm{i}+3\mathrm{j} & 7+\mathrm{k} & 6 \\ \mathrm{k}+3 & 7 & \mathrm{i}+\mathrm{j} \end{bmatrix},$$

$$A_5 = \begin{bmatrix} \mathrm{j} & \mathrm{i}+\mathrm{j} \\ \mathrm{k} & 7 \\ \mathrm{i} & 2\mathrm{i}+3\mathrm{j} \end{bmatrix}, A_6 = \begin{bmatrix} \mathrm{i} \\ \mathrm{j}+\mathrm{k} \\ 2\mathrm{i}+7\mathrm{k} \end{bmatrix}, A_7 = \begin{bmatrix} 7 & 6+\mathrm{i} \\ 8+2\mathrm{j} & 9+3\mathrm{k} \\ \mathrm{i}+\mathrm{k} & 2+\mathrm{j} \end{bmatrix},$$

$$B_4 = \begin{bmatrix} 3+\mathrm{i} \\ 2+\mathrm{j}+\mathrm{k} \\ 7 \end{bmatrix}, C_1 = \begin{bmatrix} 1 & 0 & \mathrm{j} \\ \mathrm{i} & 7 & \mathrm{k} \\ \mathrm{k} & \mathrm{j} & \mathrm{i}+\mathrm{j} \end{bmatrix}, C_2 = \begin{bmatrix} 2 & 0 & \mathrm{k} \\ \mathrm{i} & \mathrm{j}+\mathrm{k} & 3 \\ \mathrm{j} & 7 & \mathrm{i}+\mathrm{j} \end{bmatrix}, C_3 = \begin{bmatrix} 2 & 6 & \mathrm{k} \\ \mathrm{i} & 3 & \mathrm{j} \\ \mathrm{k} & 4 & \mathrm{i} \end{bmatrix}.$$

As defined in (2.1), i, j, and k are quaternionic numbers. The process to solve the system based on the matrices $A_1, \ldots, A_7, B_4, C_1, C_2$, and C_3 presented above involves the following steps:

- Step 1 — Verify that all rank conditions are satisfied as required by Theorem 5.1.
- Step 2 — Compute the i-Hermitian solution to the system stated in (2.8) using the provided matrices and Theorem 5.1.
- Step 3 — Derive the solution to the system formulated in (2.8) as

$$X_1 = \begin{bmatrix} 1+2\mathrm{i} & \mathrm{j}+5\mathrm{k} & 9 \\ 2\mathrm{j}+3\mathrm{k} & 7 & 1+\mathrm{j}+\mathrm{k} \end{bmatrix}, X_2 = \begin{bmatrix} \mathrm{i} & \mathrm{j}+\mathrm{k} & 2+\mathrm{i}+\mathrm{j}+\mathrm{k} \end{bmatrix}, X_3 = \begin{bmatrix} 6 & \mathrm{i}+3\mathrm{j} & 7\mathrm{k} \\ \mathrm{i}+\mathrm{j} & 2+7\mathrm{j} & 7\mathrm{i}+9\mathrm{j} \end{bmatrix},$$

$$Z = Z^{\mathrm{i}*} = \begin{bmatrix} 4 & 7\mathrm{i}+\mathrm{j} & 3\mathrm{j}+8\mathrm{k} \\ \mathrm{i}+\mathrm{j}+4\mathrm{k} & 3+\mathrm{j} & 17 \\ \mathrm{i}+\mathrm{k} & 1+2\mathrm{j}+5\mathrm{k} & 5\mathrm{k} \end{bmatrix}.$$

This example illustrates the practical application of Theorem 5.1 to obtain i-Hermitian solutions for a quaternionic matrix system. The steps involve verifying the rank conditions, computing intermediate matrices, and deriving the final solutions. By using similar techniques, the η-Hermitian solution can be obtained based on its i-Hermitian counterpart, showing the generality and robustness of our results in addressing quaternionic systems with Hermitian constraints. The properties and symmetry inherent in the Hermitian solution provide robust tools for solving such systems in broader contexts.

