arXiv:2305.02477v1 [math.NA] 4 May 2023

将弗罗贝尼乌斯求逆推广到四元数矩阵

Qiyuan Chen、J. UHLMANN 和 Ke Ye

Abstract. 本文推导并分析了一种用于四元数矩阵求逆的算法。该算法是复矩阵求逆中弗罗贝尼乌斯算法的一个对应版本。在理论方面，我们证明该算法比现有的其他方法更高效。此外，就复数求逆的次数最少而言，该算法是最优的。在实践方面，对于随机生成的矩阵，我们的算法优于现有算法。我们认为，在将递归斯特拉森型算法应用于四元数矩阵时，该算法可为递归分解过程提供尽可能快的基本情形，从而提升这类算法的实际效用。

1. Introduction

四元数是超复数系的典型代表，最早由哈密顿于 1843 [18]发现。此后，四元数在数学的多个领域得到了广泛研究和应用，包括接触几何与自旋几何 [3, 26], 函数论 [38, 11],非交换代数 [24, 6, 7], 线性代数 [47] 以及代数拓扑 [20]。另一方面，由于单位四元数组成的群与由 2×2特殊酉矩阵构成的 SU(2) 群同构，因此四元数为表示空间旋转提供了一种自然的方式。[1] 由于这种令人惊讶的联系，四元数被广泛应用于刚体力学和量子力学相关领域 [21, 17, 4, 13, 30, 29, 34]。四元数还有助于将相互关联的变量组合成单一的代数实体，从而能够以自然的方式对这些变量进行整体运算。例如，可以使用单个四元数的 i、 j和 k 分量来表示 RGB 像素值。在过去十年中，基于四元数的新型方法和模型已在各种信号处理应用中不断涌现 [41, 16, 44, 40]。

四元数在科学和工程领域日益普及，推动人们更加关注核心工具的高效实现，例如四元数 LU分解 [42], 四元数奇异值分解 [25], 四元数矩阵的低秩近似 [27],以及高性能四元数矩阵乘法 [43]。

本文关注的是乘法之后最基础的四元数矩阵运算：四元数矩阵求逆。目前，文献中针对四元数矩阵求逆仅有四种著名算法 [42, 47, 39, 33], ， 其应用包括计算或估计统计学和机器学习中的协方差矩阵的逆 [2, 40, 46]; 四元数批归一化 [14, 31]; 构造正交多项式 [12]; 计算数值积分 [19]; 以及 MIMO无线电设备的硬件设计[37]。

本文结构如下。我们首先介绍矩阵求逆算法的背景，以及最高效的算法如何取决于矩阵的类型（例如实数、复数或四元数）。随后，我们推导出一种针对四元数矩阵求逆优化的算法。之后，我们从两个不同角度讨论该算法的计算复杂度。最后，我们给出实证结果，证明与文献中已有方法相比，该算法具有更高的效率。

[1] SU(2) 是由所有空间旋转构成的群 SO(3) 的双覆盖。1

2. Background

实用高效的手动矩阵求逆方法，其发展可追溯到20世纪初。弗罗贝尼乌斯于1910年取得的一项重要早期成果，提出了一种将复矩阵的逆分解为仅需计算实矩阵逆的形式的方法。20世纪50年代和60年代，人们重新关注这类方法，将其用于无法原生支持复变量的场景下求逆复矩阵；例如，数值库中只有针对实矩阵优化的例程。

1969 年，Strassen 证明，通过对矩阵乘法问题进行特殊的递归分块分解，可以获得比传统 $O(n^3)$-时间算法更低的复杂度。此外，他还证明，这意味着可以采用相关的递归算法，以相同的次立方复杂度计算矩阵的行列式和逆。数十年来，对这一方法的改进使计算复杂度不断降低 [8], ，其中 $O(n^{2.37286})$ 是目前矩阵乘法的最佳复杂度 [1] ，因此也是矩阵求逆的最佳复杂度。

尽管斯特拉森型算法具有更优的渐近计算复杂度，但其实用性取决于针对具体问题所能达到的绝对效率；也就是说，对于所关注的矩阵规模，它是否比最佳的三次复杂度算法更快。更具体地说，通用斯特拉森型递归的实际效率取决于当前最快的三次复杂度算法在基本情形下的效率。换言之，存在一个固定的矩阵规模临界点：在该规模下，采用三次复杂度算法比继续递归进行分块分解直至标量基本情形更快。对于实矩阵乘法，最佳的基本情形算法是传统的三次复杂度矩阵乘法算法。然而，对于矩阵求逆，情况要微妙得多。

在 [9], 中，作者得出了一个令人惊讶的结论：用于求复矩阵逆的弗罗贝尼乌斯算法所需的数值运算开销更小，而且其速度明显快于基于 LU 和 QR 矩阵分解的最先进方法。更具体地说，他们考察了复矩阵 $Z = A + iB$ 的弗罗贝尼乌斯求逆方法：

$$Z^{-1} = (A + BA^{-1}B)^{-1} - iA^{-1}B(A + BA^{-1}B)^{-1}, \tag{1}$$

它只需计算实矩阵的逆 A 和 B。表达式（1）只要求 A 分量非奇异 Z ，但也可以改写为只要求 B 非奇异 [2]。现在我们将说明，类似的方法也可用于提高四元数矩阵的计算效率。

3. Quaternion Matrices

四元数空间记作 $\mathbb{H}$，它通过引入两个额外的复分量／维度（记作 j 和 k），推广了复空间 $\mathbb{C}$ 。因此，复数可以用两个实标量参数 a和 b表示为 $a+ib$，而四元数有四个实参数，形式为 $a+ib+jc+kd$，即包含一个实分量和三个虚分量。四元数在代数上十分重要，因为它们构成一个具有类似矩阵性质的线性空间，例如它们不满足交换律；同时又保留了复数的许多重要性质，例如构成一个赋范除代数。

四元数的虚部与复数虚部类似，因为

$$i^2 = \text{-}1,\ j^2 = \text{-}1,\ \text{and}\ k^2 = \text{-}1. \tag{2}$$

[2]随机化，例如 Zielinski 提出的随机化方式 [48], ，可以将假设减少到非奇异Z的最小要求，但代价是增加额外开销。本文仅关注（1）的推广，并认为我们的结果可以轻易地重新表述，以满足关于Z的结构的不同假设。

采用分量叉积

$$ij = k,\ jk = i,\ \text{and}\ ki = j, \tag{3}$$

并且，调换各项的顺序会使结果乘以 -1；也就是说，按照反向循环字典序，它们满足反交换律：

$$ji = \text{-}k,\ kj = \text{-}i,\ \text{and}\ ik = \text{-}j. \tag{4}$$

虽然四元数不那么为人熟知，但自 20 世纪 90 年代以来，人们对其实际应用的兴趣稳步增长。如今，四元数已广泛用于计算机图形学和地理信息系统中，以围绕指定向量进行旋转；也应用于自动控制和系统分析，例如涉及球面状态演化的应用，包括地球周围的圆形轨道。近年来，四元数矩阵（即元素为四元数的矩阵）在众多应用中的重要性日益凸显，应用范围从图像处理中颜色信息的表示，到神经网络中的应用。

四元数矩阵可以表示为以下形式：

$$H = A + iB + jC + kD, \tag{5}$$

其中 A、 B、 C 和 D 均为实值矩阵。因此，一个自然的问题是：能否借助受弗罗贝尼乌斯针对复矩阵所用方法启发的分解，使给定四元数矩阵的求逆计算更加高效?

