Exact Real Arithmetic based on Linear Fractional Transformations

Peter John Potts **Abbas Edalat**
Department of Computing
Imperial College
180 Queen's Gate, London SW7 2BZ, United Kingdom
{pjp,ae}@doc.ic.ac.uk

1 December 1996

Abstract

We introduce a feasible and incremental framework for exact real arithmetic based on the composition of linear fractional transformations with *either all non-negative or all non-positive integer coefficients*. We include a set of algorithms for the basic arithmetic operations and the elementary functions.

1 Introduction

It is generally accepted that floating-point computation is suitable for a wide range of applications. However, it is well-known that the accumulation of round-off errors due to a large number of floating-point calculations can produce grossly inaccurate or even incorrect results.

Interval analysis [18] has been used to partially circumvent this problem by maintaining a pair of bounding floating-point numbers that is guaranteed to contain the real number or interval in question. However, this interval can get unjustifiably large and thereby convey very little information.

In principle, exact real arithmetic provides an alternative technique for real number computation and verification of numerical algorithms. In practice, however, exact real arithmetic has proved too inefficient to provide a feasible alternative.

In the literature, there are broadly speaking three frameworks for exact real computer arithmetic:

(i) *Infinite sequences of linear maps* proposed by Avizienis [1] and appeared in the work of Watanuki et al [25], Boehm an Cartwright [2], Di Gianantonio [7], Escardo [4], Nielsen et al [19] and Menissier-Morain [17].

(ii) *Continued fraction expansions* proposed by Gosper [8], developed by Peyton Jones [11] and Vuillemin [23] and advanced more recently by Kornerup et al [16, 14, 13, 15].

(iii) *Infinite composition of linear fractional transformations* (also known as homographies or Möbius transformations) generalises the other two frameworks as demonstrated by Vuillemin [23]. Nielsen et al [19] showed that this framework can be used to represent quasi-normalised floating point [25].

We introduce here a new, feasible and incremental representation of the non-negative extended real numbers based on the composition of linear fractional transformations with *either all non-negative or all non-positive integer coefficients* [21].

Prototypes have been implemented in C++, Java and Miranda.

2 Linear Fractional Transformations

A natural way to represent a real number, r say, is by a sequence of nested rational intervals enclosing r such that the sequence of interval lengths converges to zero [9, 18].

Let $\mathbb{R}$ denote the set of real numbers with the Euclidean topology, $\mathbb{R}^\infty$ the one point compactification of $\mathbb{R}$ and $[0,\infty]$ the one point compactification of the non-negative real numbers $[0,\infty)$. The closed intervals $[a,b]$ in $\mathbb{R}^\infty$ are defined as the points from a to b in the numerically increasing direction, possibly including ∞. For example, the closed interval $[1,-1]$ is the complement of the open interval $(-1,1)$.

Definition 2.1 *A* 0-dimensional linear fractional transformation (lft) *with real coefficients is a fraction in* $\mathbb{R}^\infty$*, namely a homogeneous coordinate representation of an extended real number,*

$$t_{\begin{pmatrix} a \\ b \end{pmatrix}} = \frac{a}{b} \tag{1}$$

where $a \neq 0$ *or* $b \neq 0$*. A* 1-dimensional lft *with real coefficients is a function from* $\mathbb{R}^\infty$ *to* $\mathbb{R}^\infty$ *with the general form*

$$t_{\begin{pmatrix} a & c \\ b & d \end{pmatrix}}(x) = \frac{ax+c}{bx+d} \tag{2}$$

where the determinant $\begin{vmatrix} a & c \\ b & d \end{vmatrix} \neq 0$*. A* 2-dimensional lft *with real coefficients is a function from* $\mathbb{R}^\infty \times \mathbb{R}^\infty$ *to* $\mathbb{R}^\infty$ *with the general form*

$$t_{\begin{pmatrix} a & c & e & g \\ b & d & f & h \end{pmatrix}}(x,y) = \frac{axy+cx+ey+g}{bxy+dx+fy+h} \tag{3}$$

where $\begin{vmatrix} ax+e & cx+g \\ bx+f & dx+h \end{vmatrix}$ *and* $\begin{vmatrix} ay+c & ey+g \\ by+d & fy+h \end{vmatrix}$*, as functions of* x *and* y *respectively, are not identically* 0*.*

In homogeneous coordinates, a 1-dimensional lft is reduced to matrix multiplication

$$\begin{pmatrix} a & c \\ b & d \end{pmatrix} : \begin{pmatrix} p \\ q \end{pmatrix} \longmapsto \begin{pmatrix} ap+cq \\ bp+dq \end{pmatrix}.$$

Therefore, it is convenient to drop the t in Equations (1), (2) and (3). In general, we will use the letters V and W to denote vectors, M and N to denote matrices and T and U to denote rank 3 tensors, which correspond to 2-dimensional lft's. These are collectively referred to as tensors. We will also use x and y to denote non-negative extended real numbers.

