Electronic Notes in Theoretical Computer Science ?? (1999)
URL: `http://www.elsevier.nl/locate/entcs/volume??.html` 21 pages

How Many Argument Digits are Needed to Produce n Result Digits?

Reinhold Heckmann

FB 14 – Informatik, Universität des Saarlandes
Postfach 151150, D-66041 Saarbrücken, Germany

e-mail: `heckmann@cs.uni-sb.de`

Abstract

In the LFT approach to Exact Real Arithmetic, we study the question how many argument digits are needed to produce a certain number of result digits. We present upper and lower bounds for many simple functions and operations, and for exponential and square root.

1 Introduction

In this paper, we work in an approach to Exact Real Arithmetic where real numbers are represented as potentially infinite streams of information units, called *digits*. Hence, an algorithm to compute a certain expression over real numbers is a device that reads some input streams and produces an output stream. Algorithms like this never terminate, but are considered as satisfactory if they produce any desired number of output digits in finite time, i.e., from a finite number of input digits by a finite number of internal operations.

The (time) efficiency of a real number algorithm indicates how much time $T(n)$ it takes to produce n result digits. It clearly depends on the number of input digits needed to produce the n output digits, and on the complexity of the internal operations that digest the input digits. While we had mainly considered the internal complexity in [4,5], we look for lower and upper bounds for the number of input digits in the paper at hand (input complexity).

As in the previous papers, we work in the framework of *Linear Fractional Transformations* (LFT's) that provide an elegant approach to real number arithmetic [3,13,7,10,8,2]. One-dimensional LFT's are used as digits and to implement basic unary functions, while two-dimensional LFT's provide binary operations such as addition and multiplication, and can be combined to obtain infinite expression trees denoting transcendental functions. In Section 2, we present the LFT approach in some detail. In Section 3, we study the input complexity of unary functions that may be realised by a single one-dimensional

LFT. Then, we turn to two-dimensional LFT's in Section 4, and consider addition and multiplication in particular. Finally, we consider the more complex operations of exponential and square root, providing an analysis of the input complexity of $\sqrt{x}$ for *all* x in $[0, \infty]$, including the two end points 0 and ∞.

2 Exact Real Arithmetic by Linear Fractional Transformations

In this section, we recall the framework of exact real arithmetic via LFT's [3,13,7]. In contrast to previous work of the author [4,6,5], we do not follow exactly the version used by the group of Edalat and Potts at Imperial College [10,8,9,12,2], but change the *base interval* from $[0, \infty]$ to $[-1, 1]$. The reasons for this change and its pros and cons are discussed in Section 2.8 after the basic notions have been introduced.

2.1 LFT's and Matrices

Linear Fractional Transformations (LFT's) are functions $x \mapsto \frac{ax+c}{bx+d}$ from reals to reals, parameterised by numbers a, b, c, and d. In general, these numbers may be reals, but in this paper, we shall only consider LFT's with integer parameters, as it is usually done in practical implementations of exact real arithmetic.

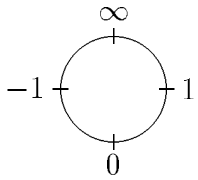
More precisely, LFT's are functions from $\mathbb{R}^{\star}{}_{\perp}$ to $\mathbb{R}^{\star}{}_{\perp}$ where $\mathbb{R}^{\star} = \mathbb{R} \cup \{\infty\}$ is the one-point compactification of the real line, and $\mathbb{R}^{\star}{}_{\perp} = \mathbb{R}^{\star} \cup \{\perp\}$ is $\mathbb{R}^{\star}$ plus an additional 'undefined' value $\perp$. The value ∞ arises as $r/0$ with $r \neq 0$, and on the other hand, the value of the LFT at ∞ is defined as a/b. The undefined value $\perp$ arises as $0/0$, and the value of any LFT at $\perp$ is $\perp$.

The four parameters of an LFT may be presented as a 2-2-matrix $M = \left(\begin{smallmatrix} a & c \\ b & d \end{smallmatrix}\right)$ of integers, hereafter *matrix*. Thus, any matrix $M = \left(\begin{smallmatrix} a & c \\ b & d \end{smallmatrix}\right)$ denotes an LFT $\langle M \rangle$ with $\langle M \rangle(x) = \frac{ax+c}{bx+d}$. Clearly, common factors of the four entries of the matrix do not matter; the matrices M and kM denote the same LFT if k is a non-zero integer. Thus, we get an equivalence $M \cong kM$ for $k \neq 0$. In particular, $M \cong -M$ holds. As a slight normalisation, we usually present matrices in a way such that the lower right entry is non-negative ($d \geq 0$). In the sequel, we shall identify matrices and LFT's, writing $M(x)$ instead of $\langle M \rangle(x)$.

Multiplication of matrices preserves equivalence and corresponds to composition of LFT's. (Addition of matrices does not preserve equivalence; so it does not lead to a well-defined LFT operation.) LFT's with non-zero determinant (non-singular LFT's) map $\mathbb{R}^{\star}$ to $\mathbb{R}^{\star}$ and are invertible. An integer representation of the inverse is given by

$$(1) \qquad \begin{pmatrix} a & c \\ b & d \end{pmatrix}^{*} = \begin{pmatrix} d & -c \\ -b & a \end{pmatrix} \quad .$$

2.2 *Applying LFT's to Intervals*



The set $\mathbb{R}^\star$ can be visualised as a circle. Intervals $[u, v]$ are anti-clockwise arcs from u to v, e.g., $[0, 1] = \{x \in \mathbb{R} \mid 0 \leq x \leq 1\}$, and $[1, 0] = \{x \in \mathbb{R} \mid 1 \leq x \text{ or } x \leq 0\} \cup \{\infty\}$.

Non-singular LFT's map intervals to intervals: if $\det M > 0$, then $M[u, v]$ is $[Mu, Mv]$, while for $\det M < 0$, we get $M[u, v] = [Mv, Mu]$. For the *base interval* $[-1, 1]$, these formulae specialise to $M[-1, 1] = [\frac{c-a}{d-b}, \frac{c+a}{d+b}]$ for $M = \left(\begin{smallmatrix} a & c \\ b & d \end{smallmatrix}\right)$ with $\det M > 0$, and $[\frac{c+a}{d+b}, \frac{c-a}{d-b}]$ if $\det M < 0$. For singular LFT's, $M[u, v]$ may contain $\perp$.

An LFT $M = \left(\begin{smallmatrix} a & c \\ b & d \end{smallmatrix}\right)$ is *bounded* if it avoids the values ∞ and $\perp$ on $[-1, 1]$, i.e., if $M[-1, 1] \subseteq \mathbb{R}$. This is equivalent to saying that the denominator $bx + d$ avoids 0 for x in $[-1, 1]$, or that $d > |b|$ (under the general assumption $d \geq 0$).

An LFT $M = \left(\begin{smallmatrix} a & c \\ b & d \end{smallmatrix}\right)$ is *refining* if it maps the interval $[-1, 1]$ to itself, i.e., if $M[-1, 1] \subseteq [-1, 1]$. (This notion was studied in [2] for arbitrary intervals.) Refining LFT's are bounded, and bounded LFT's are monotone functions from $[-1, 1]$ to $\mathbb{R}$ (monotone in the sense of analysis, i.e., increasing or decreasing). Such functions attain their extremal values at the two end points of their domain of definition. Hence, a bounded LFT M is refining iff $|M(-1)| = \left|\frac{c-a}{d-b}\right| \leq 1$ and $|M(1)| = \left|\frac{c+a}{d+b}\right| \leq 1$. By boundedness, $d - b > 0$ and $d + b > 0$ hold, and thus, a bounded LFT M is refining iff

$$d - b \geq |a - c| \text{ and } d + b \geq |a + c| \tag{2}$$

(under the general assumption $d \geq 0$). Note that addition of these two conditions yields $2d \geq |a - c| + |a + c| = 2\max(|a|, |c|)$, which implies $d \geq |a|$ and $d \geq |c|$ for refining matrices.

The product of refining matrices is refining again, and the product AB of a bounded matrix A by a refining matrix B is bounded.

An LFT $M = \left(\begin{smallmatrix} a & c \\ b & d \end{smallmatrix}\right)$ is *affine* iff $b = 0$. The non-singular affine LFT's form a subgroup of the group of all non-singular LFT's. An affine LFT $M = \left(\begin{smallmatrix} a & c \\ 0 & d \end{smallmatrix}\right)$ (with $d \geq 0$) is bounded iff $d > 0$ and refining iff $d \geq \max(|a - c|, |a + c|) = |a| + |c|$.

