# From Rotation Matrices to Quaternions: Derivations, Efficient Algorithms, and Applications in Computing and Physics

Chongjing Li
*Guangdong Guangya High School, Nanyuan Street, Liwan District, China*

Keywords: Rodrigues Rotation Formula, 3D Rotation, Quaternion Derivation, Efficient Algorithms, Applications in Computing and Physics.

Abstract: This paper explores the mathematical foundations and applications of quaternions in representing three-dimensional rotations. Quaternions, discovered by William Rowan Hamilton in 1843, provide a powerful and efficient alternative to traditional rotation matrices. The paper begins by deriving the rotational quaternion from rotation matrices, highlighting the mathematical relationship between the two. It then compares the advantages and disadvantages of using quaternions versus rotation matrices for 3D rotations, emphasizing the efficiency and numerical stability of quaternions. The paper also introduces new algorithms for reducing the computational complexity of quaternion operations while maintaining accuracy. Finally, it discusses the extensive applications of quaternions in computer animation and modern physics, demonstrating their versatility and importance in enhancing computational efficiency and realism. By addressing both theoretical and practical aspects, this paper aims to provide a comprehensive overview of quaternions and their significance in various fields.



## 1 INTRODUCTION

William Rowan Hamilton, an Irish mathematician, is renowned for his groundbreaking discovery of quaternions. The story of this discovery is both fascinating and emblematic of a moment of genius. In the early 1840s, Hamilton had been deeply engaged in the study of complex numbers and their geometric interpretation in two dimensions. He sought to extend this concept to three dimensions by finding a way to multiply "triplets" of numbers, but he faced significant challenges.

On October 16, 1843, Hamilton and his wife were taking a walk along the Royal Canal in Dublin, heading to a meeting of the Royal Irish Academy. As they crossed Brougham Bridge (now known as Broom Bridge), Hamilton experienced a sudden flash of insight. He realized that the solution lay not in three dimensions, but in four. He conceived the fundamental formula for quaternion multiplication:

$$i^2 = j^2 = k^2 = ijk = -1 \quad (1)$$

This formula defined a new algebraic system where multiplication was non-commutative, meaning that the order of multiplication mattered. Excited by his discovery, Hamilton immediately carved this equation into the stone of the bridge. This act of spontaneous graffiti has since become a celebrated moment in the history of mathematics.

Hamilton's quaternions revolutionized the field by providing a powerful tool for describing three-dimensional rotations and spatial transformations. Although quaternions were initially met with skepticism due to their departure from traditional algebraic rules, they eventually found wide applications in various fields, including computer graphics, robotics, and quantum physics. Today, the discovery is commemorated by an annual "Hamilton Walk" that retraces his steps along the Royal Canal.
This article is mainly about the summary of previous work done on the concept of quaternion and rotation, which includes the derivation of quaternion by using rotational matrix, the comparison between using quaternion and rotational matrix in representing 3D rotation, new algorithm for reducing the complexity of quaternion calculation, and the application of quaternion in computer animation and modern physics. (Kuipers, 2002). Hanson (2006) delves into quaternions, offering a comprehensive visualization analysis that aids in understanding their representation of rotations in three-dimensional space.



Li, C.
From Rotation Matrices to Quaternions: Derivations, Efficient Algorithms, and Applications in Computing and Physics.
DOI: 10.5220/0013825500004708

In *Proceedings of the 2nd International Conference on Innovations in Applied Mathematics, Physics, and Astronomy (IAMPA 2025)*, pages 342-347
ISBN: 978-989-758-774-0

Griffiths (2017) explores the application of quaternions in quantum mechanics, particularly their advantages in describing particle rotations. In the realm of data processing, Fletcher and Joshi (2007) discuss the application of quaternions in tensor data analysis, showcasing their potential for handling intricate data structures. Regarding robotic applications, Sarabandi and Ha (2018) propose an efficient quaternion-based computation method for pose estimation. Lastly, Kim and Nam (2004) introduce a quaternion-based interpolation technique that significantly enhances the smoothness of rotational transitions in computer animations.

The first section will be about introducing the concept of quaternion and how it is applied to represent 3D rotation. This section will first start from using traditional method of rotational matrix to figure out each component of a rotational quaternion. Then, using the rotational quaternion and formula to show how it represents 3D rotation. After the derivation, the section will talk about the limitations of using traditional methods in representing 3D rotations and advantages of using quaternion.

The second section will be about introducing new algorithm for reducing the complexity of quaternion calculation. This is a relatively new development in the field.

The final section will be about the application in computer animation and modern Physics. It will talk about how quaternion plays a role in representing rotation when using cameras to shoot photos. Also, it will cover how the quaternion was used in certain fields in Physics such as quantum mechanics.

# 2 FIRST SECTION

This section will be about proving the rotational quaternion by using rotational matrix.