Definition 2.2 *The* information $\mathsf{Info}(P)$ *contained by an lft P is defined by* $\mathsf{Info}(V) = \{V\}$*,* $\mathsf{Info}(M) = M([0,\infty])$ *and* $\mathsf{Info}(T) = T([0,\infty],[0,\infty])$*.*

Observe that $\mathsf{Info}(M) \subseteq [0,\infty]$ whenever the coefficients of M are *either all non-negative or all non-positive integers* In this paper, we present a framework for exact real arithmetic using lft's with either all non-negative or all non-positive coefficients. **From now onwards, unless otherwise stated, assume all lft's have either all non-negative or all non-positive coefficients.**

It is convenient to consider a rank 3 tensor T as a pair of matrices (T_0, T_1), a matrix M as a pair of vectors (M_0, M_1) and a vector V as a pair of integers (V_0, V_1).

Define the max of two vectors by

$$\max(V,W) = \begin{cases} V & \text{if } V \geq W \\ W & \text{if } V \leq W \\ V & \text{if } W = \begin{pmatrix} 0 \\ 0 \end{pmatrix} \\ W & \text{if } V = \begin{pmatrix} 0 \\ 0 \end{pmatrix} \end{cases}$$

and min similarly, where $\leq$ is the usual ordering on $[0,\infty]$ using homogeneous coordinates. Also, let $\overline{M} = \max(M_0, M_1)$, $\underline{M} = \min(M_0, M_1)$, $\overline{T} = \max(\overline{T_0}, \overline{T_1})$ and $\underline{T} = \min(\underline{T_0}, \underline{T_1})$.

Lemma 2.3 *The information contained by a matrix M and a rank 3 tensor T is given by:*

(i) $\mathsf{Info}(M) = \left[\underline{M}, \overline{M}\right] \subseteq [0,\infty]$

(ii) $\mathsf{Info}(T) = \left[\underline{T}, \overline{T}\right] \subseteq [0,\infty]$

Proposition 2.4 *Composition of 1-dimensional lft's M and N corresponds to refinement of non-negative rational intervals:*

$$M(N([0,\infty])) \subseteq M([0,\infty])$$

In other words, $[0, \infty]$ represents no information and N refines the information given by M. In fact, we have:

Proposition 2.5 *Given two rational intervals I and J, we have $I \subseteq J$ iff there exists an lft M such that $I = M(J)$.*

Any non-negative real number can therefore be represented as the intersection

$$\bigcap_{n \geq 0} M_0 M_1 M_2 \ldots M_n([0, \infty])$$

for a sequence of lft's M_n. We can denote this real number by an infinite product of matrices

$$\begin{pmatrix} a_0 & c_0 \\ b_0 & d_0 \end{pmatrix} \begin{pmatrix} a_1 & c_1 \\ b_1 & d_1 \end{pmatrix} \begin{pmatrix} a_2 & c_2 \\ b_2 & d_2 \end{pmatrix} \cdots \tag{4}$$

which we will call an *infinite normal product.* This notion generalises the concept of *interval expansion* [4] in which M_n is restricted to a linear map. Notice however that an infinite normal product does not in general represent a point, although it always represents an interval. A singular matrix is in fact a constant that can be replaced by a vector, thus terminating the product; this we will call a *finite normal product.* Thus, a finite normal product represents a rational number, whereas an infinite normal product may represent any number.

As mentioned above, the tensor representing an lft is only unique up to scaling. Hence, we can identify the lft's with the equivalence classes arising from the equivalence relation $\equiv$ on tensors induced by scaling. Let us denote by P^*, the tensor P reduced to its lowest terms after division by the greatest common divisor of the coefficients. We can then identify a unique tensor P^* in each equivalence class.

This gives a simple representation and a convenient operational semantics for the lazy representation of the reals: finite segments of the matrix product in Equation (4) give incremental approximations to the real number in question. In particular, the first matrix tells us that the result is contained in the interval $[a_0/b_0, c_0/d_0]$ or $[c_0/d_0, a_0/b_0]$ depending on the sign of the determinant of the matrix.

3 Arithmetic Operations

The spirit of Gosper's [8] Quadratic Algorithm for continued fractions [24] using 2-dimensional lft's, further developed by Vuillemin [23], can be extended to normal products.

