Evolutionary and Revolutionary Effects of Transcomputation

NOTICE

This paper will appear in the 2nd IMA Conference on Mathematics in Defence, Defence Academy of the United Kingdom, Shrivenham, England. October 2011.

Evolutionary and Revolutionary Effects of Transcomputation

James A.D.W. Anderson*

Abstract. We review transreal arithmetic and present transcomplex arithmetic. These arithmetics have no exceptions. This leads to incremental improvements in computer hardware and software. For example, the range of real numbers, encoded by floating-point bits, is doubled when all of the *Not-a-Number (NaN)* states, in IEEE 754 arithmetic, are replaced with real numbers. The task of programming such systems is simplified and made safer by discarding the *unordered* relational operator, leaving only the operators *less-than, equal-to,* and *greater than.* The advantages of using a transarithmetic in a computation, or *transcomputation* as we prefer to call it, may be had by making small changes to compilers and processor designs. However, radical change is possible by exploiting the reliability of transcomputations to make pipelined dataflow machines with a large number of cores. Our initial designs are for a machine with order one million cores. Such a machine can complete the execution of multiple in-line programs each clock tick.

1 Introduction

The greatest contribution science makes, to the defence of a nation, is to make it worth defending. In this spirit we present, in Section 2, a tutorial on transreal arithmetic that is accessible to school children. This gives children the mathematical tools to deal with the non-finite numbers: infinity, minus infinity, and nullity. Nullity is the number $0/0$ as discussed later in the present paper and in.[1] We prove, elsewhere, that transreal arithmetic, which allows division by zero, is consistent if real arithmetic is.[1] The tutorial gives a more accessible method of reckoning than the axiomatic methods used in the proof. We hope adults will also find it useful.

A large part of physics and applied mathematics deals with systems of complex numbers. In the main body of the paper we give a definition of transcomplex arithmetic which supports the division of complex numbers by zero. This allows physical problems to be solved exactly at a singularity, as noted earlier.[2]

In this earlier work,[2] we extended two's complement arithmetic to trans-two's-complement arithmetic by allowing division by zero. This transarithmetic has an equal number of positive and negative numbers. One advantage of this is that it removes bias from all signal processing and inertial navigation systems that both operate on two's complement numbers and, even sporadically, operate at full range. This removes the need to implement additional circuitry or programs to control the bias.

The arithmetic saturates at the signed infinities: positive infinity and negative infinity, leaving nullity outside the range from negative infinity to positive infinity. The encoding of nullity by the most negative bit pattern removes the weird-number fault from two's complement arithmetic in which the computed negation, or computed modulus, of the most negative number is identically this negative number. Thus the highest possible magnitude error is removed from two's complement arithmetic.

We hereby propose that integer and fixed-point arithmetics should have the same rounding modes as floating-point arithmetics so as not to introduce artificial limits to the expressibility of numerical programs, especially those involving division.

We now present, for the first time in the scientific literature, proposals to improve floating-point arithmetic by using transreal arithmetic to replace all of the *Not-a-Number (NaN)* states in IEEE 754 arithmetic[3, 4] and to remove the *unordered* relational operator, denoted by a question mark, ?, in that standard. Trans-floating-point numbers encode nullity where IEEE 754 has minus zero; they encode the signed infinities with maximum exponent and all mantissa bits set where IEEE 754 has all mantissa bits unset. The remaining bit patterns, with this exponent, represent real numbers not NaNs. This makes a small improvement to the accuracy or range of some numerical programs and greatly simplifies the programming, and improves the safety, of floating-point systems. We further propose that trans-floating-point numbers should reserve one bit as an *inexact* flag which may be tested, set and cleared by user programs, in addition to being operated on by trans-floating-point units (TFPUs). This reduces the accuracy of the arithmetic but allows all computations, whether finite or non-finite, to be sensitive to roundoff error without requiring exception handling circuitry in the TFPU.