2.3 *Representing Reals by LFT's*

Using the base interval $[-1, 1]$, an infinite stream of non-singular matrices M_0, M_1, ... defines the interval

$$\bigcap_{n=0}^{\infty} (M_0 \cdot M_1 \cdot \ldots \cdot M_n)[-1, 1] \ . \tag{3}$$

The intersection is filtered (decreasing) if the LFT's M_n are refining for all $n > 0$. If almost all M_n are sufficiently "contractive", then the intersection in (3) shrinks to a singleton set. In this case, the stream of matrices or LFT's denotes a unique real number (it *converges*).

Because of the usage of matrix multiplication in (3), we consider a stream of matrices converging to a real number x as a (formal) infinite product, and write $x = \prod_{n=0}^{\infty} M_n$. Many real numbers can be elegantly represented by such infinite products. To control the information flow in computations with reals, it turned out to be useful to convert these representations into a kind of standard form. The group of Edalat and Potts at Imperial College [9,2] proposed such a standard form, where the first matrix M_0 must be one of four *sign matrices*, while the remaining ones are taken from a finite set of *digit matrices*.

The four possible sign matrices correspond to rotations of the unit circle by 0°, 90°, 180°, and 270°. They can be explicitly described as follows:

$$
\begin{aligned}
S^0 &= \begin{pmatrix} 1 & 0 \\ 0 & 1 \end{pmatrix} & S^0[-1,1] &= [-1,1] \\
S^1 &= \begin{pmatrix} 1 & 1 \\ -1 & 1 \end{pmatrix} & S^1[-1,1] &= [0,\infty] \\
S^2 &= \begin{pmatrix} 0 & -1 \\ 1 & 0 \end{pmatrix} & S^2[-1,1] &= [1,-1] \\
S^3 &= \begin{pmatrix} 1 & -1 \\ 1 & 1 \end{pmatrix} & S^3[-1,1] &= [\infty,0]
\end{aligned}
$$

These four matrices form a cyclic group of four elements (if common factors do not matter). The upper indices are in fact exponents, and we have $S^i \cdot S^j = S^{(i+j) \bmod 4}$; in particular, S^1 and S^3 are inverse to each other.

There are many possible sets of digit matrices, one for every base $r > 1$. The implementation of Edalat and Potts [2] uses base $r = 2$. In this paper, we consider integer bases $r \geq 2$.

Fix an integer $r \geq 2$. Every real number in the interval $[-1,1]$ has a representation as $\sum_{n=1}^{\infty} k_n r^{-n}$ with integer digits k_n satisfying $|k_n| < r$. (Digits may be negative [1].) These digits correspond to affine maps $x \mapsto \frac{x+k}{r}$ that are LFT's $A_k^r = \begin{pmatrix} 1 & k \\ 0 & r \end{pmatrix}$, mapping the interval $[-1,1]$ into $[\frac{k-1}{r}, \frac{k+1}{r}]$. These image intervals have length $2/r$ and cover $[-1,1]$. The image intervals coming from successive values of k overlap in a common interval of length $1/r$. This provides the redundancy needed in exact real arithmetic.

In the special case $r = 2$, there are three digit matrices $A_{-1}^2 = \begin{pmatrix} 1 & -1 \\ 0 & 2 \end{pmatrix}$, $A_0^2 = \begin{pmatrix} 1 & 0 \\ 0 & 2 \end{pmatrix}$, and $A_1^2 = \begin{pmatrix} 1 & 1 \\ 0 & 2 \end{pmatrix}$, which map $[-1,1]$ to $[-1,0]$, $[-\frac{1}{2}, \frac{1}{2}]$, and $[0,1]$, respectively.

A product of digit matrices is again a digit matrix (in a bigger base). For instance, $\prod_{i=1}^{n} A_{k_i}^r = A_K^{r^n}$ holds with $K = \sum_{i=1}^{n} k_i \cdot r^{n-i}$.

2.4 Computation by LFT's

LFT's can be used not only to represent real numbers, but also to perform computations with real numbers. Using suitable LFT's $x \mapsto \frac{ax+c}{bx+d}$, basic functions such as $x \mapsto x+1$, $x \mapsto 2x$, and $x \mapsto \frac{1}{x}$ can be easily expressed.

Given a refining matrix M and an argument x represented as a converging stream of matrices, the computation of $M(x)$ is performed by a sequence of *absorptions* and *emissions*. Absorption means that M consumes the first matrix D_1 of x, thereby becoming $M \cdot D_1$, which is refining again. It corresponds to the equality

$$M \cdot (D_1 \cdot D_2 \cdot \ldots) \;=\; (M \cdot D_1) \cdot (D_2 \cdot \ldots) \;. \tag{4}$$

Emission means that M produces one further matrix D of the result, thereby becoming $D^* \cdot M$. It corresponds to

$$(D_1 \cdot \ldots \cdot D_n) \cdot M \;=\; (D_1 \cdot \ldots \cdot D_n \cdot D) \cdot (D^* \cdot M) \;. \tag{5}$$

Emission of D is allowed only if $D^* \cdot M$ is refining, which holds if and only if $M([-1,1]) \subseteq D([-1,1])$. The first matrix to be emitted should be a sign matrix, while the remaining ones should be digit matrices A^r_k of the chosen base r. Because of the built-in redundancy, there are often two, sometimes even three different candidates for emission. In this case, it does not matter which one is emitted.

For small bases such as $r = 2$, it is possible to check for each digit matrix A^r_k individually whether it may be emitted. The paper [5] contains a method to determine suitable values of k for large bases r.

A possible strategy for the computation of $M(x)$ is as follows: emit digits until no further emission is possible, then absorb one digit of x, again emit digits until no longer possible, etc.

2.5 Tensors

To compute sums, products, etc., *two-dimensional LFT's* are employed. They are characterised by 8 integer parameters, and thus can be represented by 2-4-matrices of integers, called *tensors*. A tensor $T = \begin{pmatrix} a & c & e & g \\ b & d & f & h \end{pmatrix}$ denotes the bistrict function $T : \mathbb{R}^\star{}_\perp \times \mathbb{R}^\star{}_\perp \to \mathbb{R}^\star{}_\perp$ given by $T(x,y) = \frac{axy+cx+ey+g}{bxy+dx+fy+h}$. Again, common factors do not matter. Thus, we may assume $h \geq 0$ as a weak normalisation.

For fixed x, the function $T|_x := (y \mapsto T(x,y))$ is an LFT with $T|_x = \begin{pmatrix} ax+e & cx+g \\ bx+f & dx+h \end{pmatrix}$. Similarly, for fixed y, an LFT $T|^y = (x \mapsto T(x,y))$ results where $T|^y = \begin{pmatrix} ay+c & ey+g \\ by+d & fy+h \end{pmatrix}$.

A tensor T is *bounded* if it avoids the values ∞ and $\perp$ on arguments from $[-1,1]$, i.e., if $T([-1,1]^2) \subseteq \mathbb{R}$. In case of $h \geq 0$, this is equivalent to $bxy + dx + fy + h > 0$ for all $x, y \in [-1,1]$. For fixed x and variable $y \in [-1,1]$, the expression $bxy + dx + fy + h = (bx+f)y + (dx+h)$ attains its minimum at $y = 1$ or $y = -1$. Similarly, the resulting expressions $(d+b)x + (h+f)$ and $(d-b)x + (h-f)$ attain their minimum at $x = 1$ or $x = -1$. Hence, boundedness is equivalent to $bxy + dx + fy + h > 0$ for $(x,y) \in \{(1,1),(1,-1),(-1,1),(-1,-1)\}$. The resulting four conditions may be summarised to $h + b > |f + d|$ and $h - b > |f - d|$. For $b = 0$, these two conditions are equivalent to the one condition $h > |d| + |f|$.

A tensor T is *refining* if it maps arguments from $[-1,1]$ to results in $[-1,1]$, i.e., $T([-1,1]^2) \subseteq [-1,1]$. For bounded T and fixed x in $[-1,1]$, $T|_x : [-1,1] \to \mathbb{R}$ is a bounded LFT, therefore monotone (increasing or decreasing), and thus attains its extremal values at 1 and -1. The resulting functions $T|^1$ and $T|^{-1}$ from $[-1,1]$ to $\mathbb{R}$ are again monotone and thus attain their extremal values at 1 and -1, too. This shows that a bounded tensor T is refining iff $|T(x,y)| \leq 1$ for $(x,y) \in \{(1,1),(1,-1),(-1,1),(-1,-1)\}$.

