Efficient Computational Framework for Matrix Equations over n-Generalized Quaternions

Lei Shi[1][0009-0000-4512-5710] and Yang Zhang[2][0000-0002-0540-0893]

[1] Department of Mathematics, University of Manitoba, Winnipeg, MB, Canada
shil4@umymanitoba.ca
[2] Department of Mathematics, University of Manitoba, Winnipeg, MB, Canada
yang.zhang@umanitoba.ca

Abstract. The algebra of n-generalized quaternions provides an effective algebraic framework for representing hierarchical multi-dimensional data. However, solving large-scale matrix equations within this algebra often suffers from severe dimensional expansion. This paper extends the theory of n-reduced quaternions to a more general class of n-generalized systems, characterized by arbitrary positive real parameters. By exploiting a Toeplitz-type isomorphism, we transform matrix equations over n-generalized quaternions into equivalent complex linear systems. Through an iterative layer-by-layer solver, the proposed framework avoids the explicit computation of the computationally expensive global Moore-Penrose inverse. Consequently, the arithmetic cost is reduced from $\mathcal{O}(64m^3n^3)$ to $\mathcal{O}(2m^3n^3)$, where m is the matrix dimension and n is the nilpotent order. Preliminary results demonstrate high numerical stability and significant memory savings, confirming the framework's practical value for high-dimensional mathematical software applications.

Keywords: n-generalized quaternions · Matrix equations · Block forward substitution · Computational complexity.

1 Introduction

Modern scientific computing, multi-channel signal processing, and cryptographic applications increasingly rely on hypercomplex algebras to model inter-channel correlations in multi-dimensional data [13]. Such algebraic structures serve as standard tools for color video analysis [12, 16] and rigid body kinematics [4]. Recent developments extend classical quaternions [18] to encompass sparse representation [19] and unified high-dimensional modeling [5, 9].

Within these frameworks, solving large-scale linear matrix equations (e.g., Sylvester and Lyapunov-type equations $\sum_{k=1}^{p} A_k X B_k = C$) is a ubiquitous and computationally demanding task [1, 2, 17]. Building upon the foundational matrix theory over generalized quaternion algebras [3, 11], recent developments in high-order tensors [6, 14] and reduced biquaternions [10, 20] have extended this modeling capacity to hierarchical multi-layer structures. In this work, we further extend these hierarchical structures to the algebra of n-generalized quaternions ($\mathbb{H}_{ng}$), allowing for arbitrary positive basis parameters α and β. This establishes a highly adaptable structure bridging classical quaternions and split-quaternions.

However, computing solutions to these high-order equations poses severe computational scaling challenges. Although accelerating large-scale matrix computations via Krylov subspace methods or Lanczos bidiagonalization is standard practice in classical and quaternion domains [7, 8, 15], extending these iterative techniques to strictly non-commutative n-generalized structures remains computationally formidable. For a system with nilpotent order n, traditional real-representation methods inflate the equivalent global matrix size to $4mn \times 4mn$, rendering standard computations intractable.

To bypass this structural expansion, this paper proposes an efficient decoupling framework accompanied by a block forward substitution algorithm. By establishing exact solvability conditions via structural isomorphism, the proposed solver demonstrates the computational superiority of the solver, paving the way for its integration into mathematical software packages.

2 The Algebra of n-Generalized Quaternions

Let $\mathbb{H}_g$ be the algebra of generalized quaternions over the real field $\mathbb{R}$ with the basis $\{1, i, j, k\}$ satisfying the multiplication rules:

$$i^2 = -\alpha, \quad j^2 = -\beta, \quad k^2 = -\alpha\beta, \quad ij = -ji = k,$$

where $\alpha, \beta > 0$. The n-generalized quaternions $\mathbb{H}_{ng}$ are defined as polynomials of a nilpotent variable ε:

$$\mathbb{H}_{ng} = \left\{ q = \sum_{l=0}^{n-1} q_l \varepsilon^l \;\middle|\; q_l \in \mathbb{H}_g, \varepsilon^n = 0 \right\},$$

where ε is a central nilpotent generator of index n (i.e., $\varepsilon^n = 0$, $\varepsilon^{n-1} \neq 0$, and $\varepsilon^0 = 1$). For any $p, q \in \mathbb{H}_{ng}$ and $\lambda \in \mathbb{R}$, the algebraic operations are defined as:

$$\lambda p + q = \sum_{l=0}^{n-1} (\lambda p_l + q_l)\varepsilon^l, \quad pq = \sum_{l=0}^{n-1} \left(\sum_{m=0}^{l} p_m q_{l-m} \right) \varepsilon^l.$$

