Quaternions: A Magical Journey to the Fourth Dimension

Linnea Caraballo [1] Tina Romansky [1]

[1]Sacred Heart University

Introduction

Quaternions are an extension of the complex number system that consist of one real part and three imaginary parts. This means that quaternions exist in the fourth dimension and they were first described by William Rowan Hamilton in 1843.

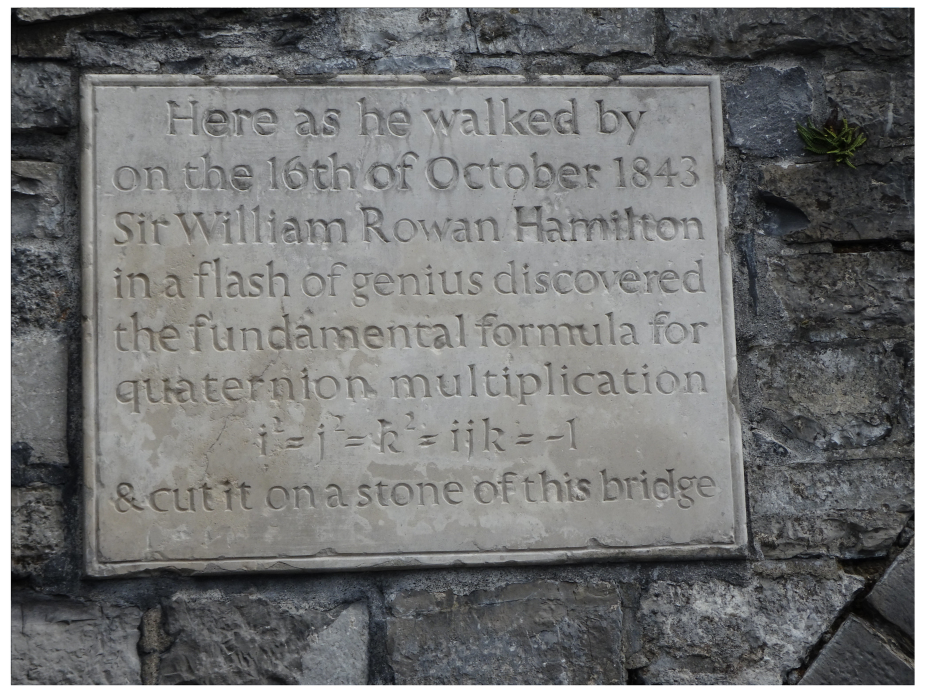

Figure 1. Picture of the Broom Bridge where Hamilton carved the fundamental formula for quaternion multiplication. [1]

A quaternion is a number of the form:

$$q = s + a\vec{i} + b\vec{j} + c\vec{k}$$

where $s, a, b, c \in \mathbb{R}$ and $\vec{i}, \vec{j}, \vec{k}$ are imaginary. We call s the **scalar** part and i, j, k the **vector** part. We denote the set of quaternions as $\mathbb{H}$.

The most common application for quaternions is to model the orientation and rotation of objects in 3D. While there are other methods to model this orientation and rotation of 3D objects, quaternions are the preferred option for multiple reasons. One of these reasons is the speed at which they are computed.

Why use Quaternions?

Other Methods:

- **Rotation Matrices:** Models the rotation and orientation of objects into a 3×3 matrix known as a rotation matrix. Rotation matrices are clunky and tedious to calculate requiring many different operations, but the main problem with rotation matrices is gimbal lock.
- **Rodrigues Rotation Formula:** Described by Benjamin Olinde Rodrigues three years prior to Hamilton, his paper provided a formula describing how one can use a single rotation about a third axis to represent two successive rotations about two different axes.

$$Rot(\vec{v}) = (1 - \cos(\theta))(\vec{u} \cdot \vec{v})\vec{u} + \cos(\theta)\vec{v} + \sin(\theta)(\vec{u} \times \vec{v})$$

The problem with Rodrigues Rotation formula lies in its complex computations. Imagine having to compute the $\sin$ and $\cos$ of something rotated by an angle $\theta = 0.236$. Doing this on top of having to perform cross and dot products make for long and complicated computation.