2.6 Computation with Tensors

It is quite obvious that addition, subtraction, multiplication, and division can be represented by suitable tensors [3,13,10,8,9]. Edalat and Potts then found representations for many transcendental functions by infinite tensor expressions of the general shape $f(x) = T_0(x, T_1(x, T_2(x, \ldots)))$. It remains to show how to actually compute $T(x,y)$ for a given integer tensor T [8,9].

Emissions can be done as in the one-dimensional case: in emitting a digit D, tensor T is replaced by $D^* \cdot T$, which is a tensor again. Emission of D is allowed only if $D^* \cdot T$ is refining.

Since digits can be absorbed from both arguments, there are two kinds of *absorptions*: absorption of a digit D from the left argument transforms T into $T \circledL D$, while absorption from the right argument yields $T \circledR D$. Right absorption can be defined by writing a tensor T as a row $(T^{\mathrm{L}},\ T^{\mathrm{R}})$ of two matrices, and specifying

(6) $$(T^{\mathrm{L}},\ T^{\mathrm{R}}) \circledR M \;=\; (T^{\mathrm{L}}M,\ T^{\mathrm{R}}M)\ .$$

To define left absorption, let $T^{\times}$ be T with the two middle columns exchanged. Then let $T \circledL M = (T^{\times} \circledR M)^{\times}$, and so

(7) $$(T \circledL M)^{\times} = T^{\times} \circledR M \qquad \text{and} \qquad (T \circledR M)^{\times} = T^{\times} \circledL M\ .$$

2.7 Changing the Base Interval

The work of Peter Potts [10,11] contains many infinite expressions representing constants and transcendental functions. These expressions cannot be taken over immediately if the base interval is changed. The reason is that the base interval is an implicit part of the definition of the meaning of any infinite expression. For instance, if the base interval is $[0,\infty]$, an infinite stream of non-singular matrices M_0, M_1, ... defines the interval

(8) $$\bigcap_{n=0}^{\infty}(M_0 \cdot M_1 \cdot \ldots \cdot M_n)[0,\infty]\ .$$

Since the sign matrices $S^1 : [-1,1] \to [0,\infty]$ and $S^3 : [0,\infty] \to [-1,1]$ are mutually inverse bijections, this can be translated to $[-1,1]$ as follows:

$$\bigcap_{n=0}^{\infty}(M_0 M_1 \cdots M_n)[0,\infty] = \bigcap_{n=0}^{\infty}(M_0 M_1 \cdots M_n)(S^1[-1,1])$$

$$= \bigcap_{n=0}^{\infty} (S^1 S^3 M_0 S^1 S^3 M_1 S^1 \cdots M_n S^1)[-1,1]$$

$$= S^1 \left(\bigcap_{n=0}^{\infty} (S^3 M_0 S^1)(S^3 M_1 S^1) \cdots (S^3 M_n S^1)[-1,1] \right)$$

This shows first that in infinite products, the base interval should be made explicit, and second that the change of base interval may be done with the following formula:

$$\prod_{n=0}^{\infty} M_n[0,\infty] \;=\; S^1 \cdot \prod_{n=0}^{\infty} (S^3 M_n S^1)[-1,1] \tag{9}$$

For instance, [11] contains an infinite product for Euler's number,

$$\mathrm{e} = \prod_{n=0}^{\infty} \begin{pmatrix} 2n+2 & 2n+1 \\ 2n+1 & 2n \end{pmatrix} [0,\infty]$$

which becomes

$$\mathrm{e} = S^1 \prod_{n=0}^{\infty} \begin{pmatrix} 0 & 1 \\ 1 & 4n+2 \end{pmatrix} [-1,1]$$

by formula (9) and cancelling a common factor of 2. In calculations as above, it is useful to know that

$$S^3 \cdot \begin{pmatrix} a & c \\ b & d \end{pmatrix} \cdot S^1 \;=\; \begin{pmatrix} a-b-c+d & a-b+c-d \\ a+b-c-d & a+b+c+d \end{pmatrix} \;. \tag{10}$$

Formula (9) also applies to infinite tensor expressions. For this, one has to understand that $f(x) = T_0(x, T_1(x, \ldots))$ really means $f(x) = \prod_{n=0}^{\infty} (T_n|_x)(I_0)$ for the chosen base interval I_0. By (9), we obtain

$$\prod_{n=0}^{\infty} (T_n|_x)[0,\infty] = S^1 \prod_{n=0}^{\infty} (S^3 \cdot T_n|_x \cdot S^1)[-1,1] = S^1 \prod_{n=0}^{\infty} (S^3 \cdot T_n \circledR S^1)|_x[-1,1] \;.$$

For instance, [11] contains the tensor expression

$$\mathrm{e}^{S^3 x} \;=\; \prod_{n=0}^{\infty} \left. \begin{pmatrix} 2n+2 & 2n+1 & 2n & 2n+1 \\ 2n+1 & 2n & 2n+1 & 2n+2 \end{pmatrix} \right|_x [0,\infty]$$

for x in $[0,\infty]$. With the formula above, this becomes

$$\mathrm{e}^{S^3 x} \;=\; S^1 \prod_{n=0}^{\infty} \left. \begin{pmatrix} 0 & 1 & 0 & -1 \\ 1 & 4n+2 & -1 & 4n+2 \end{pmatrix} \right|_x [-1,1]$$

for x in $[0,\infty]$. (In doing this calculation, note that the left half $(S^3 \cdot T \circledR S^1)^{\mathrm{L}}$ of a transformed tensor is $S^3 \cdot T^{\mathrm{L}} \cdot S^1$, and analogously with the right half. This means that a tensor is transformed by applying Formula (10) to its left and right half separately.)

As a final step, one may substitute x in $[0,\infty]$ by $S^1 u$ with u in $[-1,1]$. Using $T|_{M(u)} = (T \circledcirc M)|_u$, this yields

$$\mathrm{e}^u \;=\; S^1 \prod_{n=0}^{\infty} \begin{pmatrix} 0 & 1 & 0 & 0 \\ 1 & 0 & 0 & 4n+2 \end{pmatrix}\Bigg|_u [-1,1]$$

for u in $[-1,1]$.

2.8 The Choice of the Base Interval

In this section, we discuss the advantages and disadvantages of choosing the base interval $[-1,1]$ instead of $[0,\infty]$.

First, in the presented framework that includes ∞ as a number and employs the four sign matrices S^0 through S^3, there is not much difference between the two base intervals: Because S^1 and S^3 are bijections between $[-1,1]$ and $[0,\infty]$, everything what can be done with $[0,\infty]$ can also be done with $[-1,1]$ after some transformation, as shown in the previous section.

Edalat and Potts name two advantages of $[0,\infty]$: First, the image $M[0,\infty]$ of $[0,\infty]$ under a non-singular LFT $M = \left(\begin{smallmatrix} a & c \\ b & d \end{smallmatrix}\right)$ can be easily obtained from the entries of M: $M[0,\infty] = [\frac{c}{d}, \frac{a}{b}]$ if $\det M > 0$, and $[\frac{a}{b}, \frac{c}{d}]$ if $\det M < 0$. In contrast, the calculation of $M[-1,1]$ requires some computations.

Second, a matrix or tensor is refining for $[0,\infty]$ iff all its entries are non-negative and all its column sums are positive (if the matrix or tensor is weakly normalised so that the sum of its entries is non-negative). This condition is much simpler than the condition we have derived for refinement w.r.t. $[-1,1]$ in Sections 2.2 and 2.5.

For the practical implementation, this means that the test whether a certain matrix can be emitted, which basically is a test for refinement, is more complex in case of $[-1,1]$ than in case of $[0,\infty]$. Yet on the other hand, the base interval $[-1,1]$ comes with digit matrices $A^r_k = \left(\begin{smallmatrix} 1 & k \\ 0 & r \end{smallmatrix}\right)$, while with base interval $[0,\infty]$, one uses the slightly more complex digit matrices $D^r_k = S^1 A^r_k S^3 = \left(\begin{smallmatrix} r+k+1 & r+k-1 \\ r-k-1 & r-k+1 \end{smallmatrix}\right)$. Mainly because of the zero entry in A^r_k, performing absorptions and emissions of digit matrices is more efficient in case of $[-1,1]$ than in case of $[0,\infty]$.