The three most basic arithmetic operations closed on $[0, \infty]$ can be represented as follows:

$$\begin{pmatrix} 0 & 1 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{pmatrix} (x, y) \quad = \quad x + y$$

$$\begin{pmatrix} 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 \end{pmatrix}(x,y) = x \times y$$
$$\begin{pmatrix} 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \end{pmatrix}(x,y) = x \div y$$

Therefore, we need to be able to convert expressions containing vectors, matrices and rank 3 tensors into normal products. In other words, we need an operational semantics for the *absorption* and *emission* of normal products to and from matrices and rank 3 tensors.

In order to simplify composition of lft's of various dimensions, we define the *dot product*, the *left product* and the *right product*, denoted respectively by $\cdot$, Ⓛ and Ⓡ as follows:

$$\begin{aligned}
(M \cdot V)_i &= \sum_{j=0,1} M_{ij} V_j \\
(M \cdot N) &= (M \cdot N_0, M \cdot N_1) \\
(M \cdot T) &= (M \cdot T_0, M \cdot T_1) \\
T \circledR V &= (T_0 \cdot V, T_1 \cdot V) \\
T \circledR M &= (T_0 \cdot M, T_1 \cdot M) \\
T \,\text{Ⓛ}\, V &= T^{\top} \circledR V \\
T \,\text{Ⓛ}\, M &= (T^{\top} \circledR M)^{\top}
\end{aligned}$$

where $T^{\top}$ indicates the transpose of T defined by swapping its middle two columns.

Proposition 3.1 *The following* matrix absorption *equations hold:*

$$\begin{aligned}
M(V) &= M \cdot V \\
M(N(x)) &= (M \cdot N)(x)
\end{aligned}$$

The following rank 3 tensor absorption *equations hold:*

$$\begin{aligned}
T(V,y) &= \begin{cases} L(y) & \text{if } |L| \neq 0 \\ \underline{L} \equiv \overline{L} & \text{if } |L| = 0 \end{cases} \\
T(M(x),y) &= (T \,\text{Ⓛ}\, M)(x,y) \\
T(x,V) &= \begin{cases} R(y) & \text{if } |R| \neq 0 \\ \underline{R} \equiv \overline{R} & \text{if } |R| = 0 \end{cases} \\
T(x,M(y)) &= (T \circledR M)(x,y)
\end{aligned}$$

where $L = T \,\text{Ⓛ}\, V$ *and* $R = T \circledR V$.

Note that the left and right products of a rank 3 tensor with a vector may give a singular matrix, which is essentially a vector.

For computing the value of $T(x, y)$, we need a **strategy** for deciding whether to absorb from x or y. The aim is to reduce the length of the interval $\mathsf{Info}(T)$. Note that an x absorption reduces the lengths of the intervals $X_0 = \mathsf{Info}((T^\top)_0)$ and $X_1 = \mathsf{Info}((T^\top)_1)$, while a y absorption reduces the lengths of the intervals $Y_0 = \mathsf{Info}(T_0)$ $Y_1 = \mathsf{Info}(T_1)$. Therefore, the most reasonable approach is **not** to select x if $X_0 \cap X_1 = \emptyset$ and **not** to select y if $Y_0 \cap Y_1 = \emptyset$. If both intersections are non-empty then choose x, because this is the argument of the elementary functions defined in subsection 5.3.

The information in a 2-dimensional lft T can be represented by the 1-dimensional lft $(\overline{T}, \underline{T})$, denoted by T^{head}. Thus, any matrix M can be emitted from a rank 3 tensor T so long as $\mathsf{Info}(M) \supseteq \mathsf{Info}(T) = \mathsf{Info}(T^{\text{head}})$. The choice $M = T^{\text{head}}$ is called *naive emission*. Let us define $T^{\text{tail}} = (T^{\text{head}})^{-1} \cdot T$, where matrix inversion is defined by

$$\begin{pmatrix} a & c \\ b & d \end{pmatrix}^{-1} = \begin{pmatrix} d & -c \\ -b & a \end{pmatrix}.$$

We choose to scale the matrix inversion by the determinant in order to ensure we only get integers in the rank 3 tensor emission equation.

Proposition 3.2 *The following* rank 3 tensor emission *equation holds:*

$$T(x, y) = T^{\text{head}}(T^{\text{tail}}(x, y)).$$

This corresponds to the extraction of maximum information, but in practice it gives rise to an integer size explosion.

A number of alternative emission methods were tried in practice. However, we found that one of the simplest and most efficient scheme is to first emit a matrix of the form

$$E_e = \begin{pmatrix} 2^e & 0 \\ 1 & 1 \end{pmatrix}$$

representing the exponent $e \geq 0$ of the number, followed by a sequence of the three matrices

$$\begin{aligned} M_{-1} &= \begin{pmatrix} 0 & 1 \\ 2 & 1 \end{pmatrix} \\ M_0 &= \begin{pmatrix} 1 & 3 \\ 3 & 1 \end{pmatrix} \\ M_1 &= \begin{pmatrix} 1 & 2 \\ 1 & 0 \end{pmatrix} \end{aligned}$$

representing the mantissa of the number.