4. Inverting Quaternion Matrices

我们的第一个观察是，当矩阵 B、 C 和 D 中任意两个为零时，可以直接应用弗罗贝尼乌斯方法。例如，如果 B 和 C 为零，那么 $H = A + kD$ 和

$$H^{-1} = (A + DA^{-1}D)^{-1} - kA^{-1}D(A + DA^{-1}D)^{-1}, \tag{6}$$

其中，逆是在 j 复平面中计算的。由此可得出结论，任意广义弗罗贝尼乌斯型逆都必须满足上述条件：当另外两个矩阵系数为零时，对 i、 j和k 中的每一个均如此。

我们注意到， $\mathbb{R}$-代数之间存在包含关系：

$$\mathbb{R} \subseteq \mathbb{C} \subseteq \mathbb{H}.$$

这意味着

$$M_n(\mathbb{R}) \subseteq M_n(\mathbb{C}) \subseteq M_n(\mathbb{H}).$$

我们注意到 $M_n(\mathbb{C})$ 是一个 $M_n(\mathbb{R})$-双模，因为 $M_n(\mathbb{C}) \simeq M_n(\mathbb{R}) \otimes_{\mathbb{R}} \mathbb{C}$。此外， $\mathbb{C}$ 是 $\mathbb{R}$ 的二次域扩张，文献 [9] 对其上的矩阵求逆进行了详尽讨论。作为比较，我们注意到，根据盖尔范德-马祖尔定理[15, 28]， $\mathbb{H}$ 不是一个 $\mathbb{C}$ 上的代数。因此，文献 [9] 中提出的方法不能直接应用于 $M_n(\mathbb{H})$ 上的求逆，其中底域为$M_n(\mathbb{C})$。然而，我们仍然有

$$M_n(\mathbb{H}) \simeq M_n(\mathbb{C}) \otimes_{\mathbb{C}} \mathbb{H}.$$

这意味着，对于任意给定的 $Z, W \in M_n(\mathbb{H})$，我们都可以写成

$$\begin{aligned} Z &= A + iB + jC + kD = (A + iB) + j(C - iD), \\ W &= E + iF + jG + kH = (E + iF) + j(G - iH), \end{aligned}$$

其中 $A,\ B,\ C,\ D,\ E,\ F,\ G,\ H \in M_n(\mathbb{R})$。我们观察到

$$\begin{aligned} WZ &= [(A + iB) + j(C - iD)]\,[(E + iF) + j(G - iH)] \\ &= [(A + iB)(E + iF) - (C + iD)(G - iH)] + j\,[(C - iD)(E + iF) + (A - iB)(G - iH)]\,, \end{aligned}$$

因此，当且仅当 W 是 Z 的逆元时，

$$(A+iB)(E+iF)-(C+iD)(G-iH)=I_n, \tag{7}$$
$$(C-iD)(E+iF)+(A-iB)(G-iH)=0. \tag{8}$$

对（7）和（8）求解 E、F、G 以及 H，即可得到下一个引理的前两项。

引理 4.1。设 $Z=A+iB+jC+kD\in M_n(\mathbb{H})$。

(a) 若 $A+iB$ 在 $M_n(\mathbb{C})$ 中可逆，则 $(A+iB)+(C+iD)(A-iB)^{-1}(C-iD)$ 也可逆，并且

$$\begin{aligned} Z^{-1}=&\left[(A+iB)+(C+iD)(A-iB)^{-1}(C-iD)\right]^{-1}\\ &-j(A-iB)^{-1}(C-iD)\left[(A+iB)+(C+iD)(A-iB)^{-1}(C-iD)\right]^{-1}. \end{aligned} \tag{9}$$

(b) 若 $C+iD$ 在 $M_n(\mathbb{C})$ 中可逆，则 $(A+iB)(C-iD)^{-1}(A-iB)-(C+iD)$ 可逆，并且

$$\begin{aligned} Z^{-1}=&(C-iD)^{-1}(A-iB)\left[(A+iB)(C-iD)^{-1}(A-iB)+(C+iD)\right]^{-1}\\ &+j\left[-(A+iB)(C-iD)^{-1}(A-iB)-(C+iD)\right]^{-1}. \end{aligned} \tag{10}$$

证明。只需证明（a）。为此，我们回顾一下，四元数 $z=a+ib+jc+kd\in\mathbb{H}$可以表示为一个 2×2 复矩阵 $\begin{bmatrix} a+ib & c+id\\ -c+id & a-ib\end{bmatrix}$，其中 a，b，c，$d\in\mathbb{R}$。因此，我们也可以将

$Z=A+iB+jC+kD\in M_n(\mathbb{H})$ 表示为一个复 2×2 分块矩阵

$$\widehat{Z}=\begin{bmatrix} A+iB & C+iD\\ -C+iD & A-iB\end{bmatrix}\in M_{2n}(\mathbb{C}),$$

其中 A，B，C，$D\in M_n$（$\mathbb{R}$）。因此，对 Z 求逆等价于对 Z求逆。如果 $A-iB$可逆，则我们可以通过舒尔补对 Z 求逆，即，

$$\widehat{Z}^{-1}=\begin{bmatrix} W_1 & -\overline{W_2}\\ W_2 & \overline{W_1}\end{bmatrix},$$

其中 $\overline{X}$ 表示矩阵 X 的共轭，并且

$$\begin{aligned} W_1&=\left[(A+iB)+(C+iD)(A-iB)^{-1}(C-iD)\right]^{-1},\\ W_2&=(A-iB)^{-1}(C-iD)\left[(A+iB)+(C+iD)(A-iB)^{-1}(C-iD)\right]^{-1}. \end{aligned}$$

此处，$(A+iB)+(C+iD)(A-iB)^{-1}(C-iD)$ 的可逆性由 $A+iB$ 以及方程 (7) 和 (8) 保证。因此，Z 的逆为

$$Z^{-1}=W_1-jW_2$$

至此，证明完成。 □

显然，可以迭代 A、B、C、D 在引理 4.1 中的作用，从而在 M_n（$\mathbb{H}$）中获得更多求逆公式。

4.1. $M_n(\mathbb{C})$ **上的弗罗贝尼乌斯算法及其最优性**。为简化记号，我们定义

$$\begin{aligned} \mathcal{E}_1&\coloneqq\{A+iB+jC+kD\in M_n(\mathbb{H}):A+iB\in \mathrm{GL}_n(\mathbb{C})\},\\ \mathcal{E}_2&\coloneqq\{A+iB+jC+kD\in M_n(\mathbb{H}):C+iD\in \mathrm{GL}_n(\mathbb{C})\}. \end{aligned}$$

