The North Carolina School of
Science and Mathematics

CSC404

Second-Trimester Programming

The Quaternion Project

Author:
John Morrison

20 November 2013

In this article, we will construct a set $\mathbb{H}$ of mathematical objects called *quaternions*. The use of the letter $\mathbb{H}$ is in honor of Sir William Rowan Hamilton, an Irish mathematician who first described them. The quaternions form a special system of numbers that are especially useful in the domain of computer graphics.

1 1, i, j, and k.

A *quaternion* is defined to be a formal sum of the form

$$a + b\mathrm{i} + c\mathrm{j} + d\mathrm{k},$$

where a, b, c and d are real numbers. If a, b, c and d are all integers, we will say that we have an *integral quaternion*. The quantities 1, i, j, and k are all to be treated as unlike terms. In this way we define addition and subtraction of quaternions. To add or subtract we simply gather up like terms. Two quaternions are equal if and only if all of the coefficients of 1, i, j, and k match.

For example if $v = 3 + 2\mathrm{i} + \mathrm{j} - \mathrm{k}$ and $w = 4 + 2\mathrm{i} + 3\mathrm{j} + 5\mathrm{k}$, then we have

$$v + w = 7 + 4\mathrm{i} + 4\mathrm{j} + 4\mathrm{k}$$

and

$$v - w = -1 - 2\mathrm{j} - 6\mathrm{k}.$$

If λ is a real number an $v = a + b\mathrm{i} + c\mathrm{j} + d\mathrm{k}$, is a quaternion, we define *scalar multiplication* as follows

$$\lambda \cdot v = \lambda a + \lambda b\mathrm{i} + \lambda c\mathrm{j} + \lambda d\mathrm{k}.$$

To wit, the λ is distributed over all of the terms in the quaternion.

So far, quaternions have the structure of four-dimensional vectors. We will define the quaternion 0 to be $0+0\mathrm{i}+0\mathrm{j}+0\mathrm{k}$. Now we shall round up some basic facts about this vector characterization of quaternions.

Theorem 1. *Let λ and μ be scalars and u_0, u_1 and u_2 be quaternions. Then we have the following.*

i. $u_0 + u_1 = u_1 + u_0$

ii $0 + u_0 = u_0$

iii. $u_0 + (u_1 + u_2) = (u_0 + u_1) + u_2$

iv. $\lambda \cdot (u_0 + u_1) = \lambda \cdot u_0 + \lambda \cdot u_1$

v. $(\lambda + \mu_0) \cdot u_0 = \lambda \cdot u_0 + \mu \cdot u$

Proof. (i.) Write $u_0 = a_0 + b_0\mathrm{i} + \mathrm{c}_0\mathrm{j} + \mathrm{d}_0\mathrm{k}$ and $u_1 = a_1 + b_1\mathrm{i} + \mathrm{c}_1\mathrm{j} + \mathrm{d}_1\mathrm{k}$. Then, using the commutivity of the real numbers,

$$\begin{aligned} u_0 + u_1 &= (a_0 + b_0\mathrm{i} + \mathrm{c}_0\mathrm{j} + \mathrm{d}_0\mathrm{k}) + (\mathrm{u}_1 = \mathrm{a}_1 + \mathrm{b}_1\mathrm{i} + \mathrm{c}_1\mathrm{j} + \mathrm{d}_1\mathrm{k}) \\ &= (a_0 + a + 1) + (b_0 + b_1)\cdot\mathrm{i} + (\mathrm{c}_0 + \mathrm{c}_1)\cdot\mathrm{j} + (\mathrm{d}_0 + \mathrm{d}_1)\cdot\mathrm{k} \\ &= (a_1 + a + 0) + (b_1 + b_0)\cdot\mathrm{i} + (\mathrm{c}_1 + \mathrm{c}_0)\cdot\mathrm{j} + (\mathrm{d}_1 + \mathrm{d}_0)\cdot\mathrm{k} \\ &= u_1 + u_0. \end{aligned}$$

You can deal similarly with the other parts of this theorem; they are left as an exercise. □