We note that the absence of exceptions in transreal arithmetic means that any Turing program can be Gödelised so that it executes without exception. More practically, the absence of any logical exceptions means that a dataflow machine can be pipelined such that computational pipelines never break, except as a result of a physical fault. Such a machine, with order one million TFPUs, has been designed but has not yet been built. If built, it would complete multiple in-line programs each clock tick, offering a revolutionary improvement in computer performance.

*Systems Engineering, Reading University, Whiteknights, Reading, England, RG6 6AY.
author@bookofparagon.com

2 Tutorial on Transreal Arithmetic

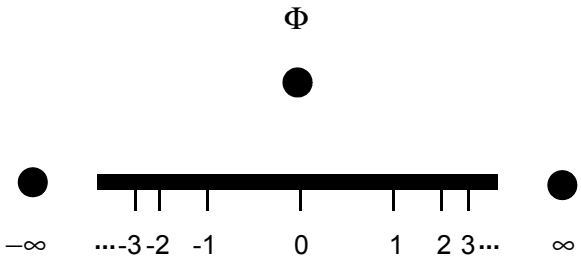

Figure 1: Transreal number-line.

The line, in Figure 1, is the *real number-line,* which contains all of the *finite numbers.* The big dots, ●, show the *non-finite numbers.* The *extended-real number-line* is the real number-line with infinity, ∞, and minus infinity, $-\infty$. Every number to the right of zero, on the extended-real number-line, is *positive;* every number to the left of zero, on the extended-real number-line, is *negative.* Zero is neither positive nor negative. The sign of zero is zero. This definition, which is widely accepted in the world, contradicts the teaching of sign in French speaking primary and secondary schools, where it is taught that zero is a positive number. The *transreal number-line* is the extended-real number-line with nullity. Nullity, Φ, lies off the extended-real number-line at distance and angle nullity. Nullity is neither positive nor negative. The sign of nullity is nullity. Infinity, ∞, is the most positive number and minus infinity, $-\infty$, is the most negative number. Infinity is bigger (further to the right on the extended-real number-line) than any number, except itself and nullity. Minus infinity is smaller (further to the left on the extended-real number-line) than any number except itself and nullity. Nullity is equal to itself, but is not bigger than or smaller than any number – because it is not on the extended-real number line. This paints a mental picture of how the transreal numbers relate to each other and gives a vocabulary for talking about these relationships. Both aspects are extended in more advanced study.

The *canonical* or *standard* or *least terms* form of certain numbers is as follows. Transreal one, 1, is real one divided by real one: $1 = 1/1$. Transreal minus-one, -1, is real minus-one divided by real one: $-1/1$. Transreal zero, 0, is real zero divided by real one: $0 = 0/1$. *Transreal infinity,* ∞, is real one divided by real zero: $\infty = 1/0$. *Transreal minus-infinity,* $-\infty$, is real minus-one divided by real zero: $-\infty = (-1)/0$. Transreal *nullity,* Φ, is real zero divided by real zero: $\Phi = 0/0$. Any irrational number x is x divided by real one: $x = x/1$.

Transreal numbers can be expressed as *transreal fractions,* n/d, of a real *numerator,* n, and a non-negative, real *denominator,* d. Transreal numbers with a non-finite numerator or denominator simplify to this form. An *improper fraction* can be written with a negative denominator, but it must be converted to a *proper fraction,* by carrying the sign up to the numerator, before applying any transreal, arithmetical operation. This can be done by multiplying both the numerator and denominator by minus one; it can be done by negating both the numerator and the denominator, using subtraction; and it can be done, lexically, by moving the minus sign from the denominator to the numerator.

$$\frac{n}{-d} = \frac{-1 \times n}{-1 \times (-d)} = \frac{-n}{-(-d)} = \frac{-n}{d} \tag{1}$$

Transreal infinity is equal to any positive number divided by zero. Transreal minus-infinity is equal to any negative number divided by zero. Zero is equal to zero divided by any positive or negative number. That is, with $k > 0$ we have:

$$\infty = \frac{1}{0} = \frac{k}{0} \qquad -\infty = \frac{-1}{0} = \frac{-k}{0} \qquad 0 = \frac{0}{1} = \frac{0}{k} = \frac{0}{-k} \tag{2}$$