6. Conclusions

Quaternions, with their unique mathematical properties and applications ranging from quantum physics to engineering, have been a rich area of academic inquiry. Our study advanced the field of quaternions by providing a framework to understand quaternionic linear matrix equations, particularly as stated in (2.7).

The proposed algorithm, based on Theorem 3.1, which outlined the solvability of quaternionic linear matrix equations, was designed for efficiency and precision, as shown by a numerical illustration. Our approach has applications in various fields, such as control theory, three-dimensional graphics rendering, and quantum computation. Additionally, the focus on η-Hermitian and i-Hermitian solutions enhances the relevance of this investigation, offering valuable insights in areas like signal processing.

Our investigation bridged the gap between theory and applications by offering an algorithm validated through numerical examples. The robustness of the proposed algorithm across a spectrum of complex matrix systems was illustrated. However, this study has limitations. The primary focus on systems described by the equations presented in (2.7) and (2.8) suggests that there is potential for exploring other types of quaternionic matrix equations. Moreover, the robustness of the algorithm against more intricate systems requires further investigation.

Future research could adapt the proposed algorithm to a broader range of quaternionic matrix equations and explore iterative procedures for solving such equations. Additionally, under the perspective of high data volume [56], more efficient computational algorithms [57] should be employed. Given its usage in three-dimensional rotations, our methodology should also be explored in robotics [58, 59].

In summary, this research enhanced the understanding of quaternionic linear matrix equations and provided practical computational and algorithmic methods, encouraging further scholarly and applied exploration in this field.

Author contributions

Conceptualization: A.R., M.Z.U.R., A.G., C.M-B., C.C., V.L., X.C.; Data curation: A.R., C.C.; Formal analysis: A.R., M.Z.U.R., A.G., C.M-B., C.C., V.L., X.C.; Investigation: A.R., M.Z.U.R., A.G., C.C., V.L.: Methodology: A.R., M.Z.U.R., A.G., C.M-B., C.C., V.L.; Writing —original draft: A.R., M.Z.U.R., A.G., C.M-B., X.C.; Writing —review and editing: V.L., C.C. All authors have read and agreed to the published version of the article.

Acknowledgments

This research was partially supported by the Vice-rectorate for Research, Creation, and Innovation (VINCI) of the Pontificia Universidad Católica de Valparaíso (PUCV), Chile, grants VINCI 039.470/2024 (regular research), VINCI 039.493/2024 (interdisciplinary associative research), VINCI 039.309/2024 (PUCV centenary), and FONDECYT 1200525 (Víctor Leiva), from the National Agency for Research and Development (ANID) of the Chilean government under the Ministry of Science, Technology, Knowledge, and Innovation; and by Portuguese funds through the CMAT—Research Centre of Mathematics of University of Minho, Portugal, within projects UIDB/00013/2020 (https://doi.org/10.54499/UIDB/00013/2020) and UIDP/00013/2020 (https://doi.org/10.54499/UIDP/00013/2020) (Cecilia Castro).

The authors would also like to thank the Editors and reviewers for their constructive comments, which led to improvements in the presentation of the article.

Conflict of interest

The authors declare no conflicts of interest.

