D: The Infinite Square Roots of –1

Bernard M Diaz[1] & Peter Rowlands[2]

[1] Department of Computer Science
[2] Science Communication Unit, Department of Physics
The University of Liverpool
Peach Street, Liverpool, UK, L69 7ZF

E-mail: b.m.diaz@liv.ac.uk, p.rowlands@liv.ac.uk,

Abstract
We present D, a symbol that can be used in the universal alphabet that provides a computational path to the nilpotent Dirac equation (Diaz & Rowlands, 2004) and which results in a tractable computer representation of the infinite square roots of –1. We outline how the representation is derived, the properties of the representation, and how the form can be used. Think of D as an infinite table of 1's in any representation e.g. binary or hexadecimal. Any specified column Di of the table has the property that when multiplied with a row Di , the result is a representation of –1. Di multiplied with Dj anticommutes as – (Dj*Di) and produces Dk in a way identical to Hamilton's quaternion i, j, and k. With an infinite and uniquely identifiable set of such triad forms D can be considered both a symbol and because of this behaviour, an alphabet.

Keywords: Quaternion, complex number, rewrite system, universal alphabet, nilpotent Dirac equation.

1 Introduction

For a universal rewrite system of the type suggested by Rowlands and Diaz (2002) there is the need to determine the nature and symbols of the alphabet generated at the complexification stage in the iterative rewrite sequence (Diaz & Rowlands, 2004).

Recall that we have a zero sum result at all times. Using the iterative explanation, we use the operators "create" and "conserve" to generate then check for new symbols in the evolving alphabet. We start with the symbol and an alphabet consisting of just 0. In a first step we "create" the symbol 1 and to return to the zero sum we generate its conjugate (in the "conserve" process), the symbol –1. From –1, we can generate the infinite square roots which when considered with their conjugate forms, e.g. ±*i,* ±*j*, etc, and with anti-commutativity (Rowlands, 2003) drives the identification of further new alphabet symbols. The details and further algebra of this process need not concern us (they may be found in Rowlands, 2003), however, there is a need to represent in a computationally tractable form these infinite square roots such that they avoid the requirement for an infinite symbol sequence: i, j, k, ….

In this paper we suggest that an acceptable solution that captures the notion of an infinite sequence of the square roots to –1 as required is *D*, where the symbol chosen is arbitrary and represents all the infinite square roots.

We begin (Section 2) by establishing that there does exist an infinite number of square roots of –1 (Ell & Sangwine, 2005). Although several routes to a proof exist, most depend on the observation that the set of quaternions that square to –1 is the infinite set of vectors of absolute value 1 (Kuipers, 2002). This leads to a discussion (Section 3) of the computational methods of representing –1 (Booth, 1951) and the observation that this maps through quaternion tesseral addressing to a specific tile, and in the limit, to a point in space (Bell & Mason, 1990). The geometric interpretation of this, for example that a multiplication of address labels using tesseral methods can be generated and can be set to corresponds to rotation and scaling of the tile point set, follows from this observation (Bell et. al.,1983). To establish the representation (Section 4) we then borrow notions of infinite series of digits from other number representation domains (e.g. p-adic representation as in Scott, 1985) and create a bracketed notation to simplify our handling of the infinite sequence. Using this and the computational and tesseral representation explanations we construct a meaning for the *D* symbol chosen to represent the infinite square roots. The paper concludes with details of some of the properties of the representation (Section 5) and an explanation of how the notation (Section 6) can be exploited in an explanation of the complexification stage used to establish the rewrite route to the nilpotent Dirac equation (Rowlands, 2003).

