

Normed division algebras with R: introducing the onion package

Robin K. S. Hankin
Auckland University of Technology

Abstract

This vignette is based on [Hankin \(2006\)](#) and corrects an algebraic error therein.

This vignette introduces the **onion** package of R routines, for manipulation of quaternions and octonions. A simple application of the quaternions in use is given, using the package.

Keywords: Onion, quaternions, octonions, R.

1. Preface

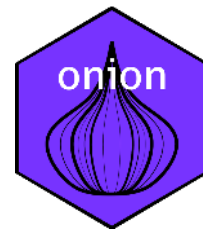
An *algebra* is a vector space V over a field (here the real numbers) in which the vectors may be multiplied. In addition to the usual vector space axioms, one requires, for any $\lambda, \mu \in \mathbb{R}$ and $\mathbf{x}, \mathbf{y}, \mathbf{z} \in V$:

- $\mathbf{x}(\lambda\mathbf{y} + \mu\mathbf{z}) = \lambda\mathbf{x}\mathbf{y} + \mu\mathbf{x}\mathbf{z}$
- $(\lambda\mathbf{y} + \mu\mathbf{z})\mathbf{x} = \lambda\mathbf{y}\mathbf{x} + \mu\mathbf{z}\mathbf{x}$

where multiplication is denoted by juxtaposition; note the absence of associativity. A *division algebra* is a nontrivial algebra in which division is possible: for any $\mathbf{a} \in V$ and any nonzero $\mathbf{b} \in V$, there exists precisely one element \mathbf{x} with $\mathbf{a} = \mathbf{b}\mathbf{x}$ and precisely one element \mathbf{y} with $\mathbf{a} = \mathbf{y}\mathbf{b}$. A *normed division algebra* is a division algebra with a norm $\|\cdot\|$ satisfying $\|\mathbf{x}\mathbf{y}\| = \|\mathbf{x}\| \|\mathbf{y}\|$.

There are precisely four normed division algebras: the reals themselves (\mathbb{R}), the complex numbers (\mathbb{C}), the quaternions (\mathbb{H}) and the octonions (\mathbb{O}); the generic term is “onion”, although the term includes other algebras such as the sedenions.

The R computer language ([R Development Core Team 2008](#)) is well-equipped to deal with the first two: here, I introduce the **onion** package that provides some functionality for the quaternions and the octonions, and illustrate these interesting algebras using numerical examples.



2. Introduction

Historically, the complex numbers arose from a number of independent lines of inquiry and our current understanding of them (*viz* $z = x + iy$; the Argand plane) developed over the eighteenth and nineteenth centuries.

Hamilton was one of many mathematicians to attempt to extend the complex numbers to a third dimension and discover what would in our terminology be a three dimensional normed division algebra. We now know that no such thing exists: division algebras all have dimension 2^n for n some non-negative integer.

Hamilton came upon the multiplicative structure of the quaternions in a now-famous flash of inspiration on 16th October 1843: the quaternions are obtained by adding the elements i , j , and k to the real numbers and requiring that

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

A general quaternion is thus written $a+bi+cj+dk$ with a, b, c, d being real numbers; complex arithmetic is recovered if $c = d = 0$. Hamilton's relations above, together with distributivity and associativity, yield the full multiplication table and the quaternions are the unique four dimensional normed division algebra.

However, Hamilton's scheme was controversial as his multiplication was noncommutative: a shocking suggestion at the time. Today we recognize many more noncommutative operations (such as matrix multiplication), but even Hamilton had difficulty convincing his contemporaries that such operations were consistent, let alone worth studying.

2.1. The octonions

The fourth and final normed division algebra is that of the octonions. These were discovered around 1844 and are an eight-dimensional algebra over the reals. The full multiplication table is given by [Baez \(2001\)](#).

3. Package onion in use

A good place to start is function `rquat()`, which returns a quaternionic vector of a specified length, whose elements are random small integers:

```
> x <- rquat(5)
> names(x) <- letters[1:5]
> print(x)
```

	a	b	c	d	e
Re	-1.42	0.40	0.15	0.50	0.29
i	-2.15	0.59	1.81	1.23	-0.42
j	-0.18	0.11	-0.27	-1.09	0.18
k	0.63	-0.60	0.50	-0.37	-0.25

This quaternionic vector, of length 5, is of the form of a four-row matrix. The rownames are the standard basis for quaternions, and their entries may be manipulated as expected; for example, we may set component k to be component j plus 10:

```
> k(x) <- j(x)+10
> x
```

```

      a      b      c      d      e
Re -1.42  0.40  0.15  0.5  0.29
i  -2.15  0.59  1.81  1.2 -0.42
j  -0.18  0.11 -0.27 -1.1  0.18
k   9.82 10.11  9.73  8.9 10.18