We will show that in fact this scheme leads to the emission of a sequence of nested 2-adic intervals of the form

$$\left[\frac{m-1}{2^n}, \frac{m+1}{2^n}\right] \tag{5}$$

which is essentially an arbitrary precision floating point number.

In order to see this, put

$$\langle n, m, e\rangle = \left(\begin{array}{cc} (2^n + m + 1)2^e & (2^n + m - 1)2^e \\ 2^{n+1} & 2^{n+1} \end{array} \right)$$

where $-2^n < m < 2^n$ and note that $E_e M_m = \langle 1, m, e\rangle$.

Recall [1] that any real number can be represented by sequences of signed binary digits, namely -1, 0 and 1, in the form

$$2^{e-1}b_{e-1} + \cdots + 4b_2 + 2b_1 + b_0 + \frac{1}{2}b_{-1} + \frac{1}{4}b_{-2} + \cdots$$

denoted by

$$b_{e-1} \ldots b_2 b_1 b_0 . b_{-1} b_{-2} \ldots$$

where $b_i \in \{-1, 0, 1\}$ for all integers $i < e$.

Proposition 3.3

$$E_e M_{c_0} M_{c_1} M_{c_2} \cdots M_{c_{n-1}} = \langle n, c_0 c_1 c_2 \ldots c_{n-1}, e\rangle$$

Observe that

$$\mathsf{Info}(\langle n, m, e\rangle) = \left(1 + \left[\frac{m-1}{2^n}, \frac{m+1}{2^n}\right]\right) 2^{e-1}.$$

Therefore, as promised, a sequence of nested 2-adic intervals of the form given in Equation (5) is emitted.

4 Continued Fractions

The development

$$a_0 + \cfrac{b_0}{a_1 + \cfrac{b_1}{a_2 + \cfrac{b_2}{a_3 + \cdots}}} \tag{6}$$

is called a *continued fraction* [3, 12, 5, 6].

The quantity

$$\frac{P_n}{Q_n} = a_0 + \cfrac{b_0}{a_1 + \cfrac{b_1}{a_2 + \cdots + \cfrac{b_{n-1}}{a_n}}} \tag{7}$$

is called the n^{th} *approximant*. The 0^{th} approximant is a_0. If the sequence P_n/Q_n converges to a real number r then the continued fraction is said to be convergent and represent the number r.

Using the lft's

$$M_n(x) = \left(\begin{array}{cc} a_n & b_n \\ 1 & 0 \end{array} \right)(x) = a_n + \frac{b_n}{x} \tag{8}$$

we can generate the continued fraction (6). Therefore, a continued fraction with either all non-negative or all non-positive coefficients corresponds to a normal product.

We can further observe that

$$\prod_{n=0}^{m} M_n \equiv \begin{pmatrix} P_m & P_{m-1} \\ Q_m & Q_{m-1} \end{pmatrix} \tag{9}$$

where $P_{-1} = b_0$ and $Q_{-1} = 0$.

Of course, we should consider the issue of convergence. The following comprehensive test was devised by Pringsheim [22].

Lemma 4.1 *The divergence of the series*

$$\sum_{n=2}^{\infty} \sqrt{\frac{b_{n-1}b_n}{a_n}} \tag{10}$$

is necessary and sufficient for the convergence of the continued fraction (6) with non-negative coefficients.

Note that the product of any two matrices is unchanged by swapping the columns of the first matrix and the rows of the second matrix at the same time. By carrying out this procedure when necessary along an infinite normal product, we can assume without loss of generality, that all the matrices have strictly positive determinants. If we combine this with the observation that

$$\begin{pmatrix} a & c \\ b & d \end{pmatrix} = \begin{pmatrix} \frac{c}{d} & \frac{k}{d} \\ 1 & 0 \end{pmatrix} \cdot \begin{pmatrix} b & d \\ 1 & 0 \end{pmatrix}$$

with $k = \begin{vmatrix} a & c \\ b & d \end{vmatrix}$, we obtain a necessary and sufficient condition for the convergence of an infinite normal product using Lemma 4.1.

5 Algorithms

This section contains algorithms by Potts [21] for π and the elementary functions; power, sine, cosine, tangent, inverse tangent, exponential, natural logarithm, hyperbolic tangent, inverse hyperbolic sine and inverse hyperbolic tangent. The other elementary functions namely inverse sine, inverse cosine, hyperbolic sine, hyperbolic cosine and inverse hyperbolic cosine can be defined in terms of the above.