2 Infinite Solutions of $x^2 = -1$

The demonstration of an infinite number of solutions is a corollary of the lemma that the square of any unit vector is –1 (Ell & Sangwine, 2005). If μ is an arbitrary unit vector defined in terms of a Cartesian representation, its square is given by $\mu =(ix + jy + kz)/b$ where b is $\surd(x^2 + y^2 + z^2)$ and x, y, z are real and i, j, and k are mutually perpendicular unit vectors that follow the rules defined by Hamilton in 1843 (Hamilton, 1899). The square is:

$$\begin{aligned}\mu^2 &= (ix + jy + ky)^2 / (x^2 + y^2 + z^2)\\ &= (i^2x^2 + j^2y^2 + k^2z^2 + ijxy + jixy + ikxy + kixz + jkyz + kjyz) / (x^2 + y^2 + z^2) \quad (1)\end{aligned}$$

and if we substitute Hamilton's rules*:* $i^2 = j^2 = k^2 = ijk = -1$ we get:

$$\begin{aligned}&= (-x^2 - y^2 - z^2)/(x^2 + y^2 + z^2)\\ &= -1. \qquad (2)\end{aligned}$$

As there are an infinite number of unit vectors there are an infinite number of unique solutions to the equation. All of these will follow the anticommutative Hamilton rules in a cyclical way as for example:

$$ij = -ji \textit{ and } jk = -kj \text{ etc.} \qquad (3)$$

The infinite solutions can also be shown to be a consequence of considering Euler's formula: $e^{ai} = \cos a + i\text{sine } a,$ for points on a unit circle where each circle is in the plane of one of the infinite planes that constitute the unit sphere in quaternion space (Kuipers, 2002).

3 Representation of –1

In the now standard Booth (1951) representation of negative integers in computers, wordlength is fixed and using a twos complement binary form (Scott, 1985 Ch. 4) every bit combination of wordlength is used, approximately half yielding positive and half negative integers. In such computational notations –1 is a sequence, wordlength long, of repeated 1's.

For general use, we can define an infinite series of 1's to represent –1, which we present as:

| 1 1 1 1 1 1 1 1 1)

where |, is repetition to the left infinitely and) repetition of 1 infinitely to the right. One is reminded of p-adic numbers (Koblitz, 1977; Scott, 1985 Ch. 7.7), infinite tesseral addresses (Diaz & Bell, 1986), and the bra ket notation (Griffiths, 2004). For this paper there is no need to invoke the convention that the repeating digit sequences are identified with an overline as only one digit repeats here.

An alternative method of establishing the same representation based on the description of tesseral quaternions (Bell & Mason, 1990) is to consider quaternion space halved at each division by four orthogonal hyperplanes. This process gives rise to an origin, orthogonal axes, and 16 equal divisions of the 4D space. If we label each axis division either side of the origin with a binary 0 and 1, then each division of space will have a 4 digit binary address 0000 to 1111. To distinguish these we can label them using hexadecimal digits 0-F. If we take the hexadecimal 1 division (binary labelled 0001), and repeat the space sub-division process hierarchically we generate a tiling with tesseracts, where one such space (point), in the limit, has the infinite hexadecimal 1 address that can be labelled in a number of ways, including | 1 1 1 1) as above or simply just |1). Providing we retain exactly the same algorithm in dividing the space and numbering sub-divisions, the hexadecimal label |1) will be adjacent to |0) and in the limit will also be adjacent to |2), |3), …, |F). An arithmetic based on the quaternions can be generated for these addresses where addition corresponds to translation through the 4D space and multiplication to scaling and rotation.

It should be noted that if we restrict the representation to 3D, the labelling is exactly that of the computer space storage structure known as an octtree (Bell & Mason, 1990) which is much used in computer graphics and image processing as a spatial data storage structure (e.g. Gargantini, 1982, Samet, 1984; Navazo et. al., 1986, etc).

4 Generating a Square Root for –1

Given the notation established above, and using * to indicate multiplication, what is the representation of the square root? We –seek to find *D* where:

$$D * D \; = \; -1 \tag{4}$$

Or more specifically given the notation from Section 3:

$$D * D = |\,1\,) \tag{5}$$

If we assume that the square of –1 is 1, then an answer might be:

$$|{-1})\, *\, |{-1}) \; = \; |1) \tag{6}$$

or in a more expanded representational form:

$$|{-1}\; {-1}\; {-1}\; {-1}\; {-1}\,) \;*\; |\,{-1}\; {-1}\; {-1}\; {-1}\; {-1}\,) \;\; = \;\; |\,1\;1\;1\;1\;1\,) \tag{7}$$

where we take each –1 in turn from one representation and multiply it with its corresponding element in the second representation. Although similar to vector and tensor representation, the infinite nature of the process renders it tangibly different