References

1. A. Rehman, Q. W. Wang, A system of matrix equations with five variables, *Appl. Math. Comput.*, **271** (2015), 805–819. https://doi.org/10.1016/j.amc.2015.09.066
2. W. R. Hamilton, LXXVIII. On quaternions; or on a new system of imaginaries in Algebra: to the editors of the Philosophical Magazine and Journal, *London Edinburgh Dublin Philos. Mag. J. Sci.*, **25** (1844), 489–495. https://doi.org/10.1080/14786444408645047
3. S. L. Adler, *Quaternionic quantum mechanics and quantum fields*, New York, USA: Oxford University Press, 1995.
4. S. D. Leo, G. Scolarici, Right eigenvalue equation in quaternionic quantum mechanics, *J. Phys. A: Math. Gen.*, **33** (2000), 2971–2995. https://doi.org/10.1088/0305-4470/33/15/306
5. C. C. Took, D. P. Mandic, Augmented second-order statistics of quaternion random signals, *Signal Process.*, **91** (2011), 214–224. https://doi.org/10.1016/j.sigpro.2010.06.024
6. J. B. Kuipers, *Quaternions and rotation sequences: a primer with applications to orbits, aerospace, and virtual reality*, Princeton: Princeton University Press, 1999. https://doi.org/10.2307/j.ctvx5wc3k
7. C. C. Took, D. P. Mandic, The quaternion LMS algorithm for adaptive filtering of hypercomplex processes, *IEEE Trans. Signal Proces.*, **57** (2009), 1316–1327. https://doi.org/10.1109/TSP.2008.2010600
8. C. C. Took, D. P. Mandic, A quaternion widely linear adaptive filter, *IEEE Trans. Signal Proces.*, **58** (2010), 4427–4431. https://doi.org/10.1109/TSP.2010.2048323
9. C. C. Took, D. P. Mandic, F. Zhang, On the unitary diagonalisation of a special class of quaternion matrices, *Appl. Math. Lett.*, **24** (2011), 1806–1809. https://doi.org/10.1016/j.aml.2011.04.038
10. A. Rehman, Q. W. Wang, I. Ali, M. Akram, M. O. Ahmad, A constraint system of generalized Sylvester quaternion matrix equations, *Adv. Appl. Clifford Algebras*, **27** (2017), 3183–3196. https://doi.org/10.1007/s00006-017-0803-1
11. A. Rehman, I. Kyrchei, I. Ali, M. Akram, A. Shakoor, Constraint solution of a classical system of quaternion matrix equations and its Cramer's rule, *Iran. J. Sci. Technol. Trans. A: Sci.*, **45** (2021), 1015–1024. https://doi.org/10.1007/s40995-021-01083-7
12. R. K. Cavin, S. P. Bhattacharyya, Robust and well-conditioned eigenstructure assignment via Sylvester's equation, *Optimal Control Appl. Methods*, **4** (1983), 205–212. https://doi.org/10.1002/oca.4660040302
13. V. L. Syrmos, F. L. Lewis, Coupled and constrained Sylvester equations in system design, *Circuits, System Signal Process.*, **13** (1994), 663–694. https://doi.org/10.1007/BF02523122
14. M. Darouach, Solution to Sylvester equation associated to linear descriptor systems, *Syst. Control Lett.*, **55** (2006), 835–838. https://doi.org/10.1016/j.sysconle.2006.04.004

15. A. Rehman, I. I. Kyrchei, Solving and algorithm to system of quaternion Sylvester-type matrix equations with $*$-hermicity, *Adv. Appl. Clifford Algebras*, **32** (2022), 49. https://doi.org/10.1007/s00006-022-01222-2

16. S. Gupta, Linear quaternion equations with application to spacecraft attitude propagation, *1998 IEEE Aerospace Conference Proceedings*, Snowmass at Aspen, CO, **1** (1998), 69–76. https://doi.org/10.1109/AERO.1998.686806

17. V. N. Kovalnogov, R. V. Fedorov, D. A. Demidov, M. A. Malyoshina1, T. E. Simos, V. N. Katsikis, et al., Zeroing neural networks for computing quaternion linear matrix equation with application to color restoration of images, *AIMS Math.*, **8** (2023), 14321–14339. https://doi.org/10.3934/math.2023733

18. A. Shakoor, I. Ali, S. Wali, A. Rehman, Some formulas on the Drazin inverse for the sum of two matrices and block matrices, *Bull. Iran. Math. Soc.*, **48** (2022), 351–366. https://doi.org/10.1007/s41980-020-00521-3

19. D. Zhang, T. Jiang, G. Wang, V. I. Vasilev, On singular value decomposition and generalized inverse of a commutative quaternion matrix and applications, *Appl. Math. Comput.*, **460** (2024), 128291. https://doi.org/10.1016/j.amc.2023.128291

20. V. N. Kovalnogov, R. V. Fedorov, I. I. Shepelev, V. V. Sherkunov, T. E. Simos, S. D. Mourtas, et al., A novel quaternion linear matrix equation solver through zeroing neural networks with applications to acoustic source tracking, *AIMS Math.*, **8** (2023), 25966–25989. https://doi.org/10.3934/math.20231323