5.1 Pi

Using Ramanujan's formula [10] for π

$$\frac{1}{\pi} = \sum_{n=0}^{\infty} (-1)^n \frac{12(6n)!}{(n!)^3(3n)!} \frac{13591409 + 545140134n}{(640320^3)^{n+\frac{1}{2}}}$$

we can derive the normal product

$$\frac{\sqrt{10005}}{\pi} = \prod_{n=0}^{\infty} \mathrm{I}^{\text{PI}}_n$$

where the iterator I^{PI}_n is given by

$$\begin{aligned} \mathrm{I}^{\text{PI}}_{n=0} &= \begin{pmatrix} 13591409 & 0 \\ 426880 & 426880 \end{pmatrix} \\ \mathrm{I}^{\text{PI}}_{n>0} &= \begin{pmatrix} b_n - a_n & b_n \\ a_n & 0 \end{pmatrix} \end{aligned}$$

with

$$\begin{aligned} a_n &= (1-6n)(1-2n)(6n-5)c_n \\ b_n &= n^3 d_n \\ c_n &= 13591409 + 545140134n \\ d_n &= 5963320012791731724288000n - 5814642789749963059200000 \end{aligned}$$

5.2 Square Root of a Positive Integer

The square root of a positive integer m is most easily calculated using the well known formula

$$\sqrt{m} = \prod_{n=0}^{\infty} \mathrm{I}^{\text{SQRT}}(m)$$

where the iterator $\mathrm{I}^{\text{SQRT}}(m)$ is given by

$$\mathrm{I}^{\text{SQRT}}(m) = \begin{pmatrix} \lceil\sqrt{m}\rfloor & m \\ 1 & \lceil\sqrt{m}\rfloor \end{pmatrix}$$

and $\lceil x \rfloor$ is the nearest integer to x.

5.3 Elementary Functions

Note that a 1-dimensional lft with an argument y and linear coefficients in a parameter x is in fact a 2-dimensional lft with the arguments x and y:

$$\begin{pmatrix} ax+e & cx+g \\ bx+f & dx+h \end{pmatrix}(y) = \begin{pmatrix} a & c & e & g \\ b & d & f & h \end{pmatrix}(x,y)$$

The coefficients a,b,c,d,e,f and g can be allowed to be negative integers provided that we restrict the domain of x to the interval I such that the entries of the matrix evaluate to elements of $[0,\infty]$. This is because if x reduces to the form $M(z)$ where $\mathsf{Info}(M) \subseteq I$ then the left product of the tensor with M can be shown to give a tensor with either all non-negative or all non-positive integer coefficients, thus allowing the emission of matrices.

Many continued fractions for elementary functions have been derived by using various techniques [20, 24] from their power series, most notably by Euler [5, 6] and Gauss. The continued fractions are suitably chosen and then converted into 2-dimensional lft's using the following matrix identities:

$$
\begin{aligned}
\begin{pmatrix} a & c \\ b & d \end{pmatrix} &\equiv \begin{pmatrix} ax & cx \\ bx & dx \end{pmatrix} \\
\begin{pmatrix} a & c \\ b & 0 \end{pmatrix}\begin{pmatrix} d & f \\ e & 0 \end{pmatrix} &\equiv \begin{pmatrix} a & cx \\ b & 0 \end{pmatrix}\begin{pmatrix} dx & fx \\ e & 0 \end{pmatrix} \\
\begin{pmatrix} a & cx \\ b & 0 \end{pmatrix}\begin{pmatrix} d & f \\ e & 0 \end{pmatrix} &= \begin{pmatrix} a & c \\ b & 0 \end{pmatrix}\begin{pmatrix} d & f \\ ex & 0 \end{pmatrix} \\
\begin{pmatrix} a & c \\ b & d \end{pmatrix}\begin{pmatrix} e & g \\ f & h \end{pmatrix} &\equiv \begin{pmatrix} c & a \\ d & b \end{pmatrix}\begin{pmatrix} f & h \\ e & g \end{pmatrix} \\
\begin{pmatrix} ax & c \\ bx & d \end{pmatrix}\begin{pmatrix} e & g \\ f & h \end{pmatrix} &= \begin{pmatrix} a & c \\ b & d \end{pmatrix}\begin{pmatrix} ex & gx \\ f & h \end{pmatrix} \\
\begin{pmatrix} 0 & 1 \\ 1 & 0 \end{pmatrix}\prod_{n=1}^{\infty}\begin{pmatrix} a_n & c_n \\ b_n & d_n \end{pmatrix} &= \prod_{n=1}^{\infty}\begin{pmatrix} d_n & b_n \\ c_n & a_n \end{pmatrix} \\
\begin{pmatrix} a & c \\ b & d \end{pmatrix} &\equiv \begin{pmatrix} c & k \\ d & 0 \end{pmatrix}\begin{pmatrix} b & d \\ 1 & 0 \end{pmatrix} \\
\text{where } k &= \begin{vmatrix} a & c \\ b & d \end{vmatrix}
\end{aligned}
$$