We can achieve a more intuitive explanation by noting that a simple resolution of a representation for the form |–1) is one in which each –1 in the row representation shown above is written as a |1) in columnar form, that is each –1 in the row representation above is replaced by a column |1), resulting in an infinite table of 1s:

$$\begin{array}{lcl} \;\text{– – – – –} & & \\ |\,1\;1\;1\;1\;1\,) & & \;\_ \\ |\,1\;1\;1\;1\;1\,) & = & |\,1\,) \;=\; D \\ |\,1\;1\;1\;1\;1\,) & & \;\smile \\ \;\smile\smile\smile\smile\smile & & \end{array} \tag{8}$$

which we read as an infinite series of column and row 1's and define generically, simply as D, the closest symbol to just the bounding brackets of the infinite table. In this generic form we can more closely imitate row/column vector behaviour by considering a column of –1's multiplied by a row of –1's each taken one at a time, to yield |1).

Although similar in arrangement to an infinite matrix, or tensor, none of the mathematical properties of these forms should be assumed. This representation is simply tabular, with the table interpreted in terms of the rows and columns. Each row and column can be numbered to identify specific rows and/or columns uniquely in an enumerated form of the representation.

5 Infinite Roots

We can define a property of each enumerated row form D_i that it multiplies with a row form D_i in a column vector with row vector way to generate |1) as the result. In practice each D_i, may be simplified for computational purposes to just one (wordlength) column of 1's and one (wordlength) row of 1's, giving rise to the enumerated mathematical form e.g.:

$$\begin{matrix} - \\ 1 \\ 1 \\ 1 \\ 1 \\ 1 \\ \smile \end{matrix} \quad * \quad |1\ 1\ 1\ 1\ 1\ ) \quad = \quad |\ 1\ ) \tag{9}$$

Although it does not matter which row and column is used, in this enumerated but fixed form there are an infinite set of such identical ways of generating |1). All of these are drawn from the generative formulation D but labeled D_1, D_2, D_2, … D_n. An alternative labeling might be the *i, j, k*, … (as used by Hamilton); a symbolism we were seeking to avoid, but which we can use to describe the behaviour.

6 An Arithmetic for *D* and *D*n

We have seen that D, in its generic form when squared yields –1 which is |1), and that this is exactly the same as the use of complex i to represent √–1.

A conjugate $–D$ also exists and would be a symbol in the universal alphabet generated by the conserve step. It's square is defined within existing rules, thus we get that:

$$–D * –D = (–1 * –1) * (D * D) = –1 \tag{10}$$

All other arithmetic associated with D in this generic form follow the rules associated with complex i.

We turn to consider the product of separate enumerated forms i.e. the result of D_i * D_j? In general these cannot be resolved until we know which row and column formulation and wordlengths are used to define D_i and D_j. When these are known, the product collapses to the anticommutative form and a closely related enumerated form, D_k. Thus:

$$D_i * D_j = –(D_j * D_i) = D_k. \tag{11}$$

where also:

$$D_j * D_k = –(D_k * D_j) = D_i \tag{12}$$

$$D_k * D_i = –(D_i * D_k) = D_j \tag{13}$$

These are Hamilton's rules as outlined above and illustrate the anticommutativity required at the outset. Their effect can also be demonstrated using a matrix representation of each form.

Finding conjugates of the enumerated forms in this way identifies one of an infinite number of three-fold groupings (triads) that are closely related. Each of these triads allows us to "create" distinguishable new symbols in the universal alphabet matched by conjugates which are then reserved and have this defined behaviour.

Other arithmetic properties of the enumerated forms follow Hamilton's quaternion rules with any triad's behaviour reserved in the same way. With this restriction we can also view the arithmetic as the arithmetic of "tiles" where in the limit each tile is a point in 4D space in exactly the way argued for tesseral quaternions in Bell & Mason (1990).