Because of these contradicting effects, it is hard to say which base interval is more efficient in practice. However, in my opinion base interval $[-1,1]$ has clear advantages on the theoretical side. It avoids the troublesome value ∞ that poses difficulties in algebraic transformations and size estimations. Furthermore, one may work with the standard metric and standard derivatives in $[-1,1]$, while working with $[0,\infty]$ excludes the standard metric. In fact, [2,11,6] use a metric on $[0,\infty]$ that is derived from the standard metric on $[-1,1]$. Here, working in $[-1,1]$ directly drastically simplifies the reasoning.

There is another important point in favour of $[-1,1]$. Many people prefer to investigate and/or implement Exact Real Arithmetic in $\mathbb{R}$ instead of $\mathbb{R}^\star = \mathbb{R} \cup \{\infty\}$. The main argument in favour of $\mathbb{R}^\star$ is the totality of reciprocal $x \mapsto \frac{1}{x}$

and indeed any non-singular LFT in $\mathbb{R}^\star$, while reciprocal is undefined for 0 if $\mathbb{R}$ is used. On the other hand, other important operations such as addition, multiplication, maximum, and exponential are total in $\mathbb{R}$, but undefined for certain argument (pairs) involving ∞.

Now, the point is that the usage of the base interval $[0, \infty]$ excludes any direct application of the LFT approach to $\mathbb{R}$-arithmetic. With the base interval $[-1, 1]$ however, the LFT approach can cover $\mathbb{R}$-arithmetic as well as $\mathbb{R}^\star$-arithmetic—one only has to change the sign matrices, while the digit matrices A_k^r are unchanged.

Any x in $\mathbb{R}$ can be represented as $r^k \cdot u$ where r is the chosen base, k is a non-negative integer, and u is in $[-1, 1]$ (*mantissa-exponent approach*). The corresponding LFT representation replaces the sign matrices S^i $(i = 0, \ldots, 3)$ by $\left(\begin{smallmatrix} r^k & 0 \\ 0 & 1 \end{smallmatrix}\right)$ for $k \geq 0$. As a slight variation, one may also admit negative exponents $k < 0$, i.e., matrices $\left(\begin{smallmatrix} 1 & 0 \\ 0 & r^{-k} \end{smallmatrix}\right)$.

Alternatively, any x in $\mathbb{R}$ can be represented as $k + u$ where k is an integer and u is in $[-1, 1]$ (*integer part/fractional part approach*). The corresponding LFT representation replaces the sign matrices by $\left(\begin{smallmatrix} 1 & k \\ 0 & 1 \end{smallmatrix}\right)$ for integers k. Summarising, the base interval $[-1, 1]$ opens the way for using LFT's in $\mathbb{R}$-arithmetic, and for comparing $\mathbb{R}$-arithmetic and $\mathbb{R}^\star$-arithmetic within the common framework of LFT's.

3 Contractivity and Expansivity of Matrices

In this section, we derive lower and upper bounds for the length of the interval $M(I)$ from the length of the interval $I \subseteq [-1, 1]$ and the data contained in the bounded matrix M. These bounds can be used to derive bounds for the number of absorptions required to guarantee a certain number of emissions.

3.1 Definition and Basic Properties

A bounded matrix $M = \left(\begin{smallmatrix} a & c \\ b & d \end{smallmatrix}\right)$ induces a function $M : [-1, 1] \to \mathbb{R}$ defined by $M(x) = \frac{ax+c}{bx+d}$. Its derivative is given by

$$M'(x) \;=\; \frac{ad - bc}{(bx + d)^2} \;. \tag{11}$$

The *contractivity* of M is the supremum of $|M'(x)|$ for x in $[-1, 1]$. For bounded M, the contractivity is finite and given by the formula

$$\operatorname{con} M \;=\; \sup_{x \in [-1,1]} |M'(x)| \;=\; \frac{|ad - bc|}{(d - |b|)^2} \;. \tag{12}$$

Note that this formula is different from the one in [6,5], the reason being the change of base interval. The contractivity assigned to a matrix M in those papers is the same as $\operatorname{con}(S^3 M S^1)$ defined here; see also Section 2.7.

The name contractivity comes from the fact that $|M(x) - M(y)| \leq \operatorname{con} M \cdot |x - y|$ holds for x, y in $[-1, 1]$ by the mean value theorem. Thus, we get an upper bound for the length $\ell(M(I))$ of the image interval $M(I)$:

$$\ell(M(I)) \leq \operatorname{con} M \cdot \ell I \tag{13}$$

for bounded M and intervals $I \subseteq [-1,1]$.

Instead of upper bounds, we may compute lower bounds as well, getting the *expansivity* of M defined by

$$\exp M \;=\; \inf_{x\in[-1,1]} |M'(x)| \;=\; \frac{|ad-bc|}{(d+|b|)^2} \tag{14}$$

with the property

$$\ell(M(I)) \geq \exp M \cdot \ell I \tag{15}$$

for bounded M and intervals $I \subseteq [-1,1]$.

If A is bounded and B is refining, the matrix product (LFT composition) of A and B is bounded again, and the chain rule for compositions gives $(A \cdot B)'(x) = A'(B(x)) \cdot B'(x)$, whence $\operatorname{con}(A \cdot B) \leq \operatorname{con} A \cdot \operatorname{con} B$ and $\exp(A \cdot B) \geq \exp A \cdot \exp B$.

Affine bounded LFT's $A = \left(\begin{smallmatrix} a & c \\ 0 & d \end{smallmatrix}\right)$ have a constant derivative of size a/d. Hence, we get $\operatorname{con} A = \exp A = |a|/d$, and the equalities $\operatorname{con}(A \cdot B) = \operatorname{con} A \cdot \operatorname{con} B$ and $\exp(A \cdot B) = \exp A \cdot \exp B$ for affine A.

3.2 Absorptions and Emissions

How do contractivity and expansivity change during absorptions and emissions? The digit matrices $A^r_k = \left(\begin{smallmatrix} 1 & k \\ 0 & r \end{smallmatrix}\right)$ are affine with $\operatorname{con} A^r_k = \exp A^r_k = \frac{1}{r}$. Hence, we have $\operatorname{con}(M \cdot A^r_k) \leq \operatorname{con} M \cdot \operatorname{con} A^r_k = \frac{1}{r} \operatorname{con} M$ and $\exp(M \cdot A^r_k) \geq \frac{1}{r} \exp M$.

The inverse of A^r_k is $\left(\begin{smallmatrix} r & -k \\ 0 & 1 \end{smallmatrix}\right)$ which is affine with derivative r, and so, $\operatorname{con}((A^r_k)^* \cdot M) = r \operatorname{con} M$ and $\exp((A^r_k)^* \cdot M) = r \exp M$ follows.

Contractivity and expansivity can be used to get information about possible emissions. If M is a refining matrix with $\operatorname{con} M \leq \frac{1}{2r}$, then $\ell(M[-1,1]) \leq \operatorname{con} M \cdot \ell[-1,1] \leq \frac{1}{r}$, and therefore, $M[-1,1] \subseteq A^r_k[-1,1]$ for some k since these intervals have length $2/r$ and overlap in pieces of length $1/r$. The inclusion $M[-1,1] \subseteq A^r_k[-1,1]$ means that digit matrix A^r_k may be emitted; the result $(A^r_k)^* \cdot M$ after emission will be refining again.

If M is refining with $\operatorname{con} M \leq \frac{1}{2r^2}$, then a digit may be emitted by the arguments above, and afterwards, $\operatorname{con}((A^r_k)^* \cdot M) \leq \frac{1}{2r}$, whence a second digit may be emitted. Iterating these arguments, we see

- If M is a refining matrix with $\operatorname{con} M \leq \frac{1}{2r^n}$ for $n \geq 0$, then at least n digits in base r may be emitted from M.

From this statement, we easily obtain that a refining M may emit at least $-\lceil \log_r(2 \operatorname{con} M) \rceil$ digits.

On the other hand, every refining matrix M has $M[-1,1] \subseteq [-1,1]$, and therefore, $2 \geq \ell(M[-1,1]) \geq \exp M \cdot \ell[-1,1] = 2 \exp M$, which implies $\exp M \leq 1$. Now, if M is a refining matrix that may emit n digits, then the matrix $\tilde{M}$ resulting from these emissions is still refining, and so, $\exp \tilde{M} = r^n \cdot \exp M \leq 1$, or $\exp M \leq \frac{1}{r^n}$. This gives a weak converse of the statement above:

- If M is a refining matrix that may emit at least n digits in base r, then $\exp M \leq \frac{1}{r^n}$ holds.