Firstly, there are some prefix rules for quaternion multiplication:

$$ii=jj=kk=-1 \tag{2}$$

$$ij=-ji=k \tag{3}$$

$$jk=-kj=i \tag{4}$$

$$ki=-ik=j \tag{5}$$

From the prefix rules above, we can see that the quaternion multiplication satisfies the right-hand rule, and it doesn't satisfy the commutative law.

The basic form of a quaternion is shown below

$$q=a+bi+cj+dk \tag{6}$$

Where a is a real number and the rest of them are imaginary number.

From the basic form above we can write down our rotational quaternion which is

$$q=cos(\frac{\theta}{2})+\sin(\frac{\theta}{2})u \tag{7}$$

## 2.1 The Derivation of Rotational Quaternion by Using Rotational Matrix

Firstly, there is a $3\times3$orthogonal Rodrigues rotation matrix, which is used to represent the rotation about a unit vector $u=(u_x,u_y,u_z)$ with the degree of $\theta$

Then, following steps can be made after using the matrix

$$q_0=1/2\sqrt{1+R_{11}+R_{22}+R_{33}} \tag{8}$$

$$q_1=1/4q_0(R_{32}-R_{23}) \tag{9}$$

$$q_2=1/4q_0(R_{13}-R_{31}) \tag{10}$$

$$q_3=1/4q_0(R_{21}-R_{12}) \tag{11}$$

After getting these final results we can have following steps and outcomes

$$q_0=\cos(\theta/2) \tag{12}$$

$$q_1=u_x\,sin(\theta/2)\,i \tag{13}$$

$$q_2=u_y\,sin(\theta/2)\,j \tag{14}$$

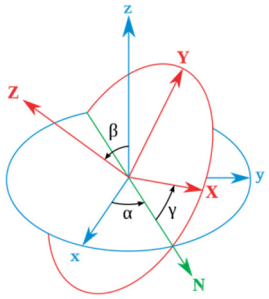
$$q_3=u_z\,sin(\theta/2)\,k \tag{15}$$

$$q=cos(\theta/2)+sin(\theta/2)u \tag{16}$$

Then, the basic form of the rotational quaternion formula which is

$$u'=pvp^{-1} \tag{17}$$

For a pure quaternion v = $(0,v_x,v_y,v_z)$




$$\vec{v} = \vec{v}_{\uparrow\uparrow} + \vec{v}_{\perp} \tag{18}$$

$$\vec{v}_{\uparrow\uparrow} = (\vec{v} \cdot \vec{u}) \cdot u \tag{19}$$

$$\vec{v}_{\perp} = \vec{v} - \vec{v}_{\uparrow\uparrow} \tag{20}$$

$$q\vec{v}_{\perp}q^{*} = cos\,\theta\,\vec{v}_{\perp} + sin\,\theta\,(\vec{u} \times \vec{v}_{\perp}) \tag{21}$$

$$\vec{v'} = \vec{v}_{\uparrow\uparrow} + cos\,\theta\,\vec{v}_{\perp} + sin\,\theta\,(\vec{u} \times \vec{v}_{\perp}) \tag{22}$$

This result agrees with the formula we need to prove.

## 2.2 The Comparison of Quaternion and Rotational Matrix in Representing 3D Rotation

Rotation matrices are a widely used mathematical tool for representing three-dimensional rotations due to their intuitive nature and compatibility with computer graphics and other applications. A rotation matrix is a 3×3 orthogonal matrix with a determinant of 1, which directly represents the rotation operation and allows for straightforward visualization and computation. This makes it particularly suitable for applications where transformations need to be combined with other operations such as translation and scaling, as it can be easily integrated into a 4×4 homogeneous transformation matrix. Additionally, the inverse of a rotation matrix is simply its transpose, simplifying computations and ensuring numerical stability.

However, rotation matrices have several drawbacks. They require storing nine elements, even though only three parameters are needed to describe a rotation, leading to inefficient storage and computation, especially in scenarios involving frequent rotation combinations. Matrix multiplication for rotation matrices is computationally intensive, requiring 27 floating-point operations for a 3×3 matrix multiplication. Furthermore, when used in conjunction with Euler angles, rotation matrices can encounter the gimbal lock problem, which results in the loss of a degree of freedom. Lastly, interpolating between rotation matrices, such as for smooth transitions in animations, is challenging and often requires conversion to alternative representations like quaternions.

Despite these limitations, rotation matrices remain a valuable tool in applications where intuitive visualization and compatibility with existing systems are prioritized.