我们将引理 4.1 (a)–(b) 中的公式描述为算法 1 中的伪代码。显然，算法1 在 $M_n(\mathbb{C})$ 中需要进行 2 次求逆和 3 次乘法。接下来，我们分析算法1 的最优性。为此，我们需要以下两个引理。

算法 1 复数弗罗贝尼乌斯求逆

Input $Z = A + iB + jC + kD \in \mathcal{E}_1 \cup \mathcal{E}_2$
Output inverse of Z
1: **if** $Z \in \mathcal{E}_1$ **then**
2: compute $X_1 = (A - iB)^{-1}$, $X_2 = X_1(C - iD)$, $X_3 = (C + iD)X_2$,
3: compute $X_4 = ((A + iB) + X_3)^{-1}$, $X_5 = -X_2X_4$.
4: **return** $Z^{-1} = X_4 + jX_5$.
5: **else if** $Z \in \mathcal{E}_2$ **then**
6: compute $X_1 = (C - iD)^{-1}$, $X_2 = X_1(A - iB)$, $X_3 = (A + iB)X_2$
7: compute $X_4 = (X_3 + (C + iD))^{-1}$, $X_5 = X_2X_4$
8: **return** $Z^{-1} = X_5 - jX_4$.
9: **end if**

引理 4.2. [9] 对于任意域 $\mathbb{k}$ 和 $A, B \in M_n(\mathbb{k})$，若 A 和 $A+BA^{-1}B$ 均可逆，则计算 $(A+BA^{-1}B)^{-1}$ 至少需要进行两次矩阵求逆，无论在 $\mathbb{k}$ 中使用多少次矩阵加法、矩阵乘法或标量乘法。

Le引理 4.3. 设 p_1, p_2, q_1, q_2 是 n 个变量上的 $\mathbb{C}$ 上的多项式。假设

$$p_1(a)q_2(a) = p_2(a)q_1(a)$$

对于任意满足 $q_1(a) \neq 0$ 且 $q_2(a) \neq 0$的 $a = (a_1, \ldots, a_n) \in \mathbb{R}^n$，有 $p_1q_2 = p_2q_1$。

证明。 注意到，任意一个 $\mathbb{C}$上的多项式 f都可以唯一地写成 $f = \mathrm{Re}(f) + i\,\mathrm{Im}(f)$，其中 $\mathrm{Re}(f)$ 和 $\mathrm{Im}(f)$ 是实系数多项式。假设 $p_1q_2 - p_2q_1 \neq 0$。那么显然 $\mathrm{Re}(p_1q_2 - p_2q_1) \neq 0$ 或 $\mathrm{Im}(p_1q_2 - p_2q_1) \neq 0$成立。不失一般性，我们可以假设 $\mathrm{Re}(p_1q_2 - p_2q_1) \neq 0$。类似地，也可以假设 $\mathrm{Re}(q_1) \neq 0$ 且 $\mathrm{Re}(q_2) \neq 0$。由于 $\mathbb{R}$ 是一个无限域，存在 $a = (a_1, \cdots, a_n) \in \mathbb{R}^n$，使得

$$\left(\mathrm{Re}(p_1q_2 - p_2q_1)\,\mathrm{Re}(q_1)\,\mathrm{Re}(q_2)\right)(a) \neq 0.$$

因此（ $p_1/q_1 - p_2/q_2$)$(a) \neq 0$, $q_1(a) \neq 0$, $q_2(a) \neq 0$，这与假设矛盾 n. □

现在我们准备证明算法 1 的最优性。为简化记号，我们定义

$$Z_1 = A + iB, \quad Z_2 = C + iD.$$

因此 $Z = A + iB + jC + kD = Z_1 + jZ_2$。根据算法 1，我们也可以假设 Z_1可逆，从而 $Z^{-1} = W_1 - jW_2$ 其中

$$W_1 = \left(Z_1 + Z_2\overline{Z_1}^{-1}\overline{Z_2}\right)^{-1}, W_2 = \overline{Z_1}^{-1}\overline{Z_2}\left(Z_1 + Z_2\overline{Z_1}^{-1}\overline{Z_2}\right)^{-1}.$$

此处 $\overline{X}$ 表示复矩阵 X 的复共轭。

定理 4.4。 计算 $\left(Z_1 + Z_2{Z_1}^{-1}Z_2\right)^{-1}$ 至少需要 *1* 次加法和 *2* 次求逆。特别地，算法 *1* 在复矩阵求逆次数最少的意义下是最优的。

我们指出，定理 4.4 应理解如下：计算 Z^{-1}所需的复矩阵求逆次数最少为 2，与所使用的加法、矩阵乘法、标量乘法和共轭运算的次数无关。

证明。 1 次加法显然是必要的，因此只需证明 2 次求逆的必要性。我们采用反证法。假设存在一个只需 1 次求逆的算法。那么我们可以找到复数非交换多项式 f 和 g 使得

$$\left(Z_1 + Z_2\overline{Z_1}^{-1}\overline{Z_2}\right)^{-1} = g(Z_1, \overline{Z_1}, Z_2, \overline{Z_2}, f(Z_1, Z_2, \overline{Z_1}, \overline{Z_2})^{-1}, \overline{f(Z_1, Z_2, \overline{Z_1}, \overline{Z_2})^{-1}}) \tag{11}$$

显然，还可以进一步找到复数非交换多项式 h 和 ϕ，使得（11）可以简化如下。

$$\begin{aligned}\left(Z_1+Z_2\overline{Z_1}^{-1}\overline{Z_2}\right)^{-1} &= g\left(Z_1,\overline{Z_1},Z_2,\overline{Z_2},f(Z_1,Z_2,\overline{Z_1},\overline{Z_2})^{-1},\overline{f(Z_1,Z_2,\overline{Z_1},\overline{Z_2})^{-1}}\right)\\ &= h\left(Z_1,\overline{Z_1},Z_2,\overline{Z_2},\left(f(Z_1,Z_2,\overline{Z_1},\overline{Z_2})\overline{f(Z_1,Z_2,\overline{Z_1},\overline{Z_2})}\right)^{-1}\right)\\ &= h\left(Z_1,\overline{Z_1},Z_2,\overline{Z_2},\phi(Z_1,Z_2,\overline{Z_1},\overline{Z_2})^{-1}\right). \end{aligned} \tag{12}$$

接下来我们考虑由以下公式定义的有理映射 $\Phi\colon M_n(\mathbb{C})\times M_n(\mathbb{C}) \dashrightarrow M_n(\mathbb{C})$

$$\Phi(Z_1,Z_2)=\left(Z_1+Z_2Z_1^{-1}Z_2\right)^{-1}-h\left(Z_1,Z_1,Z_2,Z_2,\phi(Z_1,Z_2,Z_1,Z_2)^{-1}\right).$$