We can see that quaternions hold a lot of benefits over these two methods. The main benefits being computation speed and the prevention of gimbal lock.

Gimbal Lock

When looking at something in 3D we have 3-axis of rotation which we call **degrees of freedom**. Gimbal lock occurs when one degree of freedom is lost and cannot be regained.

Figure 2. Scan to see gimbal lock in action and learn more.

When using rotation matrices which utilize **Euler Angles**, there is no way to avoid gimbal lock. There are preventative measures and ways to mitigate its effect, but the only way to be one-hundred percent sure that gimbal lock does not occur is to move up a dimension. This is where quaternions and the fourth dimension come in.

Quaternions vs. Rotation Matrices

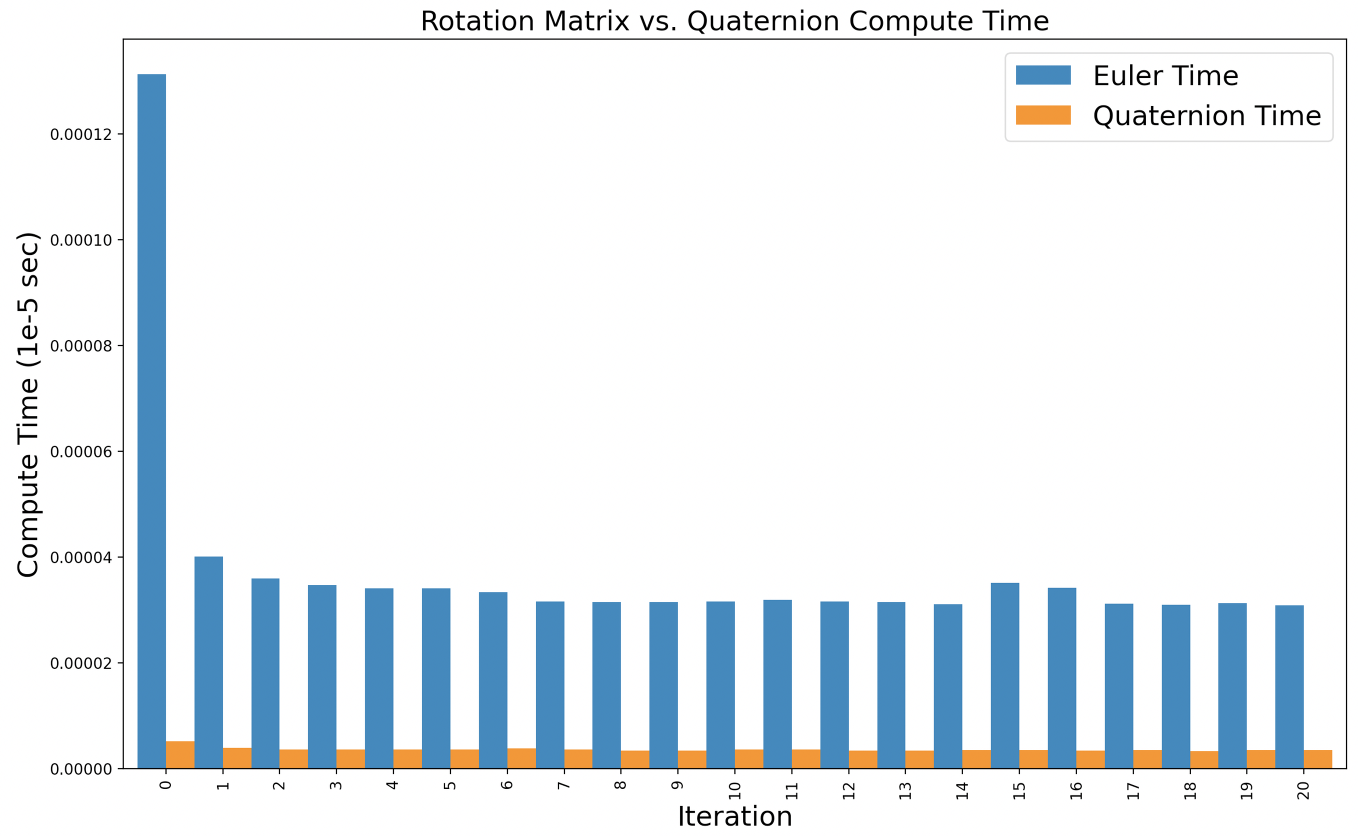

Figure 3. Graph to compare the compute time of Euler angles and quaternions. Each iteration takes random values and calculates the rotation with rotation matrices and quaternions timing the compute time.

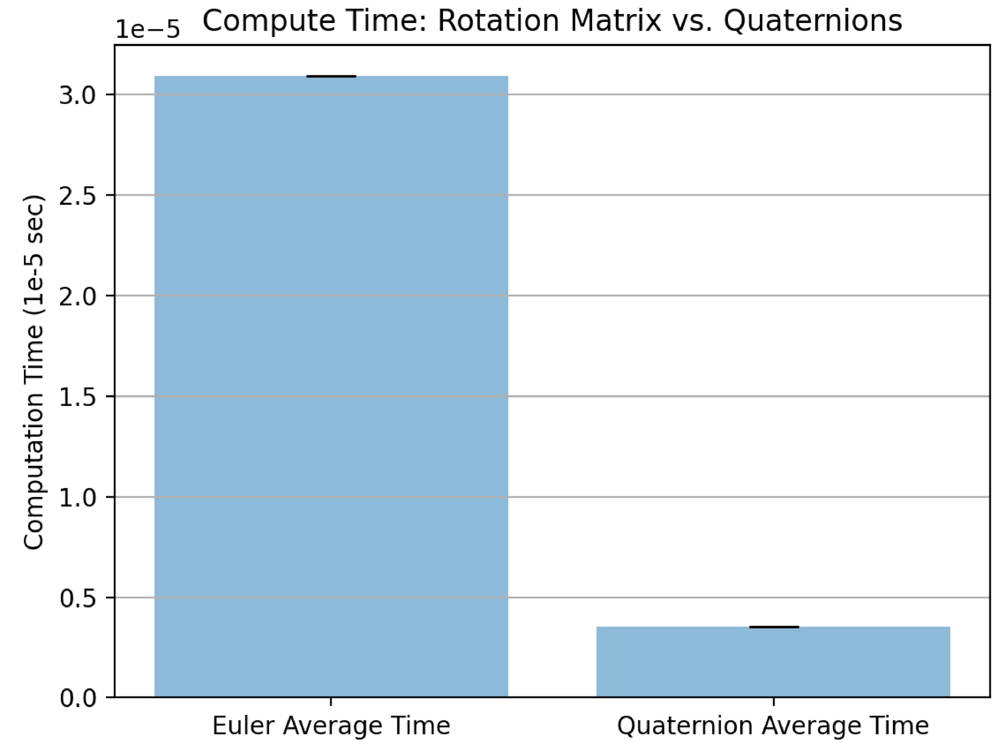

Figure 4. Graph to compare the average compute time of Euler angles and quaternions.