21. X. Liu, The η-anti-Hermitian solution to some classic matrix equations, *Appl. Math. Comput.*, **320** (2018), 264–270. https://doi.org/10.1016/j.amc.2017.09.033

22. A. Rehman, Q. W. Wang, Z. H. He, Solution to a system of real quaternion matrix equations encompassing η-Hermicity, *Appl. Math. Comput.*, **265** (2015), 945–957. https://doi.org/10.1016/j.amc.2015.05.104

23. A. Rehman, I. Kyrchei, M. Z. U. Rahman, V. Leiva, C. Castro, Solvability and algorithm for Sylvester-type quaternion matrix equations with potential applications, *AIMS Math.*, **9** (2024), 19967–19996. https://doi.org/10.3934/math.2024974

24. R. T. Farouki, *Pythagorean-hodograph curves: algebra and geometry inseparable*, New York: Springer, 2008.

25. A. Altavilla, C. de Fabritiis, Equivalence of slice semi-regular functions via Sylvester operators, *Linear Algebra Appl.*, **607** (2020), 151–189. https://doi.org/10.1016/j.laa.2020.08.009

26. J. A. Díaz-García, V. Leiva, M. Galea, Singular elliptic distribution: density and applications, *Commun. Stat.: Theory Methods*, **31** (2002), 665–681. https://doi.org/10.1081/STA-120003646

27. A. Barraud, S. Lesecq, N. Christov, From sensitivity analysis to random floating point arithmetics-application to Sylvester equations, *Numer. Anal. Appl.*, **1988** (2001), 35–41. https://doi.org/10.1007/3-540-45262-1_5

28. R. C. Li, A bound on the solution to a structured Sylvester equation with an application to relative perturbation theory, *SIAM J. Matrix Anal. Appl.*, **21** (1999), 440–445. https://doi.org/10.1137/S0895479898349586

29. V. L. Syrmos, F. L. Lewis, Output feedback eigenstructure assignment using two Sylvester equations, *IEEE Trans. Autom. Control*, **38** (1993), 495–499. https://doi.org/10.1109/9.210155

30. Y. N. Zhang, D. C. Jiang, J. Wang, A recurrent neural network for solving Sylvester equation with time-varying coefficients, *IEEE Trans. Neural Networks*, **13** (2002), 1053–1063. https://doi.org/10.1109/TNN.2002.1031938

31. Z. Z. Bai, On Hermitian and skew-Hermitian splitting iteration methods for continuous Sylvester equations, *J. Comput. Math.*, **29** (2011), 185–198.

32. L. Rodman, *Topics in quaternion linear algebra*, Princeton: Princeton University Press, 2014.

33. W. E. Roth, The equations $AX - YB = C$ and $AX - XB = C$ in matrices, *Proc. Amer. Math. Soc.*, **3** (1952), 392–396. https://doi.org/10.2307/2031890

34. J. K. Baksalary, R. Kala, The matrix equation $AX - YB = C$, *Linear Algebra Appl.*, **25** (1979), 41–43. https://doi.org/10.1016/0024-3795(79)90004-1

35. L. Wang, Q. Wang, Z. He, The common solution of some matrix equations, *Algebra Colloq.*, **23** (2016), 71–81. https://doi.org/10.1142/S1005386716000092

36. Q. W. Wang, Z. H. He, Solvability conditions and general solution for the mixed Sylvester equations, *Automatica*, **49** (2013), 2713–2719. https://doi.org/10.1016/j.automatica.2013.06.009

37. S. G. Lee, Q. P. Vu, Simultaneous solutions of matrix equations and simultaneous equivalence of matrices, *Linear Algebra Appl.*, **437** (2012), 2325–2339. https://doi.org/10.1016/j.laa.2012.06.004

38. Y. Q. Lin, Y. M. Wei, Condition numbers of the generalized Sylvester equation, *IEEE Trans. Autom. Control*, **52** (2007), 2380–2385. https://doi.org/10.1109/TAC.2007.910727

39. Q. W. Wang, A. Rehman, Z. H. He, Y. Zhang, Constraint generalized Sylvester matrix equations, *Automatica*, **69** (2016), 60–64. https://doi.org/10.1016/j.automatica.2016.02.024