The ordinary rules for multiplication and division apply universally to proper transreal-fractions. That is, they apply without side conditions. In particular, division by zero is allowed.

$$\frac{a}{b} \times \frac{c}{d} = \frac{a \times c}{b \times d} \tag{3}$$

$$\frac{a}{b} \div \frac{c}{d} = \frac{a}{b} \times \frac{d}{c} \tag{4}$$

Addition is more difficult than multiplication and division because it breaks into two cases: the addition of two signed infinities and the general case. Two infinities are added using the ordinary rule for adding fractions with a common denominator. The sign of each infinity, $\pm\infty$, may be chosen independently but the chosen sign is then carried into the corresponding term ± 1 so that $+\infty$ has $+1$ and $-\infty$ has -1:

$$(\pm\infty) + (\pm\infty) = \frac{\pm 1}{0} + \frac{\pm 1}{0} = \frac{(\pm 1) + (\pm 1)}{0} \tag{5}$$

Finite fractions may be added using this rule, if they happen to have a common denominator, but infinities cannot be added using the following general rule of addition. If infinities were added by the general rule we would have $\infty + \infty = \Phi$, but this is inconsistent with various arithmetics of the infinite that have $\infty + \infty = \infty$. See, for example.[5, 6, 7] The general case of the addition of proper transreal fractions is:

$$\frac{a}{b} + \frac{c}{d} = \frac{a \times d + b \times c}{b \times d} \tag{6}$$

Subtraction is the addition of a negated number:

$$\frac{a}{b} - \frac{c}{d} = \frac{a}{b} + \frac{-c}{d} \tag{7}$$

Transreal arithmetic is totally associative and totally commutative but it is only partially distributive at infinity. The axiom of

transreal distributivity[1] can be broken down into a number of cases. As usual, a number, a, distributes over $b + c$ only when:

$$a(b + c) = a \times b + a \times c \tag{8}$$

If a is finite or nullity then a distributes over any $b + c$. If a is infinity or minus infinity then a distributes if $b + c = \Phi$ or $b + c = 0$ or b and c have the same sign. Two numbers have the same sign if they are both positive, both negative, both zero or both nullity.

This is as all that has been presented to three hundred school children. We find that pupils in the range from 12 to 16 years, inclusive, have no psychological barriers to learning transreal arithmetic; older children, who have started the English and Welsh A-level syllabus, do have such intellectual inhibitions. Adults are particularly resistant to the notion of dividing by zero.

3 Transcomplex Arithmetic

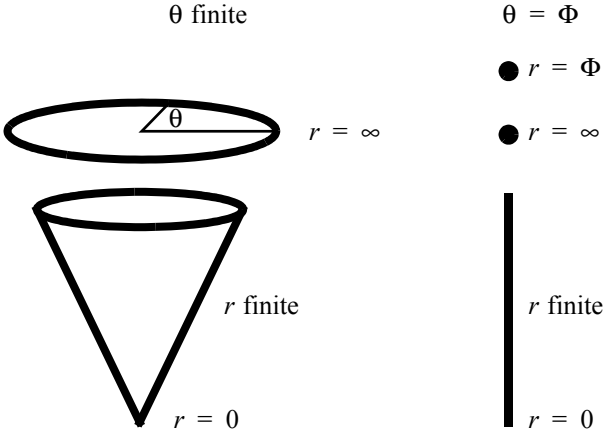

Figure 2: Transcomplex Top and Whip.

The transcomplex numbers are defined by three tuples (r, c, s) which specify the polar co-ordinates of a point with respect to complex zero, with r, the radius, measured as the Euclidean transmetric,[2] and c, s respectively the transreal cosine and sine. These transreal functions contain their real counterparts but have the property that the cosine and sine of any non-finite angle is nullity. This property is derived from their Taylor series in.[8] The Appendix defines transcomplex multiplication, division, addition and subtraction in these terms. An implementation of this arithmetic, together with examples of the computation of gravitational and electrostatic singularities, is available on-line.[9]

The three-tuple form has two significant advantages over specifying transcomplex numbers in polar form as a radius and angle. Firstly the three-tuple form is continuous in the complex plane without a cut and without winding onto a Riemann surface. Secondly the computation of the arithmetical operations, in terms of numerical sines and cosines, is more accurate and is much faster than computing these operations in terms of trigonometrical power series of angles. Further optimisations will be readily apparent to the reader of the appendix or source code.