To formulate the Moore-Penrose inverse over $\mathbb{H}_{ng}$, we introduce the principal conjugate. For any $q = \sum_{l=0}^{n-1} q_l \varepsilon^l \in \mathbb{H}_{ng}$ with base components $q_l = a_{l,0} + a_{l,1}i + a_{l,2}j + a_{l,3}k$, its principal conjugate is defined component-wise as:

$$q^* = \sum_{l=0}^{n-1} (a_{l,0} - a_{l,1}i - a_{l,2}j - a_{l,3}k)\varepsilon^l.$$

3 Main Theorems: Isomorphism and Solvability

We consider the generalized Sylvester-type matrix equation over $\mathbb{H}_{ng}$:

$$\sum_{k=1}^{p} A_k X B_k = C, \tag{1}$$

where A_k, B_k, and C are known coefficient matrices and X is the unknown matrix to be determined.

To address the computational challenges posed by non-commutativity, we construct an isomorphic mapping from the hypercomplex system into the complex field.

Definition 1. *For any n-generalized quaternion $q = \sum_{l=0}^{n-1} q_l \varepsilon^l \in \mathbb{H}_{ng}$, the Toeplitz-type isomorphism mapping $\Phi : \mathbb{H}_{ng} \to \mathbb{C}^{2n \times 2n}$ is defined as a block lower-triangular matrix:*

$$\Phi(q) = \begin{bmatrix} \psi(q_0) & \mathbf{0} & \cdots & \mathbf{0} \\ \psi(q_1) & \psi(q_0) & \cdots & \mathbf{0} \\ \vdots & \vdots & \ddots & \vdots \\ \psi(q_{n-1}) & \psi(q_{n-2}) & \cdots & \psi(q_0) \end{bmatrix},$$

where $\psi(q_l) \in \mathbb{C}^{2\times 2}$ is the complex representation of the generalized quaternion $q_l = a_0 + a_1 i + a_2 j + a_3 k \in \mathbb{H}_g$. While the generalized quaternion algebra admits distinct complex embeddings depending on the signs of α and β (e.g., split-quaternions), we focus on the definite case $\alpha, \beta > 0$ to establish the algorithmic framework. Using the imaginary unit $\mathbf{i}$ (with $\mathbf{i}^2 = -1$), the mapping ψ is defined as:

$$\psi(q_l) = \begin{bmatrix} a_0 + a_1\sqrt{\alpha}\mathbf{i} & a_2 + a_3\sqrt{\alpha}\mathbf{i} \\ -\beta(a_2 - a_3\sqrt{\alpha}\mathbf{i}) & a_0 - a_1\sqrt{\alpha}\mathbf{i} \end{bmatrix}.$$

This mapping naturally extends to matrices $A \in \mathbb{H}_{ng}^{m\times m}$ by replacing each scalar element with its corresponding $2n \times 2n$ complex block.

Theorem 1. *The mapping Φ is an injective algebra homomorphism. Specifically, for any $A, B \in \mathbb{H}_{ng}^{m\times m}$, it strictly preserves operations, yielding $\Phi(A+B) = \Phi(A) + \Phi(B)$ and $\Phi(AB) = \Phi(A)\Phi(B)$, with $\Phi(A) = \mathbf{0} \iff A = \mathbf{0}$.*

Proof. The proof follows directly by applying the multiplication rules of the nilpotent unit ε and the complex embeddings of generalized quaternions. Omitted here for brevity.

Definition 2. *To quantify numerical accuracy, we define the isomorphic Frobenius norm via the mapping Φ. For any matrix $A \in \mathbb{H}_{ng}^{m\times m}$, its norm is defined as:*

$$\|A\|_{\mathbb{H}_{ng}} := \|\Phi(A)\|_F = \sqrt{\operatorname{tr}\left(\Phi(A)^H \Phi(A)\right)},$$

where $\|\cdot\|_F$ denotes the standard Frobenius norm in the complex matrix space $\mathbb{C}^{2nm\times 2nm}$. This induced norm allows us to evaluate residual errors directly within the complex domain.

By applying Φ, the hypercomplex matrix equation (1) is transformed into an equivalent complex block-Toeplitz system.

Theorem 2. *The equation (1) over $\mathbb{H}_{ng}$ has a solution if and only if the corresponding complex matrix equation $\sum_{k=1}^{p} \Phi(A_k) Y \Phi(B_k) = \Phi(C)$ is consistent for some $Y \in \mathbb{C}^{2nm\times 2nm}$. Furthermore, the unique minimum-norm least-squares solution X can be recovered from this complex system.*

Proof. The equivalence is guaranteed by Theorem 1 and standard matrix vectorization techniques. Detailed derivations are omitted.