```

Quaternionic vectors behave largely as one might expect. For example, we may concatenate `a` with a basis quaternion (the four bases are represented in the package by `H1`, `Hi`, `Hj`, and `Hk`) and multiply the result by the first element:

```
> c(x,Hk)*x[1]
```

```

      a      b      c      d      e
Re -98.98 -98.5 -91.9 -86 -101.2 -9.82
i    6.08  1.2  -3.7 -12   3.7  0.18
j    0.52 -27.7 -38.3 -30  -18.1 -2.15
k   -27.80 -10.2 -13.2 -10  -11.1 -1.42

```

And indeed we may explicitly verify that quaternionic multiplication is not commutative using the `commutator()` function, returning $xy - yx$:

```
> y <- rquat(5)
> commutator(x,y)
```

```

      a      b      c      d      e
Re  0.0  0.0  0.0  0.00  0.0
i  -16.2 -21.4 -5.8 -1.23 -26.7
j  -31.3 -47.0  8.2  4.44 -10.8
k   -4.1  1.8  1.3  0.71  -0.9

```

It is possible to verify that quaternionic multiplication is associative using function `associator(x,y,z)` which returns $(xy)z - x(yz)$ and is thus identically zero for associative operators:

```
> associator(x,y,rquat(5))
```

```

      a      b      c      d      e
Re 2.2e-16 2.7e-15 -8.9e-16 4.4e-16 -3.6e-15
i  0.0e+00 -7.1e-15 0.0e+00 1.8e-15 3.6e-15
j  7.1e-15 3.6e-15 8.9e-16 0.0e+00 -1.8e-15
k  0.0e+00 -7.1e-15 0.0e+00 -2.2e-16 1.8e-15

```

Compare the octonions, which are not associative:

```
> associator(roct(3),roct(3),roct(3))
```

```

      [1]      [2]      [3]
Re  1.8e-15  4.4e-16  1.3e-15
i   1.3e+00  2.8e+00 -5.6e-01
j   9.5e+00 -4.3e+00 -2.6e+00
k  -4.1e+00 -1.2e+00 -5.4e+00
l  -2.0e+01  9.7e+00  5.6e-01
il  2.8e+00  2.7e+00 -1.6e+01
jl  1.8e+00  2.1e+00 -1.6e+01
kl -1.2e+01  4.2e+00  7.9e+00

```

Many transcendental functions operate on onions, via a reasonably straightforward generalization of the complex case. Consider the square root, defined as $\exp(\log(x)/2)$. That this obeys similar rules to the usual square root may be demonstrated for octonions by calculating $\sqrt{x}\sqrt{x} - x$, and showing that its modulus is small:

```

> x <- roct(3)
> Mod(sqrt(x)*sqrt(x)-x)

[1] 3.8e-16 2.5e-16 7.0e-16

```

showing acceptable accuracy in this context. However, many identities that are true for the real or complex case fail for quaternions or octonions; for example, although $\log(x^2) = 2\log(x)$, it is not generally the case that $\log(xy) = \log(x) + \log(y)$, as we may verify numerically:

```

> x <- rquat(3)
> y <- rquat(3)
> Mod(log(x*x)-2*log(x))

[1] 2.4e-16 4.5e-16 6.3e+00

> Mod(log(x*y)-log(x)-log(y))

[1] 0.28 0.57 2.84

```

3.1. Practical applications

I now show the quaternions in use: they can be used to rotate an object in 3D space in an elegant and natural way. If we wish to rotate vector \bar{v} , considered to be triple of real numbers, we first identify its 3 components with the imaginary part of a quaternion with zero real part (ie $\mathbf{v} = 0 + \bar{v}_1i + \bar{v}_2j + \bar{v}_3k$). Then the transformation defined by

$$\mathbf{v} \longrightarrow \mathbf{v}' = \mathbf{z}\mathbf{v}\mathbf{z}^{-1}$$

where $\mathbf{z} \in \mathbb{H}$, is a rotation¹. Note that the noncommutativity of the quaternions means that the mapping is not necessarily the identity. This scheme may be used to produce figure 1.

¹The real part of \mathbf{v}' is again zero and thus may be interpreted as a three-vector. It is straightforward to prove that $\bar{v} \cdot \bar{w} = \overline{v' \cdot w'}$. In R idiom, one would numerically verify these identities by evaluating $\mathbf{d}=(\mathbf{z}*\mathbf{v}/\mathbf{z})\%.\%(\mathbf{z}*\mathbf{w}/\mathbf{z}) - \mathbf{v} \%.\% \mathbf{w}$ (where $\mathbf{v}, \mathbf{w}, \mathbf{z}$ are quaternions) and observing that $\text{Mod}(\mathbf{d})$ is small.