特别地，矩阵 $\Phi(Z_1,Z_2)$ 的元素是 $M_n(\mathbb{C})\times M_n(\mathbb{C})$ 上的有理函数。由 (12) 可立即看出，只要有定义，$\Phi\equiv 0$ 在 $M_n(\mathbb{R})\times M_n(\mathbb{R})$ 上。因此，引理 4.3 意味着，只要有定义，Φ 的元素就必须在 $M_n(\mathbb{C})\times M_n(\mathbb{C})$ 上消失。特别地，只要有定义，$\Phi(Z_1,Z_2)\equiv 0$ 在 $M_n(\mathbb{C})\times M_n(\mathbb{C})$ 上；由此我们可以得出结论，$\left(Z_1+Z_2Z_1^{-1}Z_2\right)^{-1}$ 可以通过一次复矩阵求逆来计算。这与引理 4.2 矛盾。□

4.2. $M_n(\mathbb{R})$上的弗罗贝尼乌斯算法及其复杂度。根据 [9], ，在 $M_n(\mathbb{C})$中对一般元素进行求逆需要 $M_n(\mathbb{R})$中的 2 次求逆和 3 次乘法，而在 $M_n(\mathbb{C})$中的一次乘法需要 $M_n(\mathbb{R})$中的 3 次乘法。因此，使用算法 1 以及 [9] 中用于在$M_n(\mathbb{C})$中进行乘法和求逆的算法，对一般$Z\in M_n(\mathbb{H})$进行求逆的总成本至多为 $M_n(\mathbb{R})$中的 4 次求逆和 15 次乘法。然而，我们注意到其中实际上有两次重复的乘法，因此可以得到算法 2 及其后的定理。

算法 2 实数弗罗贝尼乌斯求逆

输入一般 $Z=A+iB+jC+kD\in M_n(\mathbb{H})$ 输出 Z的逆元 1：计算 $K=C+D$。2：计算 $U_1=A^{-1}B$、$U_2=(A+BU_1)^{-1}$、$U_3=U_1U_2$、$U_4=U_3C$、$U_5=U_2D$。3：计算 $V_1=(U_2+U_3)K-U_4-U_5$、$V_2=U_4-U_5$、$V_3=V_1-V_2$、$V_4=CV_2$、$V_5=DV_1$。4：计算 $W_1=KV_3+A+V_4-V_5$、$W_2=B+V_4+V_5$、$W_3=W_1^{-1}W_2$。5：计算 $E=(W_1+W_2W_3)^{-1}$、$E_1=V_2E$。6：计算 $F=-W_3E$、$F_1=V_1F$。7：计算 $G=F_1-E_1-V_3(E+F)$、$H=F_1+E_1$。8：返回 $Z^{-1}=E+iF+jG+kH$。

命题 4.5. 算法 2 通过 Z^{-1}，使用 4 次求逆和 13 次乘法，计算 $M_n(\mathbb{R})$。

5. Comparison with existing methods

本节将比较广义弗罗贝尼乌斯求逆（算法 1 和 2）与现有四元数矩阵求逆方法的计算开销。

5.1. **四种替代方法**。据我们所知，此前文献中讨论过的四元数矩阵求逆方法只有四种。在进一步讨论之前，我们简要概述这些方法。

5.1.1. 复数方法。令 $\varphi_1: M_n(\mathbb{H}) \to M_{2n}(\mathbb{C})$ 为定义如下的单射。

$$\varphi_1(A+iB+jC+kD) = \begin{bmatrix} A+iB & C+iD \\ -C+iD & A-iB \end{bmatrix}, \quad A,B,C,D \in M_n(\mathbb{R}).$$

显然 φ_1 是一个关于 $\mathbb{R}$ 的代数同态，它将一个四元数矩阵映射为一个尺寸加倍的复矩阵。将四元数矩阵转换为复矩阵是一种标准技术， [47, 42],借助该技术，我们可以得到算法 3。

算法 3 复数方法

输入 $Z = A+iB+jC+kD \in M_n(\mathbb{H})$

输出 Z的逆矩阵1: 计算 $Y = \varphi_1(Z)$。2:
计算 $V = Y^{-1}$。3: **返回** $Z^{-1} = \varphi_1^{-1}(V)$。

5.1.2. 实数方法。我们也可以将一个四元数矩阵转换为四倍大小的实矩阵[47, 42] 通过单射 $\varphi_2: M_n(\mathbb{H}) \to M_{4n}(\mathbb{R})$ 定义如下

$$\varphi_2(A+iB+jC+kD) = \begin{bmatrix} A & B & C & D \\ -B & A & -D & C \\ -C & D & A & -B \\ -D & -C & B & A \end{bmatrix}, \quad A,B,C,D \in M_n(\mathbb{R}).$$

再次， φ_2 是一个 $\mathbb{R}$-代数同态，由此得到算法4。

算法 4 实数方法

输入 $Z = A+iB+jC+kD \in M_n(\mathbb{H})$

输出 Z的逆1: 计算 $Y = \varphi_2(Z)$。2: 计算 $V = Y^{-1}$。3: **返回** $Z^{-1} = \varphi_2^{-1}(V)$。

5.1.3. 斜对称实数方法。通过探究 $\varphi_2(A+iB+jC+kD)$ 的分块斜对称结构， [39]提出了一种更高效的方法。为便于参考，我们将该方法列于算法5中。我们指出， [39] 中的原始方法以分块矩阵的形式呈现，而算法 5则是一个更简洁且更高效的版本。

算法 5 斜对称实数方法

输入 $Z = A+iB+jC+kD \in M_n(\mathbb{H})$

输出 Z的逆

1: 计算 $U_1 = A^{-1}B$, $U_2 = A+BU_1$, $U_3 = U_2^{-1}$, $U_4 = U_1U_3$。2: 计算 $V_1 = U_3C+U_4D$, $V_2 = -U_4C+U_3D$。3: 计算 $W_1 = A+CV_1-DV_2$, $W_2 = B+DV_1+CV_2$, $W_3 = W_1^{-1}W_2$。4: 计算 $X = (W_1+W_2W_3)^{-1}$, $Y = -W_3X$, $Z = -V_1X-V_2Y$, $W = V_2X-V_1Y$。5: 返回 $Z^{-1} = X+iY+jZ+kW$。