In the context of hypercomplex algebras, a square matrix is considered non-singular if it possesses a unique exact two-sided inverse. The block-triangular structure of the Toeplitz isomorphism Φ yields a straightforward criterion regarding the non-singularity of these high-order matrices.

Lemma 1. *Let $A = \sum_{l=0}^{n-1} A^{(l)}\varepsilon^l \in \mathbb{H}_{ng}^{m\times m}$. The hypercomplex matrix A is non-singular if and only if its base-layer matrix $A^{(0)} \in \mathbb{H}_{g}^{m\times m}$ is non-singular.*

Proof. By Definition 1, the isomorphic matrix $\Phi(A)$ is block lower-triangular, with $\Phi(A^{(0)})$ repeating n times on the main block diagonal. Therefore, $\det(\Phi(A)) = (\det(\Phi(A^{(0)})))^n$. Thus, $\Phi(A)$ is invertible if and only if its base layer is invertible, completely decoupling the singularity condition.

4 Efficient Block Forward Substitution Algorithm

Before presenting the decoupling algorithm, we establish a crucial spectral property of the generalized Sylvester operator, which justifies our methodology.

Lemma 2. *Define the linear matrix operator $\mathcal{L}(X) = \sum_{k=1}^{p} A_k X B_k$ over $\mathbb{H}_{ng}$. The operator $\mathcal{L}$ is non-singular (i.e., guarantees a unique exact solution X for any C) if and only if its base-layer complex operator*

$$\mathcal{L}^{(0)}(X^{(0)}) = \sum_{k=1}^{p} \Phi(A_k^{(0)}) X^{(0)} \Phi(B_k^{(0)})$$

is non-singular.

Proof. Applying the vectorization operator, the global Sylvester operator $\mathcal{L}$ maps to an expanded block lower-triangular complex matrix. Its n main diagonal blocks are identically $\sum_{k=1}^{p} (\Phi(B_k^{(0)}))^T \otimes \Phi(A_k^{(0)})$. Consequently, the nilpotent cross-terms strictly do not alter the base-layer spectrum, ensuring exact spectral equivalence.

The algebraic equivalence established in Section 3 allows us to avoid computing large-scale dense inverses. We now present the main theorem of this paper, which guarantees the exactness of our hierarchical decomposition strategy.

Theorem 3. *Let* (1) *be a consistent matrix equation over* $\mathbb{H}_{ng}$*. The block forward substitution generates the exact minimum-norm least-squares solution* $X = \sum_{l=0}^{n-1} X^{(l)}\varepsilon^l$ *without truncation error. Structurally, the Moore-Penrose inverse is evaluated exclusively at the base layer* ($l = 0$)*, while any higher-order layer* $X^{(l)}$ *is solved recursively using only basic matrix multiplications.*

Proof. The exactness is verified via mathematical induction on the nilpotent hierarchy l. By isolating the ε^l components within the Toeplitz isomorphism, the higher-order interactions structurally collapse onto the base-layer pseudo-inverse. Due to space constraints, the detailed step-by-step algebraic derivations are provided in the full journal version of this work.

The execution of block forward substitution is based on the column vectorization operator $\mathrm{vec}(\cdot)$ and the Kronecker product $\otimes$. The minimum-norm least-squares solution for the foundational base layer ($l = 0$) is explicitly formulated as:

$$\mathrm{vec}(X^{(0)}) = \left(\sum_{k=1}^{p} \left(B_k^{(0)}\right)^T \otimes A_k^{(0)}\right)^{\dagger} \mathrm{vec}(C^{(0)}), \tag{2}$$

where $(\cdot)^T$ denotes the transpose and $(\cdot)^{\dagger}$ is the Moore-Penrose inverse.

For any higher-order layer $l \in \{1, \ldots, n-1\}$, the solution is recursively updated by:

$$\mathrm{vec}(X^{(l)}) = \left(\sum_{k=1}^{p} \left(B_k^{(0)}\right)^T \otimes A_k^{(0)}\right)^{\dagger} \mathrm{vec}\left(C_{adj}^{(l)}\right). \tag{3}$$

Equations (2) and (3) demonstrate the core advantage of this approach: the Moore-Penrose inverse $\mathcal{K}^{\dagger}$ is evaluated only once at the base layer. For all subsequent layers, this inverse is reused, thereby bypassing global matrix inversions and explaining the complexity reduction discussed in Section 5.