By contraposition, we see that if $\exp M > \frac{1}{r^n}$, then at most $n-1$ digits can be emitted. Here, the hypothesis is equivalent to $n > -\lceil \log_r \exp M \rceil$. For $n = -\lceil \ldots \rceil + 1$, an upper bound of $-\lceil \ldots \rceil$ emitted digits results. Thus, we obtain by combining both sides:

- A refining matrix M may emit at least $-\lceil \log_r (2 \operatorname{con} M) \rceil$ and at most $-\lceil \log_r (\exp M) \rceil$ digits in base r.

If m digits from an argument x in $[-1, 1]$ are absorbed into M, the resulting matrix $\tilde{M}$ has $\operatorname{con} \tilde{M} \leq r^{-m} \operatorname{con} M$ and $\exp \tilde{M} \geq r^{-m} \exp M$. Inserting this in the statement above yields

- After absorbing m digits from an argument x in $[-1, 1]$, a refining matrix M may emit at least $m - \lceil \log_r (2 \operatorname{con} M) \rceil$ and at most $m - \lceil \log_r (\exp M) \rceil$ digits in base r.

This statement provides lower and upper bounds for the number of emitted digits depending on fixing a number m of absorbed digits. The statement may easily be turned around to indicate how many digits must be absorbed in order to produce a fixed number n of emitted digits:

Theorem 3.1 *Let M be a refining matrix applied to an argument x in $[-1, 1]$. In order to emit n digits of $M(x)$ in base r, at least $n + \lceil \log_r (\exp M) \rceil$ and at most $n + \lceil \log_r (2 \operatorname{con} M) \rceil$ digits from x must be absorbed.*

The lower bound was already contained in [11], and the upper bound appeared in [5] in the less explicit form "at most $n + k$ for some constant k".

For affine M, $\exp M = \operatorname{con} M$ holds, and thus, the two bounds in Theorem 3.1 are close together: For base 2, they always differ by one, while for large bases, they are even identical in most cases, allowing the exact prediction of the number of required absorptions. For non-affine M, $\exp M$ and $\operatorname{con} M$ may differ considerably, leading to less accurate estimations.

3.3 Working with Sign Matrices

Theorem 3.1 deals with the case of a refining matrix M applied to an argument of $[-1, 1]$ which means that the result is back in $[-1, 1]$, so that no sign matrices are needed.

Now, we change to the general case of an arbitrary non-singular matrix M applied to an arbitrary argument x from $\mathbb{R}^\star$. Since M maps $\mathbb{R}^\star$ onto $\mathbb{R}^\star$, nothing can be said about the result without consulting the argument. Thus, the first step is the absorption of the sign matrix of x into the matrix M.

After doing this, we are in the case of applying a non-singular matrix M to an *unsigned* argument u from $[-1, 1]$.

To see what happens in this case, one has to check which of the four special values $y_0 = S^0 0 = 0$, $y_1 = S^1 0 = 1$, $y_2 = S^2 0 = \infty$, and $y_3 = S^3 0 = -1$ are in the image $M(-1, 1)$ of the *interior* of the base interval. Technically, $y_i \in M(-1, 1)$ can be checked by testing $M^* y_i \in (-1, 1)$.

If $M(-1, 1)$ contains at most one of the four values y_0 through y_3, then $M[-1, 1] \subseteq S^k[-1, 1]$ for some k, and thus, S^k can be emitted. The resulting matrix $S^{-k} \cdot M$ is refining, and we are back in the case of Theorem 3.1.

If $M(-1, 1)$ contains more than one of the special values y_0 through y_3, then $M[-1, 1]$ is not contained in any of the four intervals $S^k[-1, 1]$, i.e., no sign matrix can be emitted. In this case, digits from the arguments must be absorbed until eventually the emission of a sign matrix will become possible. This will always happen after a bounded amount of absorptions, as the following argument shows.

Let $x_i = M^* y_i$, and let $X = \{x_i (0 \leq i \leq 3) \mid x_i \in (-1, 1)\}$. By the hypothesis of this case, X has at least two elements. Let $\delta = \min\{|x - x'| \mid x, x' \in X, x \neq x'\}$ be the minimum distance of two elements from X.

Let $m \geq 0$ be an integer such that $2r^{-m} \leq \delta$. After absorbing m digits from the argument, matrix M has changed into $\tilde{M} = M \cdot A$, where A is the product of the m digit matrices. Now, $y_i \in \tilde{M}(-1, 1)$ iff $x_i \in A(-1, 1)$. Since A is affine with derivative r^{-m}, $A(-1, 1)$ is an open interval of length $2r^{-m} \leq \delta$. Since δ is the minimum distance between elements of X, $A(-1, 1)$ contains at most one of the x_i, hence $\tilde{M}(-1, 1)$ contains at most one of the y_i, and thus $\tilde{M}$ may emit a sign matrix. This shows that at most $m = -\lfloor \log_r(\delta/2) \rfloor$ argument digits must be absorbed before the sign matrix of the result can be emitted. After this, we are back in the case of Theorem 3.1.

Summarising, the arguments of this section yield the following theorem:

Theorem 3.2 *Let M be a non-singular matrix applied to an argument x in $\mathbb{R}^\star$. In order to emit $n > 0$ digits of $M(x)$ in base r, at least the sign matrix of x, and at most $n + k$ further digits from x must be absorbed, where k is an integer constant independent from x.*

3.4 Case Study: Doubling

As a case study, let us consider the computation of $2x$ from x, i.e., the application of $M = \left(\begin{smallmatrix} 2 & 0 \\ 0 & 1 \end{smallmatrix}\right)$ to x, using base $r = 2$. While this computation is a triviality in the mantissa-exponent approach of $\mathbb{R}$-arithmetic, a more elaborate analysis is needed in case of $\mathbb{R}^\star$-arithmetic. We distinguish four cases according to the sign matrix of the argument x.

Case $x = S^0(u)$:
After absorbing $S^0 = \left(\begin{smallmatrix} 1 & 0 \\ 0 & 1 \end{smallmatrix}\right)$ into M, we get $M_0 = M = \left(\begin{smallmatrix} 2 & 0 \\ 0 & 1 \end{smallmatrix}\right)$. The four key values are $x_0 = M_0^*(0) = 0$, $x_1 = M_0^*(1) = \frac{1}{2}$, $x_2 = M_0^*(\infty) = \infty$, and $x_3 = M_0^*(-1) = -\frac{1}{2}$. From these four values, three (x_0, x_1, and x_3) are

in $(-1,1)$. Hence, no sign matrix can be emitted from M_0. The minimum distance is $\delta = \frac{1}{2}$, and therefore, theory says that at most $-\lfloor \log_2(\frac{1}{4}) \rfloor = 2$ digits of u must be absorbed to guarantee the emission of the sign matrix of $2x$. In fact, the following case analysis shows that one digit suffices.

Subcase $x = S^0(A^2_{-1}(v))$:
$M_0 \cdot A^2_{-1}$ is $M_{0,-1} = \left(\begin{smallmatrix} 1 & -1 \\ 0 & 1 \end{smallmatrix}\right)$ after cancelling a factor of 2. For this matrix, the four key values are $x_0 = \left(\begin{smallmatrix} 1 & 1 \\ 0 & 1 \end{smallmatrix}\right)(0) = 1$, $x_1 = 2$, $x_2 = \infty$, and $x_3 = 0$. Since only x_3 is in $(-1,1)$, sign matrix S^3 can be emitted. The resulting matrix is $S^1 \cdot M_{0,-1} = \left(\begin{smallmatrix} 1 & 1 \\ -1 & 1 \end{smallmatrix}\right) \cdot \left(\begin{smallmatrix} 1 & -1 \\ 0 & 1 \end{smallmatrix}\right) = \left(\begin{smallmatrix} 1 & 0 \\ -1 & 2 \end{smallmatrix}\right)$. This matrix is refining with $\exp = 2/9$ and $\mathrm{con} = 2$. Theorem 3.1 says that from now on, between $n-2$ and $n+2$ further absorptions are needed to obtain n result digits.

Subcase $x = S^0(A^2_0(v))$:
$M_0 \cdot A^2_0$ is the identity matrix after cancelling a factor of 2. This matrix is refining so that the sign matrix S^0 can be emitted. Since $\exp = \mathrm{con} = 1$, Theorem 3.1 says that n or $n+1$ absorptions are needed for n emissions. Of course, we know that n absorptions are sufficient in case of the identity matrix.