However, it is easier to explain the properties of transcomplex numbers in polar form, (r, θ), which we do now. The cone, drawn in Figure 2, is the complex plane. The apex of the cone is labelled as complex zero, $(0, 0)$. All complex numbers with zero radius and finite angle map onto this polar zero. The circle at infinity, drawn above the cone, is the locus of all points with infinite radius and finite angle: (∞, θ) with $\theta \in R$. The cone and the circle at infinity make up the *extended cone.* If desired, functions may be wound around the extended cone forming a *trans-Riemann surface.* Cuts may also be taken in the extended cone. The non-negative part of the transreal number line, at angle nullity, is shown to the right of the figure. The point at infinite radius and nullity angle, (∞, Φ), corresponds to *complex infinity.* Both complex infinity and the circle at infinity are ordinarily used as compactifications of the complex plane. The *point at nullity,* (Φ, Φ), corresponds to the point which is ordinarily punctured from the complex plane, as noted earlier.[10] This point is needed to construct bijective maps between Euclidean and Projective spaces, as may be advantageous in computer vision. The real line at angle nullity is not used in ordinary mathematics but it is needed to make transcomplex arithmetic total. For mnemonic purposes, the extended cone is known as the (transcomplex) *top* and the non-negative part of the transreal number-line, at angle nullity, is known as the (transcomplex) *whip.* This mnemonic helps avoid errors where one or other parts of the figure are omitted from an analysis. Note that both of the ordinary compactifications are present in the figure. A total analysis interprets every point in the figure, even if only a subset of these points are considered important in a practical application. We recommend that total analyses are always undertaken in case round-off leads to the computation of otherwise unexpected results.

The operations of transcomplex arithmetic are carried out on transcomplex numbers in their standard form. This form may be read off from the appendix but is given here for convenience. The term on the left of the equivalence symbol, $\equiv$, is the standard form. All transreal components are in standard form.

$$(0, 1, 0) \equiv (0, \cos\theta, \sin\theta) \ : \ \theta \in R \tag{9}$$

$$(r, \Phi, \Phi) \equiv (r, \cos\theta, \sin\theta) \ : \ \theta \in \{-\infty, \infty, \Phi\} \tag{10}$$

$$(\Phi, \Phi, \Phi) \equiv (\Phi, \cos\theta, \sin\theta) \tag{11}$$

The source code[9] gives total conditions which map any syntactically correct sentence (r, c, s), with r, c, s being any transreal numbers, onto transcomplex numbers in standard form. Hence any syntactically correct formula of transcomplex and transreal arithmetic is semantically correct – which is just to say that no evaluation of the sentence produces an arithmetical exception. In practice this means that any totally transcomplex or transreal program, that compiles correctly, executes without exception. This makes applications programs safer and removes

the need for exception handling circuitry in a TFPU, as well as removing the need for error signalling between TFPUs, apart, of course, from the need to signal detected physical faults.

The operations of transcomplex arithmetic can be understood geometrically. As usual, multiplication and division are concatenations of a rotation and a dilatation. Additions are obtained by computing the resultant angle independently of the resultant radius. For example, by projecting the arguments onto finite vectors in a common plane; performing addition using the parallelogram rule; and projecting the intermediate sum, by a dilatation, onto the resultant plane. Thus the parallelogram rule is extended to infinite and nullity magnitudes. We maintain that both scalar and vector quantities can be handled in this way so that physical equations can be computed exactly at singularities, though it remains an open question as to whether such functions can be computed in the neighbourhood of a singularity.