Algorithm 1 Block Forward Substitution for Solving $\sum_{k=1}^{p} A_k X B_k = C$ over $\mathbb{H}_{ng}$

Input: Coefficient matrices $A_k, B_k, C \in \mathbb{H}_{ng}^{m\times m}$ for $k = 1, \ldots, p$; generalized basis parameters $\alpha, \beta > 0$; nilpotent order n.
Output: The minimum-norm least-squares solution $X \in \mathbb{H}_{ng}^{m\times m}$.

Step 1: Isomorphic Mapping
Map all matrices to the complex domain $\mathbb{C}^{2n\times 2n}$ using Definition 1:

$$A_k \xrightarrow{\Phi} \Phi(A_k), \quad B_k \xrightarrow{\Phi} \Phi(B_k), \quad C \xrightarrow{\Phi} \Phi(C).$$

Step 2: Precomputation of the Base-Layer Foundation
Extract the base-layer structural blocks $A_k^{(0)}$ and $B_k^{(0)}$.
Compute the Moore-Penrose inverse matrix:

$$\mathcal{K}^\dagger = \left[\sum_{k=1}^{p} \left(B_k^{(0)}\right)^T \otimes A_k^{(0)}\right]^\dagger.$$

Obtain the foundational layer solution $X^{(0)}$:

$$\mathrm{vec}(X^{(0)}) = \mathcal{K}^\dagger \,\mathrm{vec}(C^{(0)}).$$

Step 3: Iterative Hierarchical Substitution
For $l = 1, 2, \ldots, n-1$ **do**
Update the right-hand side:

$$C_{adj}^{(l)} = C^{(l)} - \sum_{k=1}^{p}\sum_{j=0}^{l-1}\sum_{i+m=l-j} A_k^{(i)} X^{(j)} B_k^{(m)}.$$

Solve for the l-th layer using the precomputed $\mathcal{K}^\dagger$:

$$\mathrm{vec}(X^{(l)}) = \mathcal{K}^\dagger \,\mathrm{vec}(C_{adj}^{(l)}),$$

and reshape the resulting vector back to matrix form $X^{(l)}$.
End for

Step 4: Hypercomplex Reconstruction
Assemble the computed layers into the final n-generalized solution:

$$X = \sum_{l=0}^{n-1} X^{(l)} \varepsilon^l.$$

Return X.

5 Numerical Experiments and Complexity Analysis

To assess the computational performance, we examine the complexity of solving the basic case $p = 1$.

Compared to standard real-representation solvers, the proposed method offers two significant computational advantages:

1. **Time Efficiency:** The forward substitution localizes the Moore-Penrose inverse strictly to the $m \times m$ base matrices. This reduces the floating-point operations from $\mathcal{O}(64m^3n^3)$ to $\mathcal{O}(2m^3n^3)$.
2. **Memory Scalability:** By avoiding global Kronecker expansions, the memory requirement scales linearly $(\mathcal{O}(2m^2n))$ rather than quadratically $(\mathcal{O}(16m^2n^2))$.

To validate the numerical stability of the proposed solver against parameter variations, we conducted a robustness test.

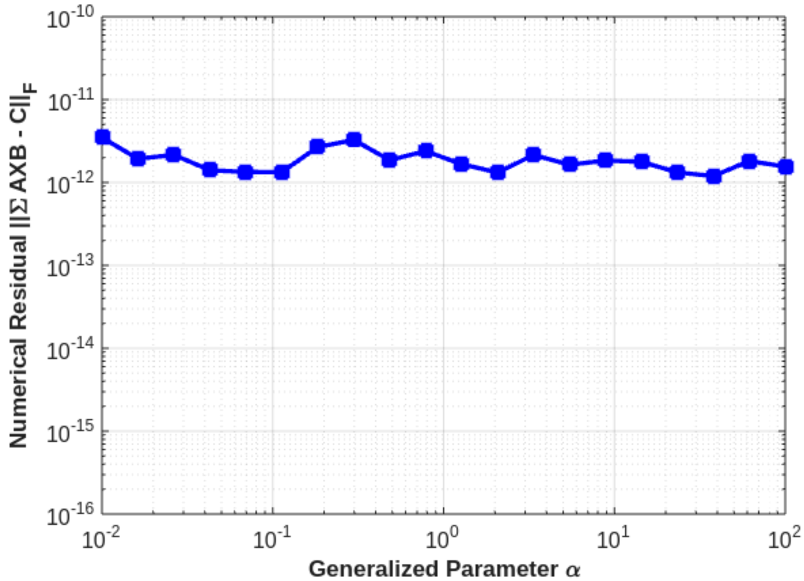

Fig. 1. Numerical residual of the computed solution as the generalized algebraic parameter α varies from 10^{-2} to 10^{2}.