Quaternions vs. Rodrigues Rotation Formula

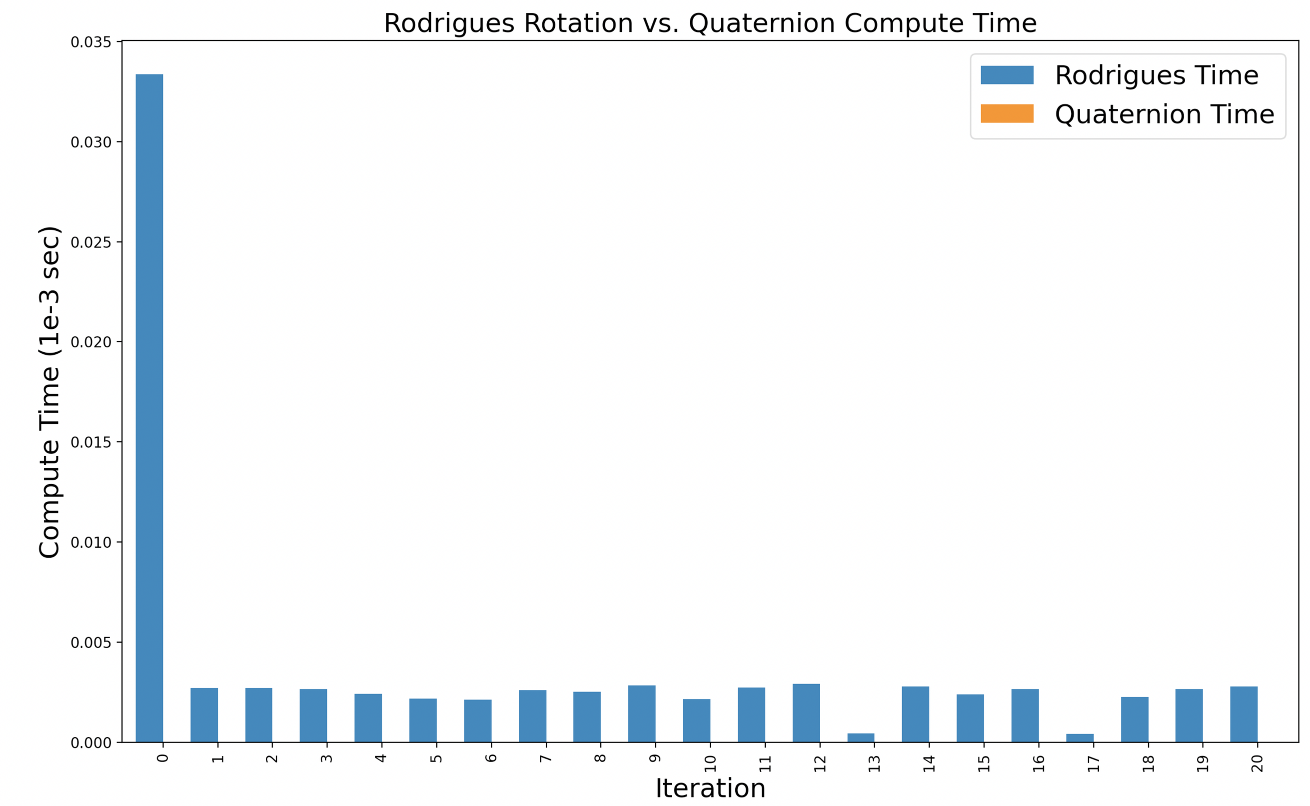

Figure 5. Graph to compare the compute time of Rodrigues rotation formula and quaternions. Each iteration takes random values and calculates the rotation with Rodrigues rotation formula and quaternions timing the compute time.