5.1.4. *Matlab* 四元数工具箱（*QTFM*）。 QTFM 中实现了一种用于四元数矩阵求逆的四元数求逆算法 [33]。这种方法的核心思想是将矩阵 $Z \in M_n(\mathbb{H})$ 划分为

$$Z = \begin{bmatrix} Z_{11} & Z_{12} \\ Z_{21} & Z_{22} \end{bmatrix}$$

对于某个 $Z_{11} \in M_{\lfloor n/2 \rfloor}(\mathbb{H})$， $Z_{12} \in \mathbb{H}^{\lfloor n/2 \rfloor \times \lceil n/2 \rceil}$， $Z_{21} \in \mathbb{H}^{\lceil n/2 \rceil \times \lfloor n/2 \rfloor}$ 以及 $Z_{22} \in M_{\lceil n/2 \rceil}(\mathbb{H})$。然后，可以通过舒尔补 Z 计算其逆矩阵，并重复此过程以得到 [22] 的算法 6 。对于 Z^{-1}而言，算法 6 区别于算法1–5 的关键特征是，算法 6 中的运算均在 $\mathbb{H}$上进行，而非 $\mathbb{C}$ 或 $\mathbb{R}$。

算法 6 工具箱

输入 $Z \in M_n(\mathbb{H})$

Outp计算 逆矩阵 Z

1: 对 $Z = \begin{bmatrix} Z_{11} & Z_{12} \\ Z_{21} & Z_{22} \end{bmatrix}$ 进行分块。2: 计算

$U_1 = Z_{11}^{-1}$、 $U_2 = U_1 Z_{12}$、 $U_3 = Z_{21} U_1$， ▷ 在 $M_{\lfloor n/2 \rfloor}(\mathbb{H})$中求逆

3: 计算 $U_4 = (Z_{22} - U_3 Z_{12})^{-1}$、 $U_5 = U_4 U_3$。 ▷ 在 $M_{\lceil n/2 \rceil}(\mathbb{H})$中求逆

4: 计算 $R = \begin{bmatrix} U_1 + U_2 U_5 & -U_2 U_4 \\ -U_5 & U_4 \end{bmatrix}$5:

返回 $Z^{-1} = X + iY + jZ + kW$。

5.2. **比较**。 在本小节中，我们比较算法 1–6 的计算复杂度。我们用 $\mathrm{inv}_{n,\mathbb{F}}$ (分别表示 $\mathrm{mult}_{n,\mathbb{F}}$, $\mathrm{add}_{n,\mathbb{F}}$)表示($\mathbb{F}$)中的求逆（分别为乘法、加法），其中 $\mathbb{F} = \mathbb{R}, \mathbb{C}$ 或 $\mathbb{H}$。表 1记录了算法 1–6 在标为“运算”的列中所需的运算。

设 $\mathcal{A}$ 是一个在 $\mathbb{H}$上进行矩阵乘法的算法。我们用 $c_{n,\mathbb{H}}$, $c_{n,\mathbb{C}}$ 和$c_{n,\mathbb{R}}$ 分别表示 $\mathcal{A}$ 限制在 $M_n(\mathbb{H})$、 $M_n(\mathbb{C})$ 和 $M_n(\mathbb{R})$ 上的复杂度。

定理5.1。 如果我们通过 乘法$_{n,\ \mathbb{H}}$ 计算 $\mathcal{A}$ ，并通过 求逆$_{n,\ \mathbb{C}}$ *LU*分解方法计算，那么算法 *1–6* 的复杂度（忽略低阶项）如表 *1* 中标为“复杂度”的列所示。特别地，如果 $\mathcal{A}$ 是通常的矩阵乘法算法[3] ，并且在 $\mathbb{H}$ 和 $\mathbb{C}$ 中的运算采用通常方法计算[4] ，且底域为 $\mathbb{R}$，那么在算法 *1–6*中，算法*2* 是最优的。

证明。 我们回顾一下，在 $\mathbb{C}$ （分别为 $\mathbb{R}$）上通过 LU分解对一个 $n \times n$ 矩阵求逆，所需代价（忽略低阶项）为 $\frac{4n^3}{3}(\mathrm{mult}_{1,\ \mathbb{C}} + \mathrm{add}_{1,\ \mathbb{C}})$（分别为 $\frac{4n^3}{3}(\mathrm{mult}_{1,\ \mathbb{R}} + \mathrm{add}_{1,\ \mathbb{R}})$）。如果 $\mathcal{A}$ 是通常的矩阵乘法算法，那么我们有

$$c_{n,\mathbb{F}} = n^3(\mathrm{mult}_{1,\mathbb{F}} + \mathrm{add}_{1,\mathbb{F}}),$$

其中 $\mathbb{F} = \mathbb{H}$， $\mathbb{C}$ 或 $\mathbb{R}$。此外，如果我们按通常方式计算 $\mathbb{H}$ 和 $\mathbb{C}$ 中的运算，那么还有

$$\mathrm{mult}_{1,\mathbb{H}} = 16\,\mathrm{mult}_{1,\mathbb{R}} + 12\,\mathrm{add}_{1,\mathbb{R}}, \quad \mathrm{add}_{1,\mathbb{H}} = 4\,\mathrm{add}_{1,\mathbb{R}},$$
$$\mathrm{mult}_{1,\mathbb{C}} = 4\,\mathrm{mult}_{1,\mathbb{R}} + 2\,\mathrm{add}_{1,\mathbb{R}}, \quad \mathrm{add}_{1,\mathbb{C}} = 2\,\mathrm{add}_{1,\mathbb{R}}.$$

这意味着，算法 1–6 的运算量依次为 $\frac{136n^3}{3}$， $\frac{110n^3}{3}$， $\frac{256n^3}{3}$， $\frac{512n^3}{3}$， $\frac{128n^3}{3}$， $128n^3$ 浮点运算次数，分别对应 $\mathbb{R}$，由此可以得出算法 2 的最优性。 □

[3]这指的是通过矩阵乘法的定义得到的算法。[4]这指的是通过 $\mathbb{H}$ 和 $\mathbb{C}$中的乘法和加法定义得到的算法。

算法	运算	复杂度
1	$2\,\mathrm{inv}_{n,\mathbb{C}} + 3\text{乘}_{n,\mathbb{C}} + \text{添加}_{n,\mathbb{C}}$	$\frac{8n^3}{3}(\mathrm{mult}_{1,\mathbb{C}} + \mathrm{add}_{1,\mathbb{C}}) + 3c_{n,\mathbb{C}}$
2	$4\,\mathrm{inv}_{n,\mathbb{R}} + 13\text{乘}_{n,\mathbb{R}} + 17\text{添加}_{n,\mathbb{R}}$	$\frac{16n^3}{3}(\mathrm{mult}_{1,\mathbb{R}} + \mathrm{add}_{1,\mathbb{R}}) + 13c_{n,\mathbb{R}}$
3	$\text{逆}_{2n,\mathbb{C}}$	$\frac{32n^3}{3}(\text{乘}_{1,\mathbb{C}} + \text{加}_{1,\mathbb{C}})$
4	$\text{逆}_{4n,\mathbb{R}}$	$\frac{256n^3}{3}(\text{乘}_{1,\mathbb{R}} + \text{加}_{1,\mathbb{R}})$
5	$4\,\text{逆}_{n,\mathbb{R}} + 16\text{乘}_{n,\mathbb{R}} + 10\text{添加}_{n,\mathbb{R}}$	$\frac{16n^3}{3}(\mathrm{mult}_{1,\mathbb{R}} + \mathrm{add}_{1,\mathbb{R}}) + 16c_{n,\mathbb{R}}$
6	$n\,\mathrm{inv}_{1,\mathbb{H}} + \sum_{j=1}^{\log_2 n} 2^j(3\,\text{乘法}_{n/2^j,\mathbb{H}} + \text{加法}_{n/2^j,\mathbb{H}})$	$3\sum_{j=1}^{\log_2 n} 2^j c_{n/2^j,\mathbb{H}}$