40. X. Zhang, A system of generalized Sylvester quaternion matrix equations and its applications, *Appl. Math. Comput.*, **273** (2016), 74–81. https://doi.org/10.1016/j.amc.2015.09.074

41. A. Dmytryshyn, B. Kåström, Coupled Sylvester-type matrix equations and block diagonalization, *SIAM J. Matrix Anal. Appl.*, **38** (2015), 580–593. https://doi.org/10.1137/151005907

42. F. O. Farid, Z. H. He, Q. W. Wang, The consistency and the exact solutions to a system of matrix equations, *Linear Multilinear Algebra*, **64** (2016), 2133–2158. https://doi.org/10.1080/03081087.2016.1140717

43. Z. H. He, M. Wang, X. Liu, On the general solutions to some systems of quaternion matrix equations, *RACSAM*, **114** (2020), 95.

44. H. K. Wimmer, Consistency of a pair of generalized Sylvester equations, *IEEE Trans. Autom. Control*, **39** (1994), 1014–1016. https://doi.org/10.1109/9.284883

45. B. Kågström, A perturbation analysis of the generalized Sylvester equation $(AR - LB, DR - LE) = (C, F)$, *SIAM J. Matrix Anal. Appl.*, **15** (1994), 1045–1060. https://doi.org/10.1137/S0895479893246212

46. Z. H. He, Q. W. Wang, A pair of mixed generalized Sylvester matrix equations, *J. Shanghai Univ. Nat. Sci. Ed.*, **20** (2014), 138–156. https://doi.org/10.3969/j.issn.1007-2861.2014.01.021

47. Q. W. Wang, Z. H. He, Systems of coupled generalized Sylvester matrix equations, *Automatica*, **50** (2014), 2840–2844. https://doi.org/10.1016/j.automatica.2014.10.033

48. Z. H. He, Q. W. Wang, A system of periodic discrete-time coupled Sylvester quaternion matrix equations, *Algebra Colloq.*, **24** (2017), 169–180. https://doi.org/10.1142/S1005386717000104

49. R. Kristiansen, P. J. Nicklasson, Satellite attitude control by quaternion-based backstepping, *Proceedings of the 2005 American Control Conference*, 2005, 16–18.

50. D. Finkelstein, J. M. Jauch, S. Schiminovich, D. Speiser, Foundations of quaternion quantum mechanics, *J. Math. Phys.*, **3** (1962), 207–220.

51. S. C. Pei, Y. Z. Hsiao, Colour image edge detection using quaternion quantized localized phase, *Proc. 18th European Signal Processing Conference*, 2010, 1766–1770.

52. G. Marsaglia, G. P. H. Styan, Equalities and inequalities for ranks of matrices, *Linear Multilinear Algebra*, **2** (1974), 269–292. https://doi.org/10.1080/03081087408817070

53. J. N. Buxton, R. F. Churchouse, A. B. Tayler, *Matrices methods and applications*, Oxford, UK: Clarendon Press, 1990.

54. Q. W. Wang, Z. C. Wu, C. Y. Lin, Extremal ranks of a quaternion matrix expression subject to consistent systems of quaternion matrix equations with applications, *Appl. Math. Comput.*, **182** (2006), 1755–1764. https://doi.org/10.1016/j.amc.2006.06.012

55. Z. H. He, Q. W. Wang, The general solutions to some systems of matrix equations, *Linear Multilinear Algebra*, **63** (2015), 2017–2032. https://doi.org/10.1080/03081087.2014.896361

56. R. G. Aykroyd, V. Leiva, F. Ruggeri, Recent developments of control charts, identification of big data sources and future trends of current research, *Technol. Forecast. Soc. Change*, **144** (2019), 221–232. https://doi.org/10.1016/j.techfore.2019.01.005

57. J. A. Ramirez-Figueroa, C. Martin-Barreiro, A. B. Nieto, V. Leiva, M. P. Galindo-Villardón, A new principal component analysis by particle swarm optimization with an environmental application for data science. *Stoch. Environ. Res. Risk Assess.*, **35** (2021), 1969–1984. https://doi.org/10.1007/s00477-020-01961-3