The geometrical construction is easier if we blow up the apex of the top to a circle, scale the conical surface of the top to a cylinder with unit radius, and draw the whip as the axis, or *axle,* of the resulting Figure 3. Following Carlström,[11] we call the union of a circle and its centre point a *wheel.* The figure we operate on may also be obtained by sweeping a wheel along the extended-real part of the non-negative, transreal *axle*, oriented at angle nullity. Thus all points on the *rim* of the *swept wheel* lie at a finite angle and all points on the axle lie at angle nullity. It is to be understood that all points on the rim at zero are identified with the unique polar zero: $(0, 0)$. It is sometimes useful to call the centre point of a wheel a *hub.*

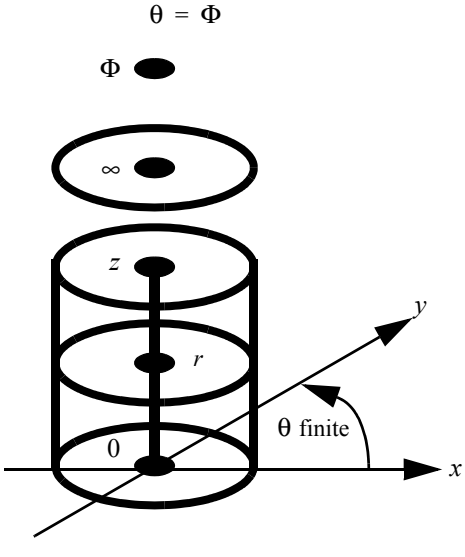

Figure 3: Transcomplex Wheel and Axle.

We illustrate the construction by verifying the transreal equation $1 + \infty = \infty$ by stepping through the addition algorithm, (A3), given in the appendix, describing the geometrical construction at each stage. The reader will find it instructive to verify the equation $\infty - \infty = \Phi$ by the same method.

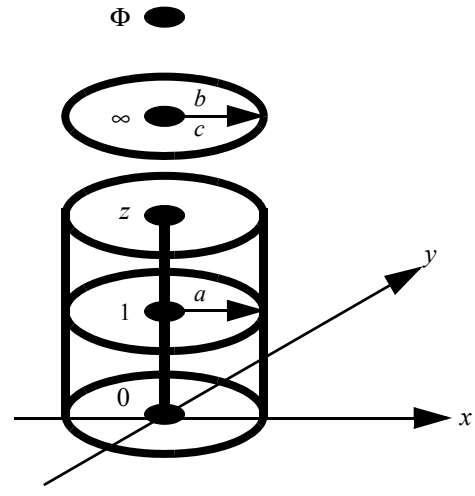

Figure 4: The sum $a + b = c$ with $a = 1$, $b = \infty$.

Consider the sum $a + b = c$ when $a = 1$ and $b = \infty$, then $c = \infty$. The arguments a, b are shown in Figure 4.

The arguments a, b are transreal and positive so they lie at angle zero. Therefore (A3) operates on the "otherwise" clause, with $c_1 = c_2 = 1$ and $s_1 = s_2 = 0$. The arguments' radii are given by $r_1 = 1$ and $r_2 = \infty$.

We compute the transreal sum $r_1 + r_2 = 1 + \infty = \infty$ and carry out the geometrical construction in the resultant common plane at $r_1 + r_2 = \infty$. Now $r_1 = r_1 + r_2$ so that $r_1' = 1$. Then:

$$r_2' = r_2 \div (r_1 + r_2) = 1 \div \infty = 0 \tag{12}$$

$$x = r_1' \times c_1 + r_2' \times c_2 = 1 \times 1 + 0 \times 1 = 1 \tag{13}$$

$$y = r_1' \times c_1 + r_2' \times c_2 = 1 \times 0 + 0 \times 0 = 0 \tag{14}$$

$$r_3' = \sqrt{x^2 + y^2} = \sqrt{1^2 + 0^2} = 1 \tag{15}$$

We have now computed r_3' in the common plane. We then dilatate it to its final length as:

$$r_3 = r_3' \times (r_1 + r_2) = 1 \times \infty = \infty \tag{16}$$

We then calculate the final angle independently of the final length as:

$$c_3 = x \div r_3' = 1 \div 1 = 1 \tag{17}$$

$$s_3 = y \div r_3' = 0 \div 1 = 0 \tag{18}$$

The final vector is shown at c which, in this particular case, lies in the common plane at infinity.