Subcase $x = S^0(A^2_1(v))$ is symmetric to $x = S^0(A^2_{-1}(v))$.

Case $x = S^1(u)$:
After absorbing $S^1 = \left(\begin{smallmatrix} 1 & 1 \\ -1 & 1 \end{smallmatrix}\right)$ into M, we get $M_1 = \left(\begin{smallmatrix} 2 & 2 \\ -1 & 1 \end{smallmatrix}\right)$ with $M_1^* = \left(\begin{smallmatrix} 1 & -2 \\ 1 & 2 \end{smallmatrix}\right)$. The four key values are $x_0 = M_1^*(0) = -1$, $x_1 = M_1^*(1) = -\frac{1}{3}$, $x_2 = M_1^*(\infty) = 1$, and $x_3 = M_1^*(-1) = -3$. Since only x_1 is in $(-1,1)$, sign matrix S^1 can be emitted, giving $S^3 \cdot M_1 = \left(\begin{smallmatrix} 3 & 1 \\ 1 & 3 \end{smallmatrix}\right)$. This matrix is refining with $\exp = \frac{1}{2}$ and $\mathrm{con} = 2$. Hence, Theorem 3.1 tells that between $n-1$ and $n+2$ absorptions are needed to obtain n result digits.

Case $x = S^2(u)$:
After absorbing $S^2 = \left(\begin{smallmatrix} 0 & -1 \\ 1 & 0 \end{smallmatrix}\right)$ into M, we get $M_2 = \left(\begin{smallmatrix} 0 & -2 \\ 1 & 0 \end{smallmatrix}\right)$ with $M_2^* = \left(\begin{smallmatrix} 0 & 2 \\ 1 & 0 \end{smallmatrix}\right)$. Hence, the four key values are ∞, 2, 0, -2. Only $x_2 = 0$ is in $(-1,1)$, and so, sign matrix S^2 can be emitted, giving $S^2 \cdot M_2 = \left(\begin{smallmatrix} 1 & 0 \\ 0 & 2 \end{smallmatrix}\right)$ (up to a common factor of -1). Since this matrix is refining with $\exp = \mathrm{con} = \frac{1}{2}$, Theorem 3.1 tells that $n-1$ or n absorptions are needed for n emissions. Because $\left(\begin{smallmatrix} 1 & 0 \\ 0 & 2 \end{smallmatrix}\right)$ is the digit matrix A^2_0, we know that exactly $n-1$ absorptions are needed for n emissions (the first emission is A^2_0, producing the identity matrix).

Case $x = S^3(u)$ is symmetric to $x = S^1(u)$.

This analysis shows that for $M = \left(\begin{smallmatrix} 2 & 0 \\ 0 & 1 \end{smallmatrix}\right)$, the constant k in Theorem 3.2 is (at most) 3. The worst case is subcase $x = S^0(A^2_{-1}(v))$, where (at most) $n+2$ absorptions are needed for n emissions, which gives $n+3$ since the first digit matrix has already been absorbed to get into this subcase.

4 Contractivity of Tensors

As in the case of matrices, the notion of contractivity of a tensor is used to obtain upper bounds for the number of absorptions needed to obtain a certain number of emissions.

4.1 Definition and Simple Properties

A main property of the contractivity of matrices is $|M(x) - M(y)| \leq \operatorname{con} M \cdot |x - y|$ for bounded M and x, y in $[-1, 1]$. To obtain a similar property for bounded tensors T, we compute for x_1, x_2, y_1, y_2 in $[-1, 1]$:

$$\begin{aligned} |T(x_1, y_1) - T(x_2, y_2)| &\leq |T(x_1, y_1) - T(x_2, y_1)| + |T(x_2, y_1) - T(x_2, y_2)| \\ &= |T|^{y_1}(x_1) - T|^{y_1}(x_2)| + |T|_{x_2}(y_1) - T|_{x_2}(y_2)| \\ &\leq \operatorname{con}(T|^{y^1}) \cdot |x_1 - x_2| + \operatorname{con}(T|_{x_2}) \cdot |y_1 - y_2| \\ &\leq \operatorname{con}_{\mathrm{L}} T \cdot |x_1 - x_2| + \operatorname{con}_{\mathrm{R}} T \cdot |y_1 - y_2| \end{aligned}$$

where $\operatorname{con}_{\mathrm{L}} T = \sup_{y \in [-1,1]} \operatorname{con}(T|^y)$ and $\operatorname{con}_{\mathrm{R}} T = \sup_{x \in [-1,1]} \operatorname{con}(T|_x)$. As a consequence, we obtain

$$\ell(T(I \times J)) \ \leq \ \operatorname{con}_{\mathrm{L}} T \cdot \ell(I) + \operatorname{con}_{\mathrm{R}} T \cdot \ell(J) \tag{16}$$

for bounded T and intervals $I, J \subseteq [-1, 1]$. In particular, we have

$$\ell(T([-1, 1]^2)) \ \leq \ 2(\operatorname{con}_{\mathrm{L}} T + \operatorname{con}_{\mathrm{R}} T) \ . \tag{17}$$

Thus, if T is refining with $\operatorname{con}_{\mathrm{L}} T + \operatorname{con}_{\mathrm{R}} T \leq \frac{1}{2r}$, then $T([-1, 1]^2)$ is a subinterval of $[-1, 1]$ with length $\leq \frac{1}{r}$, and therefore is contained in $A^r_k[-1, 1]$ for some k, i.e., T may emit digit A^r_k.

Proposition 4.1 *A refining tensor T with $\operatorname{con}_{\mathrm{L}} T + \operatorname{con}_{\mathrm{R}} T \leq \frac{1}{2r}$ may emit at least one digit in base r. Note that the hypothesis is in particular satisfied if $\operatorname{con}_{\mathrm{L}} T$ and $\operatorname{con}_{\mathrm{R}} T$ are below $\frac{1}{4r}$.*

4.2 Absorptions and Emissions

To obtain the desired estimations, it is important to know how $\operatorname{con}_{\mathrm{L}}$ and $\operatorname{con}_{\mathrm{R}}$ are affected by absorptions and emissions. Here, we concentrate on $\operatorname{con}_{\mathrm{R}}$; the proofs for $\operatorname{con}_{\mathrm{L}}$ are analogous.

For bounded $T = \begin{pmatrix} a & c & e & g \\ b & d & f & h \end{pmatrix}$, we have $T|_x = \begin{pmatrix} ax+e & cx+g \\ bx+f & dx+h \end{pmatrix}$, and thus $\operatorname{con}(T|_x) = \frac{|(ax+e)(dx+h) - (bx+f)(cx+g)|}{(dx+h-|bx+f|)^2}$. Unfortunately, it is tedious and difficult to compute $\operatorname{con}_{\mathrm{R}} T = \sup_{x \in [-1,1]} \operatorname{con}(T|_x)$ from this expression. The reason is that the expression is not necessarily monotonic in x, and therefore, the supremum may be attained at a value different from the two end points $x = 1$ and $x = -1$. At least, it is obvious that the supremum is a well-defined real number (not ∞) since for bounded T, it is the supremum of a continuous real-valued function on a compact interval. (The corresponding result for base interval $[0, \infty]$ would be less obvious.)

Even without having a simple closed formula for $\operatorname{con}_{\mathrm{R}} T$, it is possible to derive its main properties from those of the contractivity of matrices.

Let M be a bounded matrix and T a refining tensor, whence $M \cdot T$ is a bounded tensor. For any x in $[-1, 1]$, we have $(MT)|_x = M \cdot (T|_x)$. Recall $\operatorname{con}(M \cdot (T|_x)) \leq \operatorname{con} M \cdot \operatorname{con}(T|_x)$ with equality if M is affine. Forming the supremum over x in $[-1, 1]$ on both sides, we get $\operatorname{con}_{\mathrm{R}}(M \cdot T) \leq \operatorname{con} M \cdot \operatorname{con}_{\mathrm{R}} T$

with equality for affine M. Specialising M to the affine matrix $(A_k^r)^*$ with contractivity r yields

- If T is a refining tensor and $\tilde{T}$ results from T by emitting a digit in base r, then $\mathrm{con}_\mathrm{R}\,\tilde{T} = r\,\mathrm{con}_\mathrm{R}\,T$ holds.