58. A. Ghaffar, M. Z. U. Rahman, V. Leiva, C. Martin-Barreiro, X. Cabezas, C. Castro, Efficiency, optimality, and selection in a rigid actuation system with matching capabilities for an assistive robotic exoskeleton. *Eng. Sci. Technol., Int. J.*, **51** (2024), 101613. https://doi.org/10.1016/j.jestch.2023.101613

59. A. Ghaffar, M. Z. U. Rahman, V. Leiva, C. Castro, Optimized design and analysis of cable-based parallel manipulators for enhanced subsea operations, *Ocean Eng.*, **297** (2024), 117012. https://doi.org/10.1016/j.oceaneng.2024.117012

Appendix

A. Python code for verification

The following Python code was used to verify the results of the example discussed in Subsection 2.3. The code calculates the Moore-Penrose inverse of a quaternionic matrix and performs other related computations.

Listing 1. Python code for quaternionic matrix calculations.

```
import numpy as np
import quaternion

# Defining a function to calculate the product of two quaternion matrices
def quaternion_matrix_multiply(A, B):
assert A.shape[1] == B.shape[0], "Incompatible dimensions for multiplication"
result = np.zeros((A.shape[0], B.shape[1]), dtype=np.quaternion)
for i in range(A.shape[0]):
for j in range(B.shape[1]):
for k in range(A.shape[1]):
result[i, j] += A[i, k] * B[k, j]
return result

# Stating a function to calculate the inverse of a 2x2 quaternion matrix
def quaternion_matrix_inverse_2x2(A):
assert A.shape == (2, 2), "Inverse is implemented only for 2x2 matrices"
det = A[0, 0] * A[1, 1] - A[0, 1] * A[1, 0]
inv_det = 1 / det
A_inv = np.array([[A[1, 1], -A[0, 1]], [-A[1, 0], A[0, 0]]], dtype=np.quaternion)
return inv_det * A_inv

# Obtaining the quaternion matrix A
A = np.array([
[np.quaternion(1, 1, 0, 0), np.quaternion(2, 0, 1, 0)],
[np.quaternion(2, 0, 1, 0), np.quaternion(4, 0, 0, 1)]
])

# Computing the usual transpose of A
A_T = A.T

# Calculating the product A_T * A by using the custom function
A_T_A = quaternion_matrix_multiply(A_T, A)

# Printing the computed terms of the product
print("A_T * A:")
print(A_T_A)

# Generating the inverse of A_T * A by utilizing the custom function
A_T_A_inv = quaternion_matrix_inverse_2x2(A_T_A)

# Displaying the computed inverse
print("Inverse (A_T * A)^-1:")
print(A_T_A_inv)

# Determining the Moore-Penrose inverse of A
A_dagger = quaternion_matrix_multiply(A_T_A_inv, A_T)

# Providing the computed Moore-Penrose inverse
print("Moore-Penrose inverse A^dagger:")
```

```
print(A_dagger)
```

B. Python code for numerical example

The following Python code implements Algorithm 1. The code takes as input the matrices defined in the numerical example in Subsection 4.2 and computes the solutions step-by-step, as outlined in Theorem 3.1.