Quaternions offer several advantages for representing three-dimensional rotations. They require storing only four elements, making them more storage-efficient than rotation matrices, which store nine elements. This efficiency extends to computational operations, as quaternion multiplication involves only 16 floating-point operations, compared to the 27 operations required for 3×3 matrix multiplication. Quaternions also avoid the gimbal lock problem, ensuring full representation of rotations in three-dimensional space. Additionally, quaternions provide a numerically stable method for representing rotations, even after multiple combinations of rotations. Interpolation between quaternions, such as spherical linear interpolation (Slerp), is straightforward and efficient, making them ideal for applications requiring smooth transitions, such as animations and simulations. Furthermore, quaternions inherently maintain unit length when normalized, which helps in preserving rotational integrity during computations. (Shoemake, 1985)

However, quaternions also have notable disadvantages. Their mathematical concept is more abstract compared to rotation matrices, making them less intuitive for beginners. The geometric interpretation of quaternions is not as straightforward, which can complicate debugging and visualization. Quaternion operations with vectors require converting vectors into quaternion form, performing quaternion multiplication, and then converting back to vector form, adding complexity to the process. Additionally, converting between quaternions and Euler angles involves intricate calculations, which can be cumbersome in applications where Euler angles are preferred. Lastly, while quaternions are numerically stable, they require careful handling to avoid issues like double rotations or sign ambiguities.

Despite these challenges, quaternions remain a powerful tool in applications where computational efficiency, interpolation capabilities, and avoidance of gimbal lock are critical. They are particularly well-suited for fields such as computer graphics, robotics, and aerospace engineering, where frequent rotation combinations and smooth transitions are essential.

## 2.3 New Algorithms for Computing Quaternion and Rotation

There are a few new algorithms for reducing the complexity of quaternion multiplication and maintain the accuracy of its outcome. Also, there are other algorithms for simplifying storage structure and enhancing interpolation efficiency. They serve for the same goal but can be used for different purposes.

### 2.3.1 Algorithms for Reducing the Complexity

Optimizing quaternion multiplication is essential for enhancing the efficiency of 3D rotation calculations. One effective approach involves "exploiting symmetry and mathematical identities". By precomputing intermediate terms such as $q1_w q2_w, q1_x q2_x$, and others, redundant calculations can be minimized, thereby reducing the total number of floating-point operations required for quaternion multiplication. This method leverages the inherent symmetry in the multiplication process to streamline computations.

Another optimization strategy involves the use of approximation algorithms. In scenarios where minimal accuracy loss is acceptable, such as in real-time graphics or simulations, approximation methods like Taylor series expansion or polynomial approximation can replace exact calculations. These algorithms estimate complex functions involved in quaternion operations, significantly reducing computational overhead while maintaining acceptable precision.

Incremental update strategies are particularly useful in applications requiring frequent quaternion updates, such as animations or physical simulations. By adjusting quaternion values incrementally and controlling the size of these increments, the need for full quaternion multiplication and normalization can be minimized, thereby reducing computational load.

Parallel computing represents a powerful optimization technique for quaternion multiplication. By distributing the computation of quaternion components across multiple processing units, such as GPUs or multi-core CPUs, the overall computation time can be significantly reduced. This approach leverages the parallel processing capabilities of modern hardware to accelerate quaternion operations.

Lastly, storage optimization can enhance computational efficiency by reducing memory access overhead. Since quaternions are unit-length, storing only three components and deriving the fourth when needed can save memory and improve access efficiency. This method is particularly beneficial in resource-constrained environments.

These optimization methods collectively enhance the efficiency of quaternion-based 3D rotations, making them more suitable for applications demanding high computational performance. By selecting the appropriate optimization strategy based on specific requirements, developers can achieve significant improvements in both speed and resource utilization.

### 2.3.2 Algorithms for Maintaining the Accuracy of Quaternion Calculation

Enhancing the numerical stability of quaternion-based 3D rotations is crucial for ensuring accurate and reliable computations in various applications. One effective approach is "incremental normalization", which adjusts the quaternion incrementally to maintain unit length. This method avoids the computational overhead of frequent full normalization by making small adjustments during each update step, ensuring the quaternion remains close to unit length without significant computational cost. This strategy is particularly beneficial in real-time applications where efficiency is paramount.

Another strategy involves " error correction mechanisms " that monitor and correct deviations from unit length. These mechanisms prevent error accumulation by dynamically adjusting the quaternion's components when deviations are detected. By integrating these corrections into the rotation update loop, numerical errors are minimized, ensuring stable and accurate rotations over multiple operations. This approach is especially valuable in iterative algorithms where small errors can compound and affect overall accuracy.

"Robust interpolation algorithms" also play a key role in improving numerical stability. Traditional interpolation methods like Slerp can encounter instabilities, particularly when the angle between quaternions is close to 0 or π. To address this, advanced interpolation algorithms use polynomial approximations or other techniques to handle these edge cases more effectively. These methods ensure smooth and stable transitions, even under challenging conditions, making them suitable for applications requiring high precision.