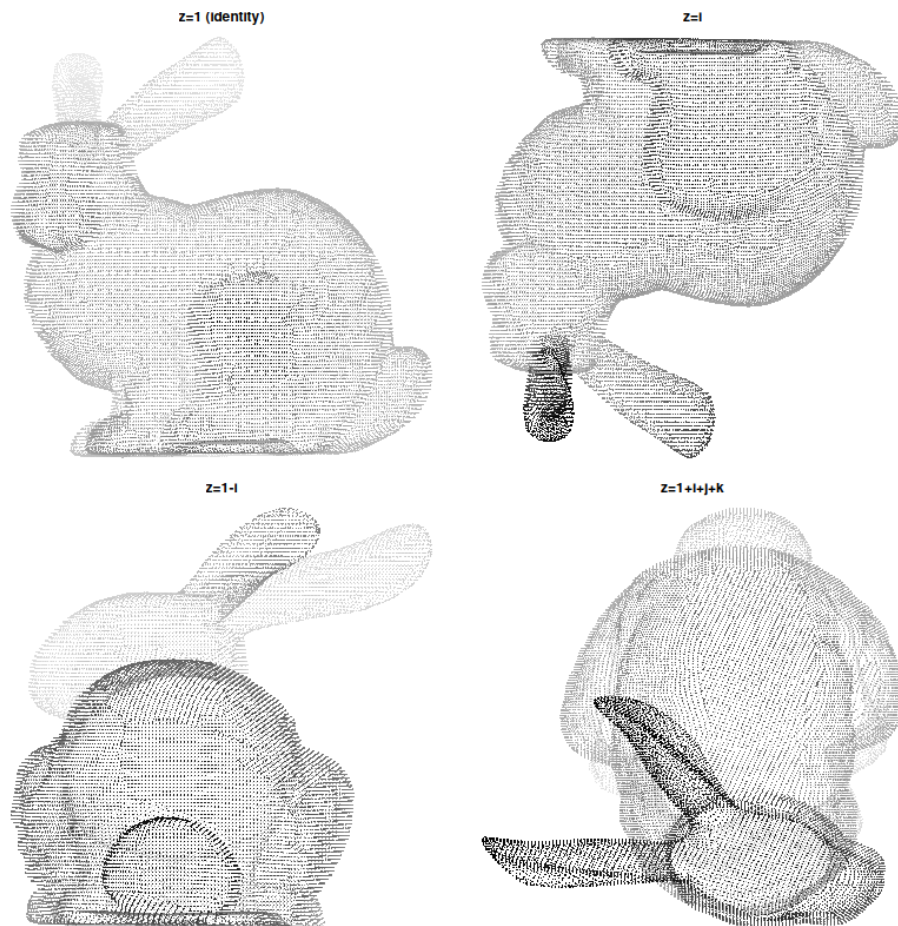


Figure 1: The Stanford Bunny (Turk 2005) in various orientations, plotted using `p3d()`; in the package, 3D rotation is carried out by function `rotate()`. Depth cue is via progressive greying out of points further from the eye, as though viewed through a mist

Translating between quaternionic rotation and (say) the equivalent orthogonal matrix is straightforward.

4. Conclusions

Quaternions and octonions are interesting and instructive examples of normed division algebras. Quaternions are a natural and efficient representation of three dimensional rotations and this paper includes a brief illustration of their use in this context.

Octonions' applicability to physics is currently unclear, but a steady trickle of papers has appeared in which the octonions are shown to be related to spinor geometry and quantum mechanics (see [Baez \(2001\)](#) and references therein). Reading the literature, one cannot help feeling that octonions will eventually reveal some deep physical truth. When this happens, the **onion** package ensures that R will be ready to play its part.

Acknowledgments

I would like to acknowledge the many stimulating and helpful comments made by the R-help list over the years.

References

- Baez JC (2001). "The octonions." *Bulletin of the American Mathematical Society*, **39**(5), 145–205.
- Hankin RKS (2006). "Normed division algebras with R: Introducing the **onion** package." *R News*, **6**(2), 49–52. URL <https://CRAN.R-project.org/doc/Rnews/>.
- R Development Core Team (2008). *R: A Language and Environment for Statistical Computing*. R Foundation for Statistical Computing, Vienna, Austria. ISBN 3-900051-07-0, URL <https://www.R-project.org>.
- Turk G (2005). "The Stanford Bunny." Stanford University Computer Graphics Laboratory, URL <https://graphics.stanford.edu/data/3Dscanrep/>.

Affiliation:

Robin K. S. Hankin
Auckland University of Technology
New Zealand
E-mail: hankin.robin@gmail.com