Listing 2. Python code for the numerical example.

```
import numpy as np
import quaternion

# Defining a function to multiply two quaternionic matrices
def quaternion_matrix_multiply(A, B):
assert A.shape[1] == B.shape[0], "Incompatible dimensions for multiplication"
result = np.zeros((A.shape[0], B.shape[1]), dtype=np.quaternion)
for i in range(A.shape[0]):
for j in range(B.shape[1]):
for k in range(A.shape[1]):
result[i, j] += A[i, k] * B[k, j]
return result

# Stating a function to manually calculate the Moore-Penrose inverse for 2x2 quaternionic matrices
def quaternion_matrix_pinv_2x2(A):
assert A.shape == (2, 2), "Moore-Penrose inverse implemented only for 2x2 matrices"
A_adj = np.array([
[A[1, 1], -A[0, 1]],
[-A[1, 0], A[0, 0]]
], dtype=np.quaternion)
A_dagger = A_adj / (A[0, 0] * A[1, 1] - A[0, 1] * A[1, 0])
return A_dagger

# Establishing a function to compute the Moore-Penrose inverse for matrices by using the 2x2 case as implemented
def quaternion_matrix_pinv(A):
if A.shape == (2, 2):
return quaternion_matrix_pinv_2x2(A)
else:
raise NotImplementedError("Moore-Penrose inverse for matrices greater than 2x2 needs to be implemented")

# Initializing quaternionic matrices
A1 = np.array([
[np.quaternion(1, 0, 0, 0), np.quaternion(0, 12, 1, 0), np.quaternion(0, 0, 1, 0)],
[np.quaternion(5, 1, 0, 0), np.quaternion(0, 1, 1, 0), np.quaternion(2, 1, 0, 0)]
], dtype=np.quaternion)

B1 = np.array([
[np.quaternion(0, 4, 0, 0), np.quaternion(2, 0, 1, 0), np.quaternion(7, 1, 1, 0)],
[np.quaternion(1, 0, 1, 0), np.quaternion(0, 2, 0, 0), np.quaternion(1, 1, 0, 0)]
], dtype=np.quaternion)

A2 = np.array([
[np.quaternion(2, 0, 1, 0), np.quaternion(0, 1, 0, 0)],
[np.quaternion(1, 1, 0, 0), np.quaternion(0, 1, 1, 0)],
[np.quaternion(1, 1, 0, 0), np.quaternion(7, 0, 1, 0)]
```

```
], dtype=np.quaternion)

B2 = np.array([
[np.quaternion(1, 5, 0, 0), np.quaternion(1, 1, 0, 0)],
[np.quaternion(0, 1, 8, 0), np.quaternion(2, 0, 1, 0)]
], dtype=np.quaternion)

A5 = np.array([
[np.quaternion(1, 1, 0, 0), np.quaternion(0, 1, 1, 0), np.quaternion(5, 7, 0, 0)],
[np.quaternion(2, 5, 0, 0), np.quaternion(7, 1, 0, 0), np.quaternion(1, 1, 0, 0)]
], dtype=np.quaternion)

B5 = np.array([
[np.quaternion(1, 0, 1, 0), np.quaternion(2, 0, 1, 0), np.quaternion(7, 0, 0, 1)],
[np.quaternion(0, 1, 0, 1), np.quaternion(1, 0, 1, 0), np.quaternion(2, 0, 0, 1)]
], dtype=np.quaternion)

C2 = np.array([
[np.quaternion(1, 1, 0, 0), np.quaternion(0, -1, 1, 0), np.quaternion(5, 0, 0, 0)],
[np.quaternion(0, 1, -9, 0), np.quaternion(1, 1, 1, 0), np.quaternion(7, 0, 1, 0)]
], dtype=np.quaternion)

D2 = np.array([
[np.quaternion(1, 1, 0, 0), np.quaternion(0, 1, 1, 0), np.quaternion(15, 0, 0, 0)],
[np.quaternion(7, 0, 7, 0), np.quaternion(3, 7, 0, 0), np.quaternion(1, 1, 0, 0)]
], dtype=np.quaternion)

## Part 1: Calculating the first auxiliary matrices
A8 = quaternion_matrix_multiply(A2[:2, :2], quaternion_matrix_pinv(A1[:2, :2]))
 # A_8 = A_2 L_{A_1}
B8 = quaternion_matrix_multiply(quaternion_matrix_pinv(B1[:2, :2]), B5[:2, :2])
 # B_8 = R_{B_1} B_5

## Part 2: Computing the next auxiliary matrices
F1 = quaternion_matrix_multiply(quaternion_matrix_pinv(A1[:2, :2]), A2[:2, :2])
 # F_1 = A_1^\dagger E_1
F4 = quaternion_matrix_multiply(B1_pinv, B5[:2, :2])
 # F_4 = E_4 B_1^\dagger

A9 = quaternion_matrix_multiply(A2[:2, :2], quaternion_matrix_pinv(A1[:2, :2]))
 # A_9 = A_2 L_{A_1}
B9 = quaternion_matrix_multiply(B5[:2, :2], B1_pinv)
 # B_9 = B_2 R_{B_1}

## Part 3: Obtaining the remaining auxiliary matrices
C5 = quaternion_matrix_multiply(C2[:, :2], quaternion_matrix_pinv(A5[:2, :2]))
 # C_5 = C_2 L_{A_5}
D5 = quaternion_matrix_multiply(quaternion_matrix_pinv(B5[:2, :2]), D2[:, :2])
 # D_5 = R_{B_5} D_2

F2 = quaternion_matrix_multiply(quaternion_matrix_pinv(A2[:2, :2]), A5[:2, :2])
 # F_2 = A_2^\dagger E_2
F5 = quaternion_matrix_multiply(A5[:2, :2], quaternion_matrix_pinv(B2[:2, :2]))
 # F_5 = E_5 B_2^\dagger

# Printing results for final verification
print("A8 =", A8)
print("B8 =", B8)
print("F1 =", F1)
```

```
print("F4 =", F4)
print("A9 =", A9)
print("B9 =", B9)
print("C5 =", C5)
print("D5 =", D5)
print("F2 =", F2)
print("F5 =", F5)
```