Exercises

1. Prove the rest of Theorem 1.
2. Create a class in Java called `Quaternion`. Here is a shell for it.

```
public class Quaternion
{
    //coefficients for a + bi + cj + dk
    private double a;
    private double b;
    private double c;
    private double d;
    public Quaternion(double _a, double _b,
                      double _c, double _d)
    {
        //code goes here
    }
    public Quaternion()
    {
        //default is the zero quaternion
    }
    public Quaternion(double x)
    {
        //promote x from a real number to a quaterion.
    }
    //return this + other
    public Quaterion add(Quaternion other)
    {
        return null;
    }
    //return this - other
    public Quaterion subtract(Quaternion other)
```

```
    {
        return null;
    }
    //do scalar multiplication here.
    public Quaternion multiply(double lambda)
    {
        return null;
    }
    public static void main(String[] args)
    {
        //put test code here to test your class.
    }
}
```

2 Multiplication

We shall now endow quaternions with multiplication. So far, they only have the structure of scalar multiplication. Now we will have *vector* multiplication; the product of two quaternions will be a quaternion.

We begin by defining multiplication for the *basis elements* 1, i, j and k. Multiplication for these is summarized in the multiplication table below.

$\cdot$	1	i	j	k
1	1	i	j	k
i	i	-1	k	$-$j
j	j	$-$k	-1	i
k	k	j	$-$i	-1

A Useful Mnemonic You might ask, "How do I remember all of this stuff?" You don't. Use this mnemonic. Take out a pencil and a piece of paper. Make i, j and k the vertices of an equilateral triangle, arranging them clockwise. Suppose you want to multiply i and j. Go from i to j clockwise, then arrive at k. This give $\mathrm{i} \cdot \mathrm{j} = \mathrm{k}$. If you go counterclockwise, put in a factor of -1. For example, for $\mathrm{k} \cdot \mathrm{j}$, you go from k to j counterclockwise and arrive at i. Therefore the product is $-$i. All three of i, j and k square to -1. You should draw the picture, and complete the table shown here to test out the mnemonic.

We now bootstrap this into a general definition of multiplication. We define multiplication of arbitrary quaternions by the following procedure. First use the FOIL rule to multiply out all of the terms. Then use the basis element multiplication table to do the basis element multiplications. We finish by collecting all of the like terms. Voila! We now know how to multiply. You can see some things right away.

First of all, if λ is a scalar and v is any quaternion,

$$\lambda \cdot v = (\lambda + 0\mathrm{i} + 0\mathrm{j} + 0\mathrm{k}) \cdot \mathrm{v.v} \cdot (\lambda + 0\mathrm{i} + 0\mathrm{j} + 0\mathrm{k}) =$$

You can check this out very easily for yourself.

You can also see some other things. For example, any quaternion whose i, j and k terms are all zero basically acts like a scalar; when you multiply a quaternion by one of them, it is scalar multiplication. Also, multiplication of these quaternions is commutative. It behaves like a copy of the real numbers embedded inside of the quaternions.

As an example, let us compute ijk. We have

$$(\mathrm{i} \cdot \mathrm{j}) \cdot \mathrm{k} = \mathrm{k} \cdot \mathrm{k} = -1.$$

Since $\mathrm{ij} \neq \mathrm{ji}$, we know this form of multiplication will not be commutative. It is associative; this can be shown by grinding out the computation. A question you might be tempted to ask is *Is division possible?* We shall answer that question by proceeding down a slight detour.

We shall first define a little terminology. If $z = a + b\mathrm{i} + \mathrm{cj} + \mathrm{dk}$ is a quaternion, we say that a is the *scalar part* of z and $b\mathrm{i} + \mathrm{cj} + \mathrm{dk}$ is the *vector part* of z. We define the *conjugate* of z by

$$\overline{z} = a - b\mathrm{i} - \mathrm{cj} - \mathrm{dk};$$

to obtain this we just change the sign of the vector part of z. Observe that the *conjugation function* $z \mapsto \overline{z}$ is an *involution* on the quaternions, i. e. it is its own inverse.