Listed below is a set of algorithms for the elementary functions in the form of infinite right products of 2-dimensional lft's, which are valid for values of x and y such that the entries of the matrices in the corresponding infinite products evaluate to non-negative extended real numbers:

$$
\begin{aligned}
x^y &= \prod_{n=0}^{\infty} \mathrm{I}_n^{\text{POWY}}(y)\mathrm{I}_n^{\text{POWX}}(x) \\
\frac{\sin\sqrt{x}}{\sqrt{x}} &= \prod_{n=0}^{\infty} \mathrm{I}_n^{\text{SIN}}(x) \\
\cos\sqrt{x} &= \prod_{n=0}^{\infty} \mathrm{I}_n^{\text{COS}}(x) \\
\frac{\tan\sqrt{x}}{\sqrt{x}} &= \prod_{n=0}^{\infty} \mathrm{I}_n^{\text{TAN}}(x) \\
\arctan x &= \prod_{n=0}^{\infty} \mathrm{I}_n^{\text{ARCTAN}}(x)
\end{aligned}
$$

$$
\begin{aligned}
\exp x &= \prod_{n=0}^{\infty} \mathrm{I}_n^{\text{EXP}}(x) \\
\ln x &= \prod_{n=0}^{\infty} \mathrm{I}_n^{\text{LN}}(x) \\
\tanh x &= \prod_{n=0}^{\infty} \mathrm{I}_n^{\text{TANH}}(x) \\
\operatorname{arcsinh}\, x &= \prod_{n=0}^{\infty} \mathrm{I}_n^{\text{ARCSINH}}(x) \\
\operatorname{arctanh}\, x &= \prod_{n=0}^{\infty} \mathrm{I}_n^{\text{ARCTANH}}(x)
\end{aligned}
$$

where

$$
\begin{aligned}
\mathrm{I}_{n=0}^{\text{POWY}}(y) &= \begin{pmatrix} y & 1 \\ 0 & 1 \end{pmatrix}, \\
\mathrm{I}_{n>0}^{\text{POWY}}(y) &= \begin{pmatrix} 0 & n-y \\ n+y & 2 \end{pmatrix}, \\
\mathrm{I}_{n}^{\text{POWX}}(x) &= \begin{pmatrix} 0 & x-1 \\ x-1 & 1+2n \end{pmatrix}, \\
\mathrm{I}_{n=0}^{\text{SIN}}(x) &= \begin{pmatrix} 1 & 0 \\ 1 & x \end{pmatrix}, \\
\mathrm{I}_{n>0}^{\text{SIN}}(x) &= \begin{pmatrix} 2n(2n+1)-x & 2n(2n+1)x \\ 1 & 0 \end{pmatrix}, \\
\mathrm{I}_{n=0}^{\text{COS}}(x) &= \begin{pmatrix} 1 & 0 \\ 1 & x \end{pmatrix}, \\
\mathrm{I}_{n>0}^{\text{COS}}(x) &= \begin{pmatrix} 2n(2n-1)-x & 2n(2n-1)x \\ 1 & 0 \end{pmatrix}, \\
\mathrm{I}_{n=0}^{\text{TAN}}(x) &= \begin{pmatrix} 15-2x & x \\ 15-7x & x \end{pmatrix}, \\
\mathrm{I}_{n>0}^{\text{TAN}}(x) &= \begin{pmatrix} p_n-(7+12n)x & (1+4n)x \\ p_n-4(3+4n)x & (1+4n)x \end{pmatrix} \\
\text{with } p_n &= (1+4n)(3+4n)(5+4n), \\
\mathrm{I}_{n}^{\text{ARCTAN}}(x) &= \begin{pmatrix} 0 & x \\ (1+n)^2 x & 1+2n \end{pmatrix},
\end{aligned}
$$

$$
\begin{aligned}
\mathrm{I}^{\text{EXP}}_{n=0}(x) &= \begin{pmatrix} x & 2+x \\ x & 2-x \end{pmatrix}, \\
\mathrm{I}^{\text{EXP}}_{n>0}(x) &= \begin{pmatrix} 0 & x \\ x & 2(1+2n) \end{pmatrix}, \\
\mathrm{I}^{\text{LN}}_{n}(x) &= \begin{pmatrix} (x-1)q_n & 2(x-1) \\ (1+2n)q_n & (1+3n)+q_n x \end{pmatrix} \\
\text{with } q_n &= 1+n, \\
\mathrm{I}^{\text{TANH}}_{n}(x) &= \begin{pmatrix} 0 & x \\ x & 1+2n \end{pmatrix}, \\
\mathrm{I}^{\text{ARCSINH}}_{n=0}(x) &= \begin{pmatrix} 1 & 0 \\ 1 & x \end{pmatrix}, \\
\mathrm{I}^{\text{ARCSINH}}_{n>0}(x) &= \begin{pmatrix} 2nr_n-(2n-1)^2x & 2nr_n x \\ (2n-1)^2 & 0 \end{pmatrix}, \\
\mathrm{I}^{\text{ARCTANH}}_{n}(x) &= \begin{pmatrix} (1+n)x & x \\ (1+n)r_n(1-x) & r_n-nx \end{pmatrix} \\
\text{with } r_n &= 1+2n.
\end{aligned}
$$