TABLE 1. 复杂度比较

推论5.2。设 $\mathcal{A}$ 是一个比通常算法更快的矩阵乘法算法。如果我们在 $\mathbb{H}$ 和 $\mathbb{C}$ 中通过通常方法对 $\mathbb{R}$进行运算，那么算法 *2* 在算法*1–6*中仍然是最优的。

我们通过对 $\mathbb{H}$ 和 $\mathbb{C}$ 在 $\mathbb{R}$上的运算作一番说明来结束本小节。容易证明，用于加$_{1,\mathbb{H}}$ 和加$_{1,\mathbb{C}}$ 的通常算法已经是最优的。已知[23, 10] 任何用于乘$_{1,\mathbb{H}}$ 的非交换算法至少需要 8 次乘法运算（在 $\mathbb{R}$中），然而目前尚无已知的用于乘$_{1,\mathbb{H}}$ 的算法仅需 8 次乘法运算且在 $\mathbb{R}$中少于 32 次加法运算。因此，就所有现有的[5] 用于乘$_{1,\mathbb{H}}$的算法而言，通常算法实际上是最优的（即乘法运算和加法运算总成本意义下的最优算法）。

类似地，众所周知， [35, 45, 36, 32, 5] 对于多$_{1,\mathbb{C}}$中的任何非交换算法都至少需要 3 次乘法，$\mathbb{R}$ 且高斯算法达到了最优性。然而，高斯算法在 $\mathbb{R}$ 中需要 5 次加法。因此，在所有已知的多$_{1,\mathbb{C}}$算法中，通常算法效率最高。

5.3. **实验**。对于每个整数 $1 \le m \le 50$，我们取 $n = 100m$ 并应用算法 1–6对 $M_n(\mathbb{H})$ 中的随机矩阵进行求逆。更具体地说，我们生成 $n \times n$ 实矩阵 A、B、C和 D，其元素均匀地取自区间 $(-1,1)$。那么，以概率 1，算法 1–6 均适用于 $Z = A + iB + jC + kD \in M_n(\mathbb{H})$。对于每个 m，我们将上述过程重复 50 次，从而获得 50 个测试样本。

我们在图 1中记录了算法 1–6各自的平均运行时间。为便于比较，我们还计算了比值

$$r_{n,s} = \frac{t_{n,5}}{t_{n,s}}, \quad s \in \{1,2,3,4,6\}.$$

此处 $t_{n,p}$表示算法 p在规模为 $n \times n$的实例上的平均运行时间。根据定义，$r_{n,s} > r_{n,s'}$ 表示算法 s 比算法 s'更高效。图 2 展示了$r_{n,s}$ 随 n 增大时在每个 $s \in \{1,2,3,4,6\}$下的变化情况。从图 1 和 2 可以清楚看出，算法 2明显快于其他五种算法。

Let Z 按平均右残差矩阵 Z 由算法 1–6 得到。我们对.....的质量进行评估

$$\mathrm{res}_n = \frac{\|Z\widehat{Z} - I_n\|_F}{n^2}.$$

此处 $\|\cdot\|_F$ 是 $M_n(\mathbb{H})$ 上的弗罗贝尼乌斯范数。我们在图 3 中记录了算法 1–6各自 50 个样本的平均右残差均值。由图3可轻易得出

[5]我们再次强调，我们是根据目前可用的三次复杂度四元数算法集合来评估最优性。当然，这些算法以及我们提出的算法，都可以作为基本情形应用于当前最佳 Strassen 型算法所使用的通用递归中，而该集合中最优的算法显然能够为整体次立方复杂度提供最佳系数。

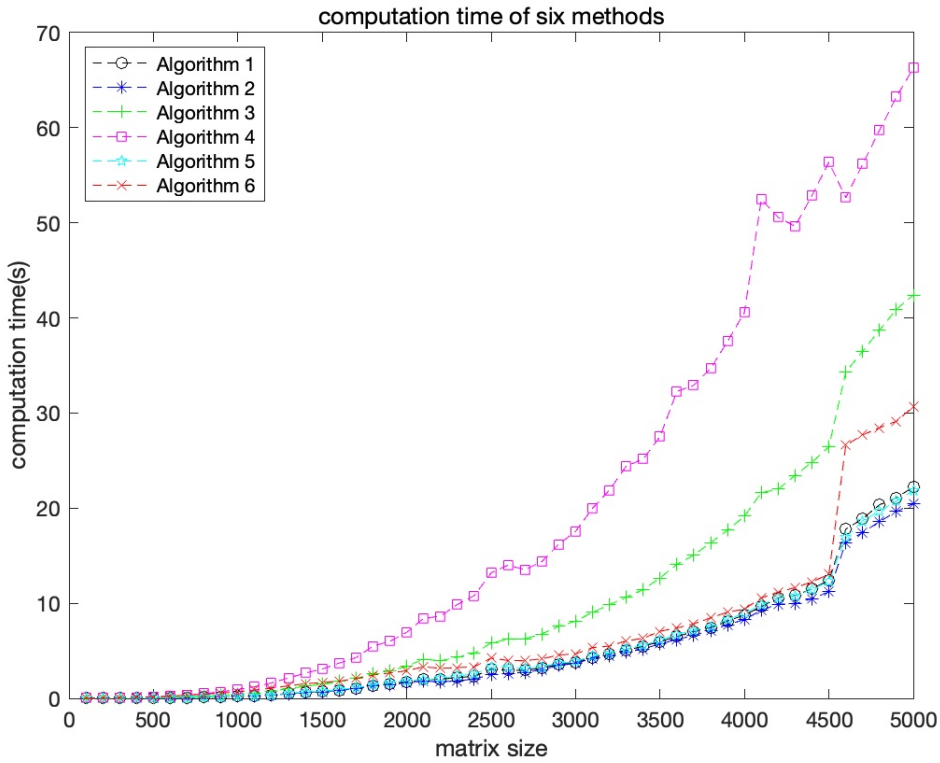

FIGURE 1. 算法 2（蓝色）与其他五种算法相比，计算开销显著更低。

算法的平均右残差 2、5和 6小于 $5\cdot 10^{-13}$，而其他三种算法的平均右残差相对较小。

6. DISCUSSION

本文提出了一种高效求逆四元数矩阵的新算法，并通过实证结果表明，该算法是文献中现有四元数求逆算法里速度最快的。因此，它可作为 Strassen 型求逆算法的优化递归基例，从而在求逆四元数矩阵时实现计算复杂度与实际性能的最佳结合。