As illustrated in Fig. 1, the numerical residual, measured by the isomorphic Frobenius norm $\|\sum_{k=1}^{p} A_k X B_k - C\|_{\mathbb{H}_{ng}}$ defined in Section 3, remains consistently below 10^{-11}. This demonstrates that the algorithm achieves high-precision accuracy across a wide range of algebraic parameters. These results confirm that the solver is highly robust against variations in the basis parameters, making it ideal for integration into high-dimensional mathematical software.

6 Conclusion

We have developed a computational framework for solving matrix equations over n-generalized quaternions. By exploiting the nilpotent hierarchical structure through a block forward substitution algorithm, we effectively mitigate the severe dimensional expansion inherent in traditional hypercomplex solvers. Because this approach drastically reduces memory and computational requirements, future work will focus on implementing these algorithms in parallel computing environments and developing an open-source library for hypercomplex mathematics.

Acknowledgments. The author would like to thank the anonymous referees for their valuable comments. This study was funded by Canada NSERC (RGPIN 2020-06746).

Disclosure of Interests. The authors have no competing interests to declare that are relevant to the content of this article.

References

1. Benner, P., Saak, J.: Numerical solution of large and sparse continuous time algebraic matrix riccati and lyapunov equations: a state of the art survey. Mitteilungen Der Gesellschaft Für Angewandte Mathematik Und Mechanik **36**(1), 32–52 (2013)

2. Bini, D.A., Iannazzo, B., Meini, B.: Numerical Solution of Algebraic Riccati Equations. SIAM, Philadelphia (2012)
3. Bunse-Gerstner, A., Byers, R., Mehrmann, V.: A quaternion QR algorithm. Numerische Mathematik **55**(1), 83–95 (1989)
4. Daniilidis, K.: Hand-eye calibration using dual quaternions. The International Journal of Robotics Research **18**(3), 286–298 (1999)
5. Flamant, J., Le Bihan, N., Chainais, P.: A general framework for hypercomplex signal processing. IEEE Transactions on Signal Processing **70**, 525–538 (2022)
6. Grasedyck, L., Kressner, D., Tobler, C.: A literature survey of low-rank tensor approximation techniques. Mitteilungen Der Gesellschaft Für Angewandte Mathematik Und Mechanik **36**(1), 53–78 (2013)
7. Jia, Z., Ng, M.K.: Structure preserving quaternion generalized minimal residual method. SIAM Journal on Matrix Analysis and Applications **42**(2), 616–634 (2021)
8. Jia, Z., Ng, M.K., Song, G.J.: Lanczos method for large-scale quaternion singular value decomposition. Numerical Algorithms **82**(2), 699–717 (2019)
9. Le Bihan, N., Mars, J.: Singular value decomposition of quaternion matrices: a new tool for vector-sensor signal processing. Signal Processing **84**(7), 1177–1199 (2004)
10. Liu, T.T., Yu, S.W.: Some properties of reduced biquaternion tensors. Symmetry **16**(10), 1260 (2024)
11. Liu, X., Zhang, Y.: Matrices over quaternion algebras. In: Matrix and Operator Equations and Applications, pp. 139–184. Springer, Singapore (2023)
12. Miao, J.F., Kou, K.I., Liu, W.K.: Low-rank quaternion tensor completion for recovering color videos and images. Pattern Recognition **107**, 107505 (2020)
13. Parcollet, T., Morchid, M., Linarès, G.: A survey of quaternion neural networks. The Artificial Intelligence Review **53**(4), 2957–2982 (2020)
14. Qi, L.Q., Luo, Z.: Tensor Analysis: Spectral Theory and Special Tensors. SIAM, Philadelphia (2017)
15. Saad, Y., Schultz, M.H.: Gmres: A generalized minimal residual algorithm for solving nonsymmetric linear systems. SIAM Journal on Scientific and Statistical Computing **7**(3), 856–869 (1986)
16. Sangwine, S.J.: Fourier transforms of colour images using quaternion or hypercomplex numbers. Electronics Letters **32**(21), 1979–1980 (1996)
17. Simoncini, V.: Computational methods for linear matrix equations. SIAM Review **58**(3), 377–441 (2016)
18. Voight, J.: Quaternion algebras. Springer, Cham, 1st edn. (2021)
19. Xu, Y., Yu, L.C., Xu, H.T., Zhang, H., Nguyen, T.: Vector sparse representation of color image using quaternion matrix analysis. IEEE Transactions on Image Processing **24**(4), 1315–1329 (2015)
20. Yu, C.E., Liu, X., Luo, H., Zhang, Y.: Reduced biquaternion tensors and applications in color video processing. Journal of Scientific Computing **106**(1), 7 (2026)