Now consider absorption from the right. If T is a bounded tensor and M a refining matrix, then $T \circledR M$ is bounded. For any x in $[-1, 1]$, we have $(T \circledR M)|_x = T|_x \cdot M$, whence $\mathrm{con}((T \circledR M)|_x) \le \mathrm{con}(T|_x) \cdot \mathrm{con}\,M$. Forming the supremum over x in $[-1, 1]$ on both sides, we get $\mathrm{con}_\mathrm{R}(T \circledR M) \le \mathrm{con}_\mathrm{R}\,T \cdot \mathrm{con}\,M$. Specialising M to the digit matrix A_k^r with contractivity $\frac{1}{r}$ yields

- If T is a refining tensor and $\tilde{T}$ results from T by absorbing a digit in base r from the right, then $\mathrm{con}_\mathrm{R}\,\tilde{T} \le \frac{1}{r}\,\mathrm{con}_\mathrm{R}\,T$ holds.

For absorption from the left, note that $(T \circledL M)|_x = T|_{M(x)}$. Hence, $\mathrm{con}_\mathrm{R}(T \circledL M) = \sup_{x \in M[-1,1]} \mathrm{con}(T|_x) \le \mathrm{con}_\mathrm{R}\,T$ if $M[-1, 1] \subseteq [-1, 1]$, i.e., if M is refining. Digit matrices are refining, and so we get:

- If T is a refining tensor and $\tilde{T}$ results from T by absorbing a digit from the left, then $\mathrm{con}_\mathrm{R}\,\tilde{T} \le \mathrm{con}_\mathrm{R}\,T$ holds.

These results are sufficient to derive an upper bound for the number of absorptions needed to lead to the possibility of at least n emissions. Let T be a refining tensor, and let m^R and m^L be two integers such that $\mathrm{con}_\mathrm{R}\,T \le \frac{r^{m^\mathrm{R}}}{4r^n}$ and $\mathrm{con}_\mathrm{L}\,T \le \frac{r^{m^\mathrm{L}}}{4r^n}$. After m^L absorptions from the left and m^R absorptions from the right, we have a tensor $\tilde{T}$ with $\mathrm{con}_\mathrm{R}\,\tilde{T} \le \frac{1}{4r^n}$ and $\mathrm{con}_\mathrm{L}\,\tilde{T} \le \frac{1}{4r^n}$. By Prop. 4.1 and the result about emissions, $\tilde{T}$ can emit at least n digits. These arguments lead to the following result:

Theorem 4.2 *Let T be a refining tensor applied to arguments x and y from $[-1, 1]$. In order to emit n digits of $T(x, y)$ in base r, at most $n + \lceil \log_r(4\,\mathrm{con}_\mathrm{L}\,T) \rceil$ digits from x and $n + \lceil \log_r(4\,\mathrm{con}_\mathrm{R}\,T) \rceil$ digits from y must be absorbed. This gives a total number of at most $3n + \lceil \log_r(4\,\mathrm{con}_\mathrm{L}\,T) \rceil + \lceil \log_r(4\,\mathrm{con}_\mathrm{R}\,T) \rceil$ LFT operations.*

In the following, we consider two specific examples. The first is mean value computation, $T(x, y) = \frac{1}{2}(x + y)$, i.e., $T = \left(\begin{smallmatrix} 0 & 1 & 1 & 0 \\ 0 & 0 & 0 & 2 \end{smallmatrix}\right)$. This tensor is refining with $T|_x = \left(\begin{smallmatrix} 1 & x \\ 0 & 2 \end{smallmatrix}\right)$, whence $\mathrm{con}(T|_x) = \frac{2}{2^2} = \frac{1}{2}$, and thus, $\mathrm{con}_\mathrm{R}\,T = \frac{1}{2}$. By symmetry, we also have $\mathrm{con}_\mathrm{L}\,T = \frac{1}{2}$, and Theorem 4.2 tells that at most $n + 1$ digits from each of x and y are needed to compute n digits of $\frac{1}{2}(x + y)$, giving a total of at most $3n + 2$ LFT operations.

The second example is multiplication: $T(x, y) = xy$, i.e., $T = \left(\begin{smallmatrix} 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 \end{smallmatrix}\right)$, which is refining. We have $T|_x = \left(\begin{smallmatrix} x & 0 \\ 0 & 1 \end{smallmatrix}\right)$, whence $\mathrm{con}(T|_x) = \frac{|x|}{1^2} = |x|$, and thus $\mathrm{con}_\mathrm{R}\,T = 1$. By symmetry, $\mathrm{con}_\mathrm{L}\,T$ is also 1, and therefore, at most $n + 2$ digits from each of x and y are needed to compute n digits of xy, giving a total of at most $3n + 4$ LFT operations.

4.3 Working with Sign Matrices

So far, we have applied refining tensors to arguments from $[-1,1]$ so that no sign matrices were needed. In the case of (non-singular) matrices, we could reduce the general case to the refining case in a constant number of operations.

Unfortunately, this is not possible for tensors. Consider for instance the tensor $T = \left(\begin{smallmatrix} 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \end{smallmatrix}\right)$ for division. This tensor is undefined for the argument pair $(0,0)$ since $T(0,0) = \frac{0}{0} = \perp$. In the sequel, we show that this singularity induces a singularity in complexity; in particular, for $(x,y) \neq (0,0)$, there is no upper bound in the number of argument digits required to emit the first result digit.

Since T is undefined at $(0,0)$, it cannot emit anything without absorbing something. Suppose that the first k digits of x and the first l digits of y are A_0^r. After having absorbed those k digits from the left and l digits from the right, T becomes $\left(\begin{smallmatrix} 0 & r^l & 0 & 0 \\ 0 & 0 & r^k & 0 \end{smallmatrix}\right)$. This tensor is still undefined at $(0,0)$, and thus cannot emit anything without absorbing further argument digits. Therefore, the division tensor T can only start to emit digits after having found the first non-zero digit in at least one argument. For arguments close to $(0,0)$ this can happen arbitrarily late so that there is no uniform upper bound for the complexity of computing even the first digit of $T(x,y)$ for $(x,y) \in [-1,1]^2 \setminus \{(0,0)\}$.

Note that this unfortunate behaviour is not confined to division. It also happens for multiplication near the argument pairs $(0,\infty)$ and $(\infty,0)$, and for addition near (∞,∞).

5 Infinite Tensor Expressions

Consider a function $f : [-1,1] \to [-1,1]$ given by an infinite tensor expression $x \mapsto \prod_{n=0}^{\infty}(T_n|_x)[-1,1]$ where all tensors T_n are refining. If all tensors T_n have the property that at most $k-1$ digits from their right argument are needed to compute k digits of their result, then f is computable: To compute k digits of $\prod_{n=0}^{\infty}(T_n|_x)[-1,1]$, at most $k-1$ digits of $\prod_{n=1}^{\infty}(T_n|_x)[-1,1]$ are needed, hence at most $k-2$ digits of $\prod_{n=2}^{\infty}(T_n|_x)[-1,1]$, etc.; finally, no digits of $\prod_{n=k}^{\infty}(T_n|_x)[-1,1]$ are needed, i.e., the tensors T_k, T_{k+1} etc. need not be considered any more. In the following, we shall consider some specific examples using base $r = 2$.

5.1 Exponential

Recall from Section 2.7 that $\mathrm{e}^x = S^1 \prod_{n=0}^{\infty} T_n|_x[-1,1]$ for x in $[-1,1]$, where $T_n = \left(\begin{smallmatrix} 0 & 1 & 0 & 0 \\ 1 & 0 & 0 & 4n+2 \end{smallmatrix}\right)$. Here, we have

$$T_n|_x = \begin{pmatrix} 0 & x \\ x & 4n+2 \end{pmatrix} \Rightarrow \mathrm{con}(T_n|_x) = \frac{x^2}{(4n+2-|x|)^2} \Rightarrow \mathrm{con}_{\mathrm{R}}\, T_n = \frac{1}{(4n+1)^2}$$

$$T_n|^y = \begin{pmatrix} 1 & 0 \\ y & 4n+2 \end{pmatrix} \Rightarrow \mathrm{con}(T_n|^y) = \frac{4n+2}{(4n+2-|y|)^2} \Rightarrow \mathrm{con}_\mathrm{L}\, T_n = \frac{4n+2}{(4n+1)^2}$$

Applying Theorem 4.2, we see that T_0 needs at most $k+3$ digits from x and $k+2$ digits from T_1 to produce k output digits. The remaining tensors are more efficient: For $n > 0$, T_n needs at most k digits from x and $k-2$ digits from T_{n+1} to produce k output digits. Hence, we get the following scheme for even k:

	emit	absorb from x	absorb from the right
T_0	k	$k+3$	$k+2$
T_1	$k+2$	$k+2$	k
T_2	k	k	$k-2$
T_3	$k-2$	$k-2$	$k-4$
$\vdots$	$\vdots$	$\vdots$	$\vdots$
$T_{\frac{1}{2}k+1}$	2	2	0

Thus, for x in $[-1, 1]$, e^x needs at most $k+3$ digits from x to produce k result digits. For the computation, at most $\frac{3}{4}k^2 + \frac{13}{2}k + 9$ LFT operations are needed (the sum of all numbers in the table).