REFERENCES

[1] J. Alman 和 V. V. Williams。一种改进的激光方法及更快的矩阵乘法。载于 D. Marx 编， *Proceedings of the 2021 ACM-SIAM Symposium on Discrete Algorithms SODA 2021 Virtual Conference January ,,,,, 第 – 10 - 13 2021 522 539 2021 页。SIAM, 。[2] T. W. Anderson。 An introduction to multivariate statistical analysis。John Wiley & Sons Inc., 1958年。[3] V. I. Arnold。球面曲线的几何与四元数代数。 Russian Mathematical Surveys, 50 1 68, 1995年: –。[4] M. Arribas、A. Elipe 和 M. Palacios。四元数与刚体旋转。 Celestial Mechanics and Dynamical Astronomy 96 3 239 251, 2006 , (): –。

[5] P. B ¨ Algebraic complexity theory 315 Grundlehren der mathematischen Wissenschaften [Fundamental Principles of Mathematical Sciences] urgisser、M. Clausen 和 M. A. Shokrollahi。[5] , 第 卷。Springer-Verlag, 柏林, 1997年。Thomas Lickteig 参与合作。

[6] C. C. Chevalley、P. Cartier 和 C. Chevalley。旋量与克利福德代数的代数理论。1996年。

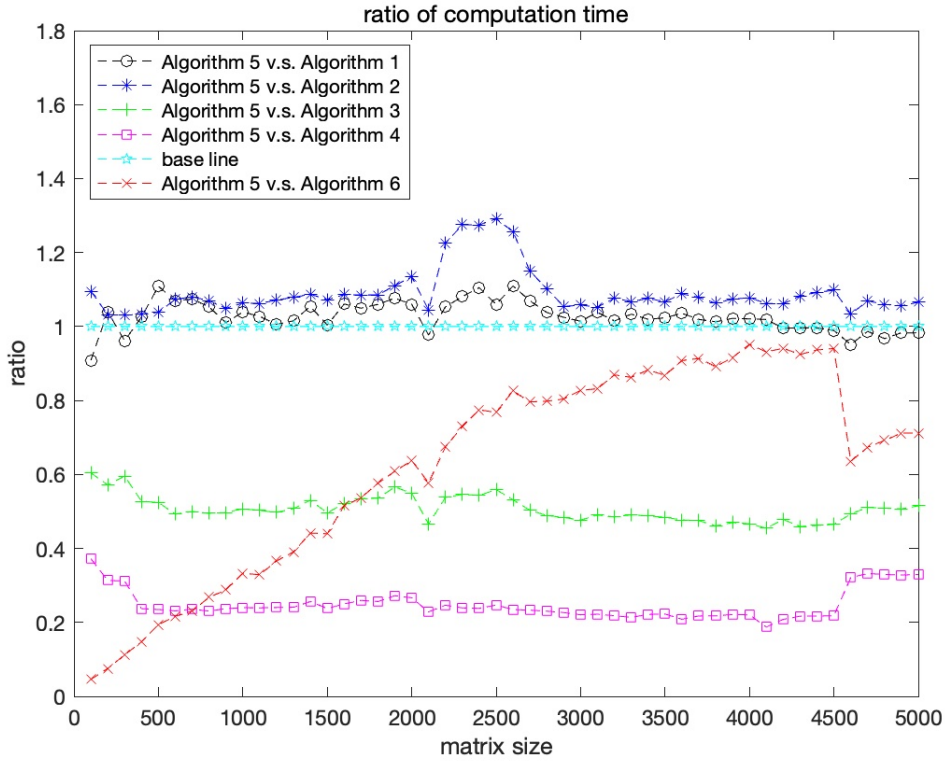

FIGURE 2. 算法 2 的比率（蓝色）显著高于其他算法的比率。

[7] P. M. Cohn。 *Skew Fields: Theory of General Division Rings*。《数学及其应用百科全书》。剑桥大学出版社，1995年。[8] D. Coppersmith 和 S. Winograd。通过等差数列进行矩阵乘法。 *J. Symb. Comput.*，9(3):251–280，1990年。[9] Z. Dai、L.-H. Lim 和 K. YE。复矩阵求逆。 *arXiv*，预印本，2022年。[10] H. F. de Groote。关于四元数乘法的复杂度。 *Information Processing Letters* 3 6 177 179, , ( ): –。1975
[11] C. A. Deavours。四元数微积分。 *The American Mathematical Monthly*，80(9):995–1008，1973年。[12] B. A. Dent 和 A. Newhouse。离散域上的正交多项式。 *SIAM Review*，1(1):55–59，1959年。[13] H. M. Faugeras OD。 3 *The International Journal of* 维物体的 *Robotics Research* 5 3 27 52, 1986
表示、识别与定位。, ( ): –。[14] C. J. Gaudet 和 A. Maida。深度四元数网络。 *2018 International Joint Conference on Neural Networks (IJCNN)* 1 8, 2017 , 第 – 页。

[15] I. Gelfand。范数环。 *Mat. Sb.*，9:43–54，1941年。[16] L. Ghouti。利用四元数奇异值分解进行稳健的感知彩色图像哈希。载于 *International Conference on Acoustics Speech and Signal Processing (ICASSP)* 3794 3798, 2014 , , 第 – 页。[17] C. GRUBIN。通过欧拉轴和角度推导四元数方案。 *Journal of Spacecraft and Rockets*，7 10 1261 1263，1970 ( ): –。[18] W. R. Hamilton。 *Elements of Quaternions*。《剑桥图书馆藏书系列——数学》。剑桥大学出版社，2010年。[19] M. T. Heath、G. A. Geist 和 J. B. Drake。在橡树岭国家实验室使用 Intel iPSC/860 的早期经验。 *The International Journal of Supercomputing Applications*，5(2):10–26，1991年。[20] H. Hopf。 ¨关于三维球面到球面的映射。 *Mathematische Annalen*，104 1 637 665，1931 ( ): –。[21] B. K. P. Horn。利用单位四元数求解绝对定向的闭式解。 *of America A-optics Image Science and Vision* 4 629 642, 1987 *Journal of The Optical Society* , : –。