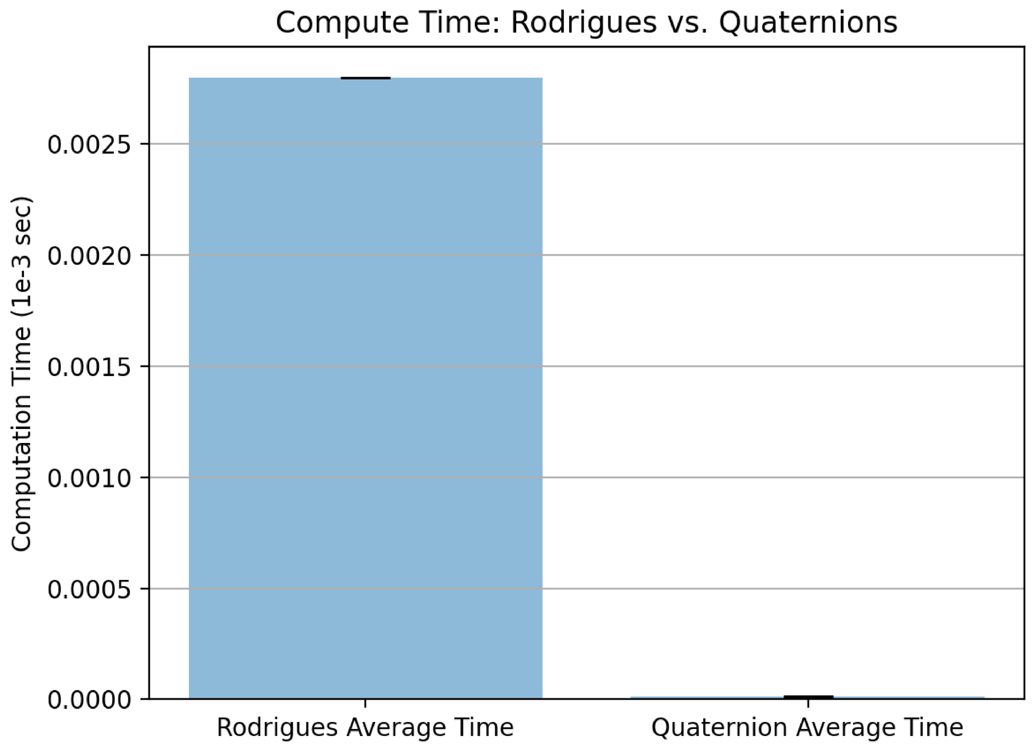

Figure 6. Graph to compare the average compute time of Rodrigues rotation formula and quaternions.

Why is a Faster Compute Time Important?

As we can see in Figure 3, 4, 5, 6, quaternions have a significantly faster compute time, sometimes so fast that they do not even appear on the graph. The main benefit of a faster compute time can be seen in animation and game development. A faster compute time means less time waiting for things to compile, less lag, and the benefit of not wasting time working on being resilient against gimbal lock or figuring out how to mitigate the effects of gimbal lock.

xkcd Comic

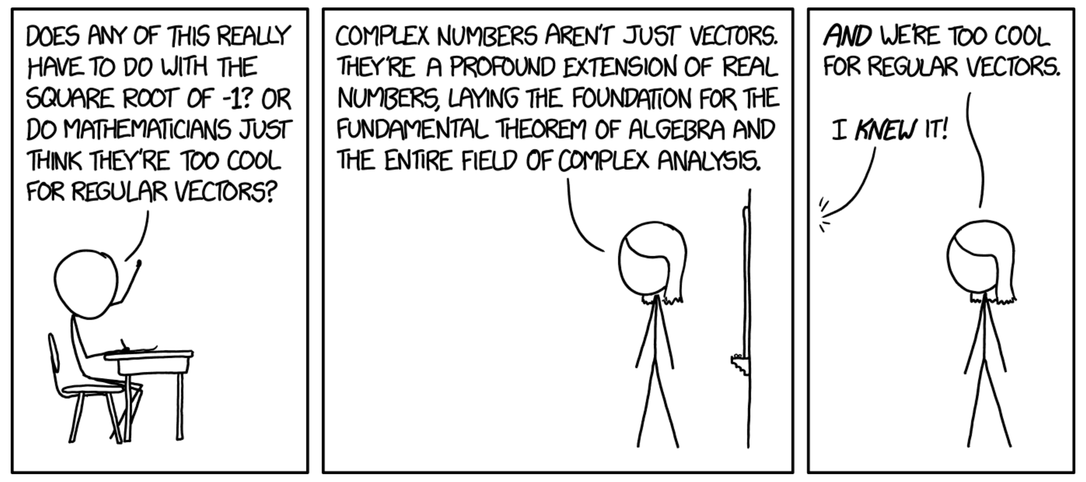

Figure 7. https://xkcd.com/2028/