This script computes all the necessary matrices and variables, as described in Theorem 3.1 and applied to the numerical example. Each function is commented to ensure clarity and guide the reader through the computational process.

C. *Python code for Hermitian solution example*

The following `Python` code calculates the solutions X_1, X_2, X_3, and Z as described in the example of Subsection 5.2.

Listing 3. Python code for Hermitian solution example.

```
import numpy as np
import quaternion

# Providing matrices
A1 = np.array([
[np.quaternion(0, 1, 1, 0), np.quaternion(2, 0, 0, 1)],
[np.quaternion(3, 0, 0, 0), np.quaternion(7, 1, 0, 0)],
[np.quaternion(0, 12, 0, 0), np.quaternion(0, 1, 0, 1)]
], dtype=np.quaternion)

A2 = np.array([
[np.quaternion(0, 1, 1, 1)],
[np.quaternion(2, 0, 5, 0)],
[np.quaternion(3, 4, 0, 1)]
], dtype=np.quaternion)

A3 = np.array([
[np.quaternion(2, 0, 0, 0), np.quaternion(0, 0, 0, 1)],
[np.quaternion(0, 1, 0, 0), np.quaternion(0, 0, 1, 0)],
[np.quaternion(0, 1, 1, 0), np.quaternion(6, 0, 0, 1)]
], dtype=np.quaternion)

# Calculating Z
Z = np.array([
[np.quaternion(4, 0, 7, 3), np.quaternion(0, 7, 1, 8), np.quaternion(3, 0, 8, 0)],
[np.quaternion(0, 1, 1, 4), np.quaternion(3, 0, 1, 0), np.quaternion(17, 0, 0, 0)],
[np.quaternion(0, 1, 0, 1), np.quaternion(0, 2, 1, 5), np.quaternion(5, 0, 0, 0)]
], dtype=np.quaternion)

# Computing solutions X1, X2, X3
X1 = np.array([
[np.quaternion(1, 2, 0, 0), np.quaternion(0, 0, 1, 5), np.quaternion(9, 0, 0, 0)],
[np.quaternion(0, 2, 3, 0), np.quaternion(7, 0, 0, 0), np.quaternion(1, 0, 1, 1)]
], dtype=np.quaternion)

X2 = np.array([
[np.quaternion(0, 1, 0, 0), np.quaternion(0, 0, 1, 1), np.quaternion(2, 1, 1, 1)]
], dtype=np.quaternion)

```

```
X3 = np.array([
[np.quaternion(6, 0, 0, 0), np.quaternion(0, 1, 3, 0), np.quaternion(0, 0, 0, 7)],
[np.quaternion(0, 1, 1, 0), np.quaternion(2, 0, 7, 0), np.quaternion(0, 7, 9, 0)]
], dtype=np.quaternion)

# Printing the solutions
print("X1 =", X1)
print("X2 =", X2)
print("X3 =", X3)
print("Z =", Z)
```

The script above calculates the matrices X_1, X_2, X_3, and Z as presented in the example of Subsection 5.2. Each matrix is constructed and manipulated using quaternionic arithmetic to obtain the solutions as specified in the example.