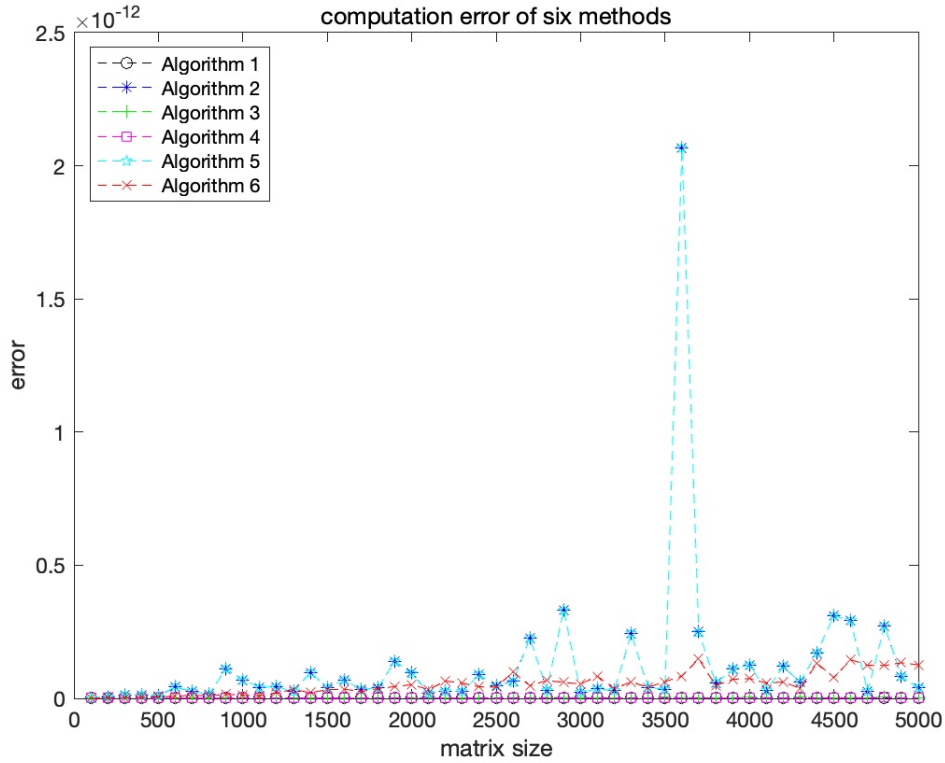

FIGURE 3. 算法 1 的平均右残差至多为 $5 \cdot 10^{-13}$ 适用于除一个维度外的所有维度。

[22] R. A. Horn 和 C. R. Johnson。《 *Matrix Analysis*》。剑桥大学出版社，第2版，2012年。[23] T. D. Howell 和 J. P. J. Lafon。四元数乘积的复杂度。1975年。[24] A. Hurwitz。 ¨关于二次型的合成。 *Mathematische Annalen*, 88(1):1–25，1922年。[25] Z. Jia、M. K. Ng 和 G.-J. Song。用于大规模四元数奇异值分解的Lanczos方法。 *Numerical Algorithms*，82(2):699–717，2019年。[26] H. B. Lawson 和 M.-L. Michelsohn。《 *Spin Geometry (PMS-38)*》。普林斯顿大学出版社，1989年。[27] Q. Liu、S. Ling 和 Z. Jia。用于低秩矩阵近似的随机化四元数奇异值分解。 *SIAM J. Sci. Comput.*，44(2):A870–A900，2022年1月。[28] A. Mallios。《 *Topological algebras. Selected topics*》， *North-Holland Mathematics Studies*第124卷。North-Holland Publishing Co.，阿姆斯特丹，1986年。Notas de Matem´atica [《数学札记》], 109。[29] M. Nakano、J. Seino 和 H. Nakai。具有时间反演对称性的自旋相关相对论开壳层Hartree–Fock理论的发展（I）：非限制方法。 *International Journal of Quantum Chemistry*，117 10 25356，2017 ( ):e 。[30] J. Paldus、T. Sako、X. Li 和 G. H. F. Diercksen。独立粒子模型中的对称性破缺：Hartree–Fock势奇异行为的本质。 *Journal of Mathematical Chemistry*，51(2):427–450，2013年。[31] T. Parcollet、M. Morchid 和 G. Linar`es。四元数神经网络综述。 *Artificial Intelligence Review*，53 2957 2982，2019 : – 。[32] K. Ramachandra 和 V. Strassen。避免除法。 *Journal für die reine und angewandte Mathematik (Crelles Journal)* 1973 184 202, 1973 , : – 。[33] S. J. Sangwine 和 N. L. Bihan。Matlab 四元数工具箱，3.4 版，2022 年。 *available online*: https://qtfm.sourceforge.net。[34] K. Shoemake。使用四元数曲线制作旋转动画。SIGGRAPH '85，纽约，纽约州，美国，1985年。计算机协会。[35] V. Strassen。一般张量的秩与最优计算。 *Linear Algebra and its Applications*, 52-53:645–685，1983年。[36] V. STRASSEN。相对双线性复杂度与矩阵乘法。1987(375-376):406–443，1987年。

[37] C. Studer、S. Fateh 和 D. Seethaler。使用最小均方误差并行干扰消除实现软输入软输出多输入多输出检测的专用集成电路。 *IEEE Journal of Solid-State Circuits*，46:1754–1765，2011年。[38] A. Sudbery。四元数分析。 *Mathematical Proceedings of the Cambridge Philosophical Society*，85 2 199( ): –225，1979年。[39] J. D. Turner。使用实变量矩阵运算求逆n×n四元数方程的斜对称算法。 *The Journal of the Astronautical Sciences*，57(1-2):275–287，2009年。[40] J. V´ıa、D. P. Palomar、L. Vielva 和 I. Santamar´ıa。四元数高斯向量的最大似然独立成分分析。载于 *2011 IEEE International Conference on Acoustics Speech and Signal Processing (ICASSP)* 4260 4263, , , 2011 页码 – 。[41] J. V´ıa、D. Ram´ırez 和 I. Santamar´ıa。四元数随机向量的适当性与广义线性处理。 *IEEE Transactions on Information Theory*，56(7):3502–3515，2010年。[42] M. Wei、Y. Li、F. Zhang 和 J. Zhao。《 *Quaternion matrix computations*》。Nova Science Publishers, Incorporated，2018年。[43] D. B. Williams-Young 和 X. Li。关于四元数矩阵乘法的效能及高性能实现。 *ArXiv*，abs/1903.05575，2019年。[44] Y. Xu、L. Yu、H. Xu、H. Zhang 和 T. Nguyen。利用四元数矩阵分析进行彩色图像的向量稀疏表示。24(4)，2015年。[45] K. Ye 和 L.-H. Lim。快速结构化矩阵计算：张量秩与Cohn–Umans方法。 *Foundations of Computational Mathematics* 18 1 45 95, 2018 , ( ): – 。[46] M. Yuan。通过线性规划估计高维逆协方差矩阵。 *J. Mach. Learn. Res.*，11 2261 2286，2010 年 8 月 : – 。[47] F. Zhang。四元数与四元数矩阵。 *Linear Algebra and its Applications*，251:21–57，1997年。[48] A. Zielinski。复矩阵求逆。 *Internat. J. Numer. Methods Engrg.*，14(10):1563–1566，1979年。

KLMM, Academy of Mathematics and Systems Science, Chinese Academy of Sciences, Beijing 100190,China*Email address*: chenqiyuan@amss.ac.cn
Department of Electrical Engineering and Computer Science, University of Missouri - Columbia *Email address*: uhlmannj@missouri.eduKLMM, Academy of Mathematics and Systems Science, Chinese Academy of Sciences, Beijing 100190,China*Email address*: keyk@amss.ac.cn