Subtraction is obtained as the addition of a negative number but there is no negative axis at angle nullity, by construction and by derivation:

$$\theta = \Phi + \pi = \Phi \tag{19}$$

$$\cos(\Phi + \pi) = \cos\Phi = \Phi \tag{20}$$

$$\sin(\Phi + \pi) = \sin\Phi = \Phi \tag{21}$$

Thus the transcomplex top, or equivalently the transcomplex wheel, manifests signed arithmetic and the transcomplex whip, or equivalently the transcomplex axle, manifests unsigned arithmetic. In physical problems, motions are generally computed in the wheel and energies in the axle.

4 Discussion

Some of the work of this paper is done in the introduction where we define trans-floating-point arithmetic. This arithmetic is irredundant – every bit pattern describes a unique transreal number. By contrast, IEEE floating-point arithmetic[3, 4] wastes a great many states. Taking m as the number of bits in the mantissa, there are $2(2^m - 1) = 2^{m+1} - 2$ signed NaN states, only half of which are distinguished by the standard, and only two of which must be implemented. The term, +1, in the exponent, arises from the sign bit; the term, -2, arises from the two signed infinities. The number of wasted states is tabulated below, using the nomenclature of the 2008 version of the standard.[4]

Name	m	Wasted NaN States
binary16	10	2 046
binary32	23	16 777 214
binary64	52	9 007 199 254 740 990
binary128	112	10 384 593 717 069 655 257 060 992 658 440 190

Table 1: Wasted NaN states.

If these wasted states are instead assigned to real numbers and the floating-point exponent's bias is kept constant then we almost double the range of real numbers described by the floating-point bits. But if we decrement the bias we improve accuracy by delaying underflow to denormal floating-point numbers near zero. Thus, an evolutionary improvement is made to floating-point arithmetic.

Trans-floating-point arithmetic discards the unordered relational operator, ?, given by the standard. This greatly simplifies the implementation of order tests in programs. This reduces the cognitive load on programmers, making it more likely that their programs will work correctly. It also reduces the number of conditions which must be tested to verify a program. In contrast to their IEEE 754 counterparts, the transreal relations *less-than, equal-to* and *greater-than* have no exceptions so they cannot cause a program to crash. Thus trans-floating-point programs are safer than IEEE 754 floating-point programs for both psychological and hardware reasons.

The tutorial on transreal arithmetic is offered both as an example of how arithmetic might be taught in schools and as an aid to the researcher. The transcomplex numbers offer many avenues for future research: we maintain that they have better continuity than any other system of complex numbers; they are total so they have no need for compactifications; they allow the evaluation of physical functions exactly at a singularity; they support both signed and unsigned arithmetic; they contain all previously described, complex number systems as special cases.

5 Conclusion

We have given a tutorial on transreal arithmetic which is accessible to school children. We have introduced an irredundant trans-floating-point arithmetic, essentially by abolishing NaNs and the *unordered* relational operator. We have extended complex arithmetic so that it allows division by zero. We have proposed to build massively pipelined machines by exploiting the absence of exceptions in transarithmetics.