5.2 Square Root

Using the base interval $[0, \infty]$, $\sqrt{x}$ can be obtained as $\prod_{n=0}^{\infty}(T_n|_x)[0, \infty]$ with $T_n = \left(\begin{smallmatrix} 1 & 1 & 1 & 0 \\ 0 & 1 & 1 & 1 \end{smallmatrix}\right)$ [11] or $T_n = \left(\begin{smallmatrix} 1 & 2 & 1 & 0 \\ 0 & 1 & 2 & 1 \end{smallmatrix}\right)$ [6]. (In both cases, T_n does not depend on n.) Here, we work with the second tensor because it has better contractivity.

Since all the tensors T_n are identical, some work can be saved by considering the infinite tensor expression as a feedback loop $y = T(x, y)$ instead of as an infinite product [11]. This means that it suffices to work with just one instance of the tensor; once a digit A has been emitted from T giving A^*T, it is immediately absorbed into the tensor from the right, giving $A^*T \circledR A$.

Usually, A may be emitted if A^*T is refining. In the case of a feedback loop, this condition may be relaxed to $A^*T \circledR A$ being refining [11]. This means that some emissions may be done earlier, increasing performance.

The tensor expression introduced at the beginning uses base interval $[0, \infty]$. Using the methods of Section 2.7, it can be transformed to the base interval $[-1, 1]$. The result is $\sqrt{S^1 x} = S^1 \prod_{n=0}^{\infty}(T|_x)[-1, 1]$ for x in $[-1, 1]$ with $T = \left(\begin{smallmatrix} 0 & 1 & 0 & 0 \\ -1 & 0 & 0 & 2 \end{smallmatrix}\right)$, i.e., $T(x, y) = \frac{x}{2-xy}$. This tensor is refining with

$$T|_x = \begin{pmatrix} 0 & x \\ -x & 2 \end{pmatrix} \Rightarrow \det(T|_x) = x^2 \text{ and } \mathrm{con}(T|_x) = \frac{x^2}{(2-|x|)^2} \Rightarrow \mathrm{con}_\mathrm{R}\, T = 1$$

$$T|^y = \begin{pmatrix} 1 & 0 \\ -y & 2 \end{pmatrix} \Rightarrow \det(T|^y) = 2 \text{ and } \mathrm{con}(T|^y) = \frac{2}{(2-|y|)^2} \Rightarrow \mathrm{con}_\mathrm{L}\, T = 2$$

Note that both determinants $\det(T|_x)$ and $\det(T|^y)$ are non-negative, and thus, $T(x,y)$ is increasing in x and y.

By Theorem 4.2, T needs at most $k+3$ digits from its left argument and $k+2$ digits from its right argument to produce k result digits. For the evaluation of the feed back loop $y = T(x,y)$, the upper bound for digits from the right must be k. Thus, we do not know so far that the loop can be evaluated for any argument x in $[-1,1]$.

Assume that the given representation of the argument x starts with the digit A_0^2. Absorbing this digit into T yields a tensor $\tilde{T} = \left(\begin{smallmatrix} 0 & 1 & 0 & 0 \\ -1 & 0 & 0 & 4 \end{smallmatrix}\right)$ with $\mathrm{con}(\tilde{T}|_x) = \frac{x^2}{(4-|x|)^2}$ and $\mathrm{con}(\tilde{T}|^y) = \frac{4}{(4-|y|)^2}$, whence $\mathrm{con}_\mathrm{R}\,\tilde{T} = \frac{1}{9}$ and $\mathrm{con}_\mathrm{L}\,\tilde{T} = \frac{4}{9}$. Theorem 4.2 implies that $\tilde{T}$ needs at most $k+1$ digits from its left argument and $k-1$ digits from its right argument to produce k result digits. Thus, the evaluation of the loop is possible in this case. It needs at most $3k$ LFT operations to produce k output digits: $k+1$ absorptions from x, and k emissions with subsequent $k-1$ absorptions from the right.

Going back to the original loop $y = T(x,y)$, we may ask what happens if the given argument does not start with digit A_0^2. A careful analysis shows that the computation is also possible for such arguments, but less efficient than in the case of start digit A_0^2.

First, let us see what function is defined by $y = T(x,y)$. Actually, it is not square root itself, since we have $\sqrt{S^1(x)} = S^1(y)$ for x in $[-1,1]$. Hence, we obtain for $f(x) = y$ the following equation:

$$\begin{aligned} f(x) = S^3\left(\sqrt{S^1(x)}\right) &= \frac{w-1}{w+1} \text{ where } w = \sqrt{\frac{1+x}{1-x}} \\ = \frac{\sqrt{1+x}-\sqrt{1-x}}{\sqrt{1+x}+\sqrt{1-x}} &= \frac{1-\sqrt{1-x^2}}{x} = \frac{x}{1+\sqrt{1-x^2}} \end{aligned}$$

Function f maps $[-1,1]$ onto itself with $f(-1) = -1$, $f(0) = 0$, and $f(1) = 1$. Using the last expression, its derivative can be computed as

$$f'(x) = \frac{1}{(1+\sqrt{1-x^2})\cdot\sqrt{1-x^2}} \ .$$

This shows that f is increasing. Because of $f'(0) = \frac{1}{2}$ and $f'(\pm 1) = +\infty$, we see that f has small slope near 0, but huge slope near the ends ± 1. This is the reason why evaluation of f is more efficient near 0 than near the two end points.

The (partial) correctness of the proposed implementation of f by T can easily be verified by showing that for x, y in $[-1,1]$, $y = f(x)$ holds if and only if $y = T(x,y)$. Recall $\mathrm{con}_\mathrm{R}\,T = 1$, whence the derivative of $T|_x = (y \mapsto T(x,y))$ is at most 1 for all x, y in $[-1,1]$. Thus, $T|_x$ is above the diagonal $y \mapsto y$ for $y < f(x)$ and below for $y > f(x)$. This shows

$$f(x) \le y \Leftrightarrow T(x,y) \le y \quad \text{and} \quad f(x) \ge y \Leftrightarrow T(x,y) \ge y \ . \tag{18}$$

From these statements, we may conclude

$$f(I) \subseteq J \Leftrightarrow T(I, J) \subseteq J \text{ for intervals } I, J \subseteq [-1, 1]. \tag{19}$$

For a proof, let $I = [x_1, x_2]$ and $J = [y_1, y_2]$. The left hand side means $f(x_1) \geq y_1$ and $f(x_2) \leq y_2$. Since T is increasing in both arguments, the right hand side means $T(x_1, y_1) \geq y_1$ and $T(x_2, y_2) \leq y_2$. By (18), both sides are equivalent.

Now consider evaluation of the loop $y = T(x, y)$. Assume that n digits $A^2_{l_1}, \ldots, A^2_{l_n}$ of the result were emitted (and absorbed from the right) while m digits $A^2_{k_1}, \ldots, A^2_{k_m}$ of x were absorbed from the left. This is possible if and only if for all $i \leq n$, there is $j \leq m$ such that

$$(A^2_{l_i})^* \cdots (A^2_{l_1})^* T \circledL \left(A^2_{k_1} \cdots A^2_{k_j}\right) \circledR \left(A^2_{l_1} \cdots A^2_{l_i}\right) \tag{20}$$

is refining; j is the number of digits of x that must be absorbed before emission of $A^2_{l_i}$ becomes possible.

With $I_j = A^2_{k_1} \cdots A^2_{k_j}[-1, 1]$ and $J_i = A^2_{l_1} \cdots A^2_{l_i}[-1, 1]$, condition (20) is equivalent to $T(I_j, J_i) \subseteq J_i$. By (19), this may be replaced by $f(I_j) \