7 Discussion and Further Work

We have proposed a single symbol based method of extending the alphabet needed for the complexification stage of the universal rewrite mechanism described in Diaz and Rowlands (2004) and extended in principle in Rowlands (2005). This complexification stage alphabet now consists of the symbols (0, 1, –1, D, and –D) with a method for extending the symbol set indefinitely by considering and resolving the nature of the enumerated form triads as and when required by the iterative procedure. Thus D and –D each constitute alphabet generators as well as symbols in the existing alphabet.

Rowlands (2005) shows how this process can be used to generate the nilpotent Dirac equation using the anticommutativity property as the method that determines if a symbol (or indeed an entire sub-alphabet) is "new". It is, in fact, part of our fundamental argument that the rewrite system generates unique entities through the property of anticommutativity being available only within the original quaternion set at each stage of the alphabet's extension. This is how we generate numbers, including 1 and -1. To program such a process without an infinite number of symbols we need an enumerable method for identifying the infinite Square Root for –1. This has been achieved here by extending the common Booth method of representing negative numbers but done in two directions to yield a table of 1's, which we call D. As any indexed row/column of D when squared gives the repeated 1 representation of –1, that is |1), this provides a method for constructing an appropriate rewrite system. (Physically, we may suggest a connection between this representation and the fact that the nilpotent creation operator acting on vacuum leads to a filled vacuum for negative energy states as outlined in Rowlands, 2005.) An implementation of this approach will need to examine the braiding pattern implied by this symbol creation and use it to uniquely identify each created symbol.

8 References

[1] Bell, S.B.M., Diaz, B.M., Holroyd, F.C. and Jackson, M.J.J. (1983). Spatially Referenced Methods of Processing Raster and Vector Data. *Image and Vision Computing*, **1**(4), 211-220.

[2] Bell, S.B.M. & Mason, D.C. (1990) Tesseral quaternions for the octtree. *Computer Journal* **33**(5), 386-397.

[3] Booth, A.D. (1951) A signed binary multiplication technique, *Quart. Jl. Mech. Appl. Math*, **4**(2), 236-240.

[4] Diaz, B.M. and Bell, S.B.M. (1987). *Spatial Data Processing using Tesseral Methods*. NERC publication, Swindon, England, 420pp.

[5] Diaz, B.M. & Rowlands P. (2004) A computational path to the nilpotent Dirac algebra, *Int. J. Comput. Anticipatory Sys*. **16**, 140-150.

[6] Ell, T.A. & Sangwine, S.J. (2005) Quaternion involutions. ArXiv.org/pdf/math.RA/0506190.

[7] Gargantini, I. (1982) Linear octtrees for fast processing of three-dimensional objects. *Comput. Graphics and Image Processing*, **20**, 365-374.

[8] Griffiths, D.J. (2004). *Introduction to Quantum Mechanics* (2nd ed.). Prentice Hall, 230pp.

[9] Hamilton, W.R. (1899) *Elements of Quaternions*. Longman, 195pp.

[10] Koblitz, N. (1977) *p-Adic Numbers, p-adic Analysis, and Zeta Functions.* Springer-Verlag, 220p.

[11] Kuipers, J.B. (2002). *Quaternions and Rotation Sequences: A Primer With Applications to Orbits, Aerospace, and Virtual Reality*. Princeton University Press, 371pp.

[12] Navazo, I., Ayala, D. & Brunet, P. (1986) A geometric modeller based on the exact Octtree representation of polyhedra. *Computer Graphics Forum*, **5,** 91-104.

[13] Rowlands, P. & Diaz, B.M. (2002) A universal alphabet and rewrite system, arXiv.org/pdf/cs.OH/0209026

[14] Rowlands, P. (2003) The nilpotent Dirac equation and its application in particle physics, arXiv.org/pdf/quant-ph/0301071.

[15] Rowlands, P. (2005) Removing redundancy in relativistic quantum mechanics, arXiv.org/pdf/physics/0507188.

[16] Scott, N.R. (1985) *Computer Number Systems and Arithmetic*, Prentice-Hall, 255pp.