Now we are going to perform a very interesting multiplication. Notice how we get the multiplications of i, j and k into alphabetical order so cancellations will stand out.

$$\begin{aligned}
z\overline{z} &= (a + b\mathrm{i} + \mathrm{cj} + \mathrm{dk}) \cdot (\mathrm{a} - \mathrm{bi} - \mathrm{cj} - \mathrm{dk}) \\
&= a^2 + ab\mathrm{i} + \mathrm{acj} + \mathrm{adk} - \mathrm{abi} - \mathrm{b}^2\mathrm{i}^2 - \mathrm{bcji} - \mathrm{bdki} \\
&\quad -ac\mathrm{j} - \mathrm{bcij} - \mathrm{c}^2\mathrm{j}^2 - \mathrm{cdkj} - \mathrm{adk} - \mathrm{bdik} - \mathrm{cdjk} - \mathrm{d}^2\mathrm{k}^2 \\
&= a^2 + ab\mathrm{i} + \mathrm{acj} + \mathrm{adk} - \mathrm{abi} + \mathrm{b}^2 + \mathrm{bck} - \mathrm{bdj} \\
&\quad -ac\mathrm{j} - \mathrm{bck} + \mathrm{c}^2 + \mathrm{cdi} - \mathrm{adk} + \mathrm{bdj} - \mathrm{cdi} + \mathrm{d}^2 \\
&= a^2 + b^2 + c^2 + d^2.
\end{aligned}$$

We now define the *norm* of a quaternion $z = a + b\mathrm{i} + \mathrm{cj} + \mathrm{dk}$ to be

$$\|z\| = \sqrt{a^2 + b^2 + c^2 + d^2}.$$

We immediately see that for any quaternion z, $z\overline{z} = \|z\|^2$. Notice that in the previous calculation, we also have $\overline{z}z = \|z\|^2$ as well. A quaternion and its conjugate commute under multiplication.

Now let us ask a question. Given a nonzero quaternion z, can we find a quaternion w so that $z \cdot w = 1$? Let us suppose we can. Were this so, we would have $\overline{z}zw = \overline{z}$, so $\|z\|^2 w = \overline{z}$. Therefore, we must have

$$w = \frac{\overline{z}}{\|z\|^2};$$

this division is possible because we are just scalar multiplying by the number $1/\|z\|^2$.

Theorem 2. *Every nonzero quaternion has a two-sided multiplictative inverse. To wit, for each quaternion z there is a quaternion w so that $wz = zw = 1$. In fact, we know that*

$$w = \frac{\overline{z}}{\|z\|^2}.$$

The proof of this theorem is now an easy exercise. Now we see how to divide; we just multiply by the multiplicative inverse we have seen here. One consequence of our development is that this multiplicative inverse is unique. We will henceforth use the notation z^{-1} for the multiplicative inverse of a quaternion z.

Exercises

1. Grind it out: show that multiplication is associative.
2. Let us define
$$V = \{a + b \cdot \mathrm{i} | \mathrm{a}, \mathrm{b} \in \mathbb{R}\}.$$
Show that if $u, v \in V$ that $u + v \in V$ and $u \cdot v \in V$. Show that $u \cdot v = v \cdot u$.
3. Show that if z and w are quaternions, that
$$\|zw\| = \|z\|\|w\|.$$
4. Now add these methods to your `Quaternion` class. Method stubs have been supplied so all will compile.

```
Quaternion multiply(Quaternion other)
{
    return null;
}
Quaternion divide(Quaternion other)
{
    return null;
}
```

```
Quaternion conjugate()
{
    return null;
}
//handle the case for n < 0 as well by
//exponentiating the inverse.  Use divide
//and conquer recursion to make this efficient.
Quaternion pow(int n)
{
    return null;
}
double norm()
{
    return 0.0;
}
```

3 The Product

You are to produce the quaternion class. Doing the exercises will help you to understand quaternions. You do not need to turn these in and you can ask to see them worked in class. Name your file `Quaternion.java`. Also, give it static constants `Quaterniion.ZERO`, `Quaternion.ONE`, `Quaternion.I`, `Quaternion.J`, and `Quaternion.K`. This project will be graded on a five-point scale. You should build some test cases and place them in a `main` method. Work out some cases on paper and test them.
You will learn about Javadoc during the case study. You are expected to Javadoc your Quaternion class, so that others can use it without having to read your actual code. When you turn it in, do not submit the files generated by Javadoc; we can generate them locally.

4 References

Why quaternions? They are a significant tool if you are interested in 3D computer graphics. See this Stack Overflow post.

`http://stackoverflow.com/questions/10926546/rotation-vectors-vs-quaternions`

This article in a game developer forum can yield some insight into their efficiency in handling rotation about an axis.

`http://www.gamedev.net/page/resources/_/technical/math-and-physics/do-we-really-need-quaternions-r1199`

Several references on quaternions can be found in this Math Stackexchange post.

http://math.stackexchange.com/questions/71/real-world-uses-of-quaternions