5.4 Example

The Stieltjes type continued fraction for $\arctan x$ is given by

$$\arctan x = \cfrac{x}{1+\cfrac{\frac{x^2}{3}}{1+\cfrac{\frac{4x^2}{15}}{1+\cdots}}}$$

This can be transformed to

$$\arctan x = \prod_{n=0}^{\infty} \begin{pmatrix} 0 & x \\ (1+n)^2x & 1+2n \end{pmatrix}$$

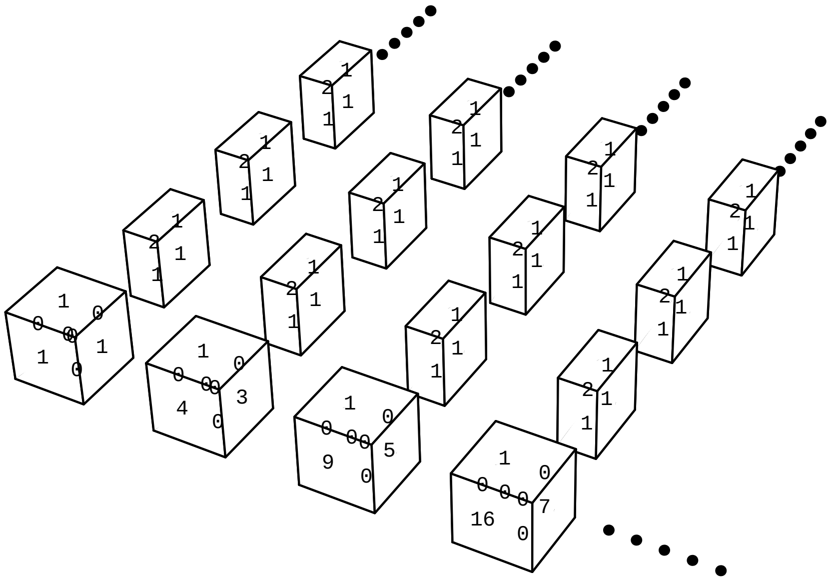
or put another way

$$
\begin{aligned}
&\arctan x = \\
&\quad \begin{pmatrix} 0 & 1 & 0 & 0 \\ 1 & 0 & 0 & 1 \end{pmatrix}(x, \begin{pmatrix} 0 & 1 & 0 & 0 \\ 4 & 0 & 0 & 3 \end{pmatrix}(x, \\
&\quad \begin{pmatrix} 0 & 1 & 0 & 0 \\ 9 & 0 & 0 & 5 \end{pmatrix}(x, \begin{pmatrix} 0 & 1 & 0 & 0 \\ 16 & 0 & 0 & 7 \end{pmatrix}(x, \cdots)))).
\end{aligned}
$$

For example, the tree of lft's representing $\arctan\phi$, where ϕ is the Golden Ratio given by

$$\phi = \prod_{n=0}^{\infty} \begin{pmatrix} 2 & 1 \\ 1 & 1 \end{pmatrix},$$

can be pictured as



6 Conclusion

We have presented here a framework for exact real arithmetic on the non-negative extended real numbers using linear fractional transformations with either all non-negative or all non-positive integer coefficients, including a set of algorithms for the basic arithmetic operations and various elementary functions.

This framework has been implemented in C++ and Java. In practice, the integer sizes have the same order of magnitude as the precision requested and the performance is promising.

Some of the algorithms can be extended over the whole of $[0,\infty]$ by using well known identities, such as $\sin(x) = \sin(\pi - x)$. Others can be similarly extended by prefixing normal products with a finite number of integer coefficient lft's. In fact the latter technique can be used to extend the whole framework to $\mathbb{R}^{\infty}$.

Consider for example the natural cover of $\mathbb{R}^{\infty}$ by the four intervals $[0,\infty]$, $[1,-1]$, $[\infty,0]$ and $[-1,1]$. Four lft's map each of the above intervals to the interval $[0,\infty]$. A real number can be located in one of the above quarters in finite time. Therefore, by using one of the four lft's above, its computation can be eventually made in the interval $[0,\infty]$.