References

1. J. A. D. W. Anderson, Norbert Völker, Andrew A. Adams "Perspex Machine VIII: Axioms of Transreal Arithmetic" in *Vision Geometry XV,* Longin Jan Lateki, David M. Mount, Angela Y. Wu, Editors, Proceedings of SPIE Vol. 6499 (2007).
2. J.A.D.W. Anderson "Perspex Machine XI: Topology of the Transreal Numbers" in *IMECS 2008,* S.I. Ao, Oscar Castillo, Craig Douglas, David Dagan Feng, Jeong-A Lee Editors, Hong Kong, pp. 330-338, March (2008).
3. *IEEE Standard 754 for Binary Floating-Point Arithmetic* (1985).
4. *IEEE Standard 754 for Floating-Point Arithmetic* (2008).
5. S. Swierczkowski, *Sets and Numbers*, Routledge & Kegan Paul, (1972).
6. L. Arkeryd, "The Evolution of Nonstandard Analysis" in *The Mathematical Association of America Monthly,* no.112, 926-928 (2005).
7. G. Leibman, "A Nonstandard Proof of the Fundamental Theorem of Algebra" in *The Mathematical Association of America Monthly,* no.112, 705-712 (2005).
8. J. A. D. W. Anderson "Perspex Machine IX: Transreal Analysis" in *Vision Geometry XV,* Longin Jan Lateki, David M. Mount, Angela Y. Wu, Editors, Proceedings of SPIE Vol. 6499 (2007).
9. An implementation of transcomplex arithmetic, in Pop11, along with examples of the calculation of gravitational and electrostatic singularities has been available since 5 Nov. 2010 at http://www.bookofparagon.com/Pages/Downloads.htm
10. J.A.D.W. Anderson, "Representing Geometrical Knowledge" in *Phil. Trans. Roy. Soc. Lond.* series B, **352**, no. 1358, 1129-1139 (1997).
11. J. Carlström, Wheels — on division by zero *Mathematical Structures in Computer Science,* **14**, no. 1, 143-184 (2004).

Appendix - Transcomplex Operations

$$
\begin{aligned}
(r_1, c_1, s_1) \times (r_2, c_2, s_2) &= (r_3, c_3, s_3) \text{ where} \\
r_3 &= r_1 \times r_2 \\
c_3 &= \begin{cases} 1 & : r_3 = 0 \\ c_1 \times c_2 - s_1 \times s_2 & : \text{otherwise} \end{cases} \\
s_3 &= \begin{cases} 0 & : r_3 = 0 \\ s_1 \times c_2 + c_1 \times s_2 & : \text{otherwise} \end{cases}
\end{aligned} \tag{A1}
$$

$$
\begin{aligned}
(r_1, c_1, s_1) \div (r_2, c_2, s_2) &= (r_3, c_3, s_3) \text{ where} \\
r_3 &= r_1 \div r_2 \\
c_3 &= \begin{cases} 1 & : r_3 = 0 \\ c_1 \times c_2 + s_1 \times s_2 & : \text{otherwise} \end{cases} \\
s_3 &= \begin{cases} 0 & : r_3 = 0 \\ s_1 \times c_2 - c_1 \times s_2 & : \text{otherwise} \end{cases}
\end{aligned} \tag{A2}
$$

$$
(r_1, c_1, s_1) + (r_2, c_2, s_2) = \begin{cases}
(r_1 + r_2, \Phi, \Phi) & : \text{any of } c_1, s_1, c_2, s_2 = \Phi \\
(r_3, c_3, s_3) & : \text{otherwise, where} \\
\qquad r_1' = \begin{cases} 1 & : r_1 = r_1 + r_2 \\ r_1 \div (r_1 + r_2) & : \text{otherwise} \end{cases} \\
\qquad r_2' = \begin{cases} 1 & : r_2 = r_1 + r_2 \\ r_2 \div (r_1 + r_2) & : \text{otherwise} \end{cases} \\
\qquad x = r_1' \times c_1 + r_2' \times c_2 \\
\qquad y = r_1' \times s_1 + r_2' \times s_2 \\
\qquad r_3' = \sqrt{x^2 + y^2} \\
\qquad r_3 = r_3' \times (r_1 + r_2) \\
\qquad c_3 = \begin{cases} 1 & : r_3' = 0 \\ x \div r_3' & : \text{otherwise} \end{cases} \\
\qquad s_3 = \begin{cases} 0 & : r_3' = 0 \\ y \div r_3' & : \text{otherwise} \end{cases}
\end{cases} \tag{A3}
$$

$$
(r_1, c_1, s_1) - (r_2, c_2, s_2) = (r_1, c_1, s_1) + (1, -1, 0) \times (r_2, c_2, s_2) \tag{A4}
$$