Finally, "hybrid representation methods" offer a powerful solution by switching between quaternions and other representations like rotation matrices or Euler angles. This approach avoids singularities and numerical issues by leveraging the strengths of different representations based on the current rotation

state. Hybrid methods are particularly effective in complex simulations and robotics applications where singularities are more likely to occur.

By integrating these strategies, developers can achieve stable and accurate quaternion-based rotations, extending their applicability to demanding scenarios from real-time animations to complex simulations. These methods collectively address common numerical challenges, ensuring reliable performance in a wide range of applications.

# 3 THE APPLICATION OF QUATERNION IN MODERN PHYSICS AND COMPUTER ANIMATION

Quaternions have become indispensable in modern physics and computer graphics, offering efficient and numerically stable representations of 3D rotations. Their ability to avoid the gimbal lock problem inherent in Euler angles makes them particularly valuable in various scientific and creative applications. In rigid body dynamics simulations, quaternions provide a robust framework for representing rotational motion. For example, in simulations of celestial mechanics, quaternions are used to model the complex rotational dynamics of planets and moons, ensuring precise and stable calculations that would be challenging to achieve with traditional methods. This precision is crucial for predicting celestial events and understanding the behavior of astronomical bodies. (Chang, 2011)

In the realm of robotics, quaternions are employed for real-time motion planning and control. Robots often need to adjust their orientation quickly and accurately, and quaternions facilitate these adjustments by enabling efficient computation of rotational paths. This is particularly important in applications such as robotic arm control, where precise orientation is essential for tasks like assembly and manipulation.

In computer graphics, quaternions are pivotal for creating realistic animations and interactions. In film production, they are used to animate characters with natural and fluid movements. For instance, when animating a character performing a complex dance routine, quaternions ensure that each movement is smooth and realistic, from spins to leaps. Similarly, in video game development, quaternions are used to control camera movements, providing players with a seamless and immersive visual experience. The camera can smoothly transition between different angles and perspectives, enhancing the overall gameplay. (Shoemake, 1998)

Moreover, in virtual and augmented reality applications, quaternions play a crucial role in tracking and rendering. They are used in head-mounted displays to track the user's head movements accurately, ensuring that the virtual environment responds instantly to the user's orientation. This real-time tracking is essential for creating immersive experiences where the virtual world aligns perfectly with the user's movements. (Cutler & Eustice, 2010) These examples illustrate the versatility and importance of quaternions in modern computational methods, highlighting their ability to enhance both the efficiency and realism of simulations and visualizations across diverse fields.

# 4 CONCLUSIONS

This paper delves into the concept of quaternions and their pivotal role in representing three-dimensional rotations, offering a comprehensive comparison with traditional rotation matrices. We elucidate the derivation of quaternions from rotation matrices and highlight the advantages and disadvantages of both methods. Quaternions, with their efficiency and ability to avoid issues like gimbal lock, emerge as a superior choice for many applications. We introduce novel algorithms aimed at reducing the computational complexity of quaternion operations while maintaining numerical stability, which is crucial for accurate and reliable rotations in various computational tasks. Furthermore, we explore the extensive applications of quaternions in modern physics and computer graphics, demonstrating their indispensable role in simulations, animations, and virtual reality environments. These applications underscore the versatility and importance of quaternions in enhancing both the efficiency and realism of computational methods across diverse fields.

# REFERENCES

Shoemake, K. (1985). Animating rotation with quaternion curves. SIGGRAPH Computer Graphics, 19(3), 245-254.

Chang, D. E. (2011). Quaternion-based dynamics of rigid bodies. IEEE Transactions on Robotics, 27(5), 1012-1022.

Cutler, M., & Eustice, R. M. (2010). Quaternion-based pose estimation for augmented reality. IEEE Transactions on Visualization and Computer Graphics, 16(3), 491‑504.
Kuipers, J. B. (2002). Quaternions and rotation sequences: A primer with applications to orbits, aerospace, and virtual reality. Princeton University Press.
Shoemake, K. (1998). Quaternions. SIGGRAPH Course Notes.
Hanson, A. J. (2006). Visualizing quaternions. Morgan Kaufmann.
Griffiths, D. J. (2017). Introduction to quantum mechanics (3rd ed.). Cambridge University Press.
Fletcher, P. T., & Joshi, S. C. (2007). Riemannian geometry for the statistical analysis of diffusion tensor data. Signal Processing, 87(2), 250‑262.
Sarabandi, S., & Ha, J. (2018). Quaternion-based pose estimation for robotic applications. IEEE Robotics and Automation Letters, 3(3), 2012‑2019.
Kim, C. H., & Nam, K. W. (2004). Quaternion-based interpolation for computer animation. The Visual Computer, 20(6), 383‑393.