Comparison of real numbers may be implemented using the quasi-relational comparison operator $<_\epsilon$ described by Boehm and Cartwright [2].

Finally, we note that this framework may take intervals as inputs and produce intervals as outputs. Therefore, it may be used for verification of numerical algorithms.

References

[1] A. Avizienis. Signed-digit number representations for fast parallel arithmetic. *IRE Transactions on electronic computers*, (10):389–400, 1961.

[2] H. Boehm and R. Cartwright. Exact real arithmetic: Formulating real numbers as functions. In D. Turner, editor, *Research Topics in Functional Programming*, pages 43–64. Addison-Wesley, 1990.

[3] C. Brezinski. *History of Continued Fractions and Padé Approximants*, volume 12 of *Springer series in Computational mathematics*. Springer-Verlag, 1991.

[4] M. H. Escardó. PCF extended with real numbers. *Theoretical Comput. Sci.*, 162(1):79–115, August 1996.

[5] L. Euler. Commentatio in fractionem continuam qua illustris La Grange potestates binomiales expressit. *Mémoires Acad. impér. Sci. Petersb.*, 6:3–11, 1813–1814.

[6] L. Euler. An Essay on Continued Fractions. *Maths. Systems Theory*, 18:295–328, 1985.

[7] P. D. Gianantonio. Real Number Computability and Domain Theory. In *Proceedings of the* 18^{th} *International Symposium on Mathematical Foundations of Computer Science*, pages 413–422, Gdansk, Poland, September 1993. LNCS 711.

[8] R. W. Gosper. Continued Fraction Arithmetic. Technical Report HAKMEM Item 101B, MIT AI MEMO 239, MIT, February 1972. Available from `ftp://ftp.netcom.com/pub/hb/hbaker/hakmem`.

[9] A. Grzegorczyk. On the definition of computable real continuous functions. *Fund. Math.*, 44:61–77, 1957.

[10] G. Hardy. *Ramanujan's Collected Papers*. Chelsea, New York, 1962.

[11] S. L. P. Jones. Arbitrary precision arithmetic using continued fractions, 1984. INDRA Note 1530, University College London.

[12] D. E. Knuth. *The Art of Computer Programming: Seminumerical Algorithms*, volume 2. Addison-Wesley, 1981.

[13] P. Kornerup and D. W. Matula. An On-line Arithmetic Unit for Bit-Pipelined Rational Arithmetic. *Journal of Parallel and Distributed Computing*, 5(3):310–330, 1988.

[14] P. Kornerup and D. W. Matula. Exploiting Redundancy in Bit-Pipelined Rational Arithmetic. In *Proceedings of the* 9^{th} *IEEE Symposium on Computer Arithmetic*, pages 119–126, Santa Monica, 1989. IEEE Computer Society Press.

[15] P. Kornerup and D. W. Matula. An Algorithm for Redundant Binary Bit-Pipelined Rational Arithmetic. *IEEE Transactions on Computers*, C-39(8):1106–1115, 1990.

[16] P. Kornerup and D. W. Matula. Finite Precision Lexicographic Continued Fraction Number Systems. In *Proceedings of the* 7^{th} *IEEE Symposium on Computer Arithmetic*, pages 207–214, Urbana, 1995. IEEE Computer Society Press.

[17] V. Menissier-Morain. Arbitrary precision real arithmetic: design and algorithms. submitted to J. Symbolic Computation, 1996.

[18] R. Moore. *Interval Analysis.* Prentice-Hall, Englewood Cliffs, 1966.

[19] A. M. Nielsen and P. Kornerup. MSB-First Digit Serial Arithmetic. *Journal of Universal Computer Science*, 1(7):527–547, July 1995.

[20] O. Perron. *Die Lehre von den Kettenbrüchen*, volume 1. B. G. Teubner Verlagsgesellschaft, Stuttgart, 1954.

[21] P. J. Potts. Computable Real Arithmetic using Linear Fractional Transformations, June 1996. Early draft PhD Thesis, Imperial College, available from `http://www-tfm.doc.ic.ac.uk/~pjp`.

[22] A. Pringsheim. Ueber ein Convergenz-Kriterium für die Kettenbrüche mit positiven Gliedern. *Sitzungsber. der math.-phys.*, 29:261–268, 1899.

[23] J. Vuillemin. Exact real computer arithmetic with continued fractions. *IEEE Transactions on computers*, 39(8):1087–1105, August 1990.

[24] H. S. Wall. *Analytic Theory of Continued Fractions.* Chelsea Publishing Company, 1948.

[25] O. Watanuki and M. D. Ercegovac. Error Analysis of Certain Floating-Point On-Line Algorithms. *IEEE Transactions on Computers*, C-32(4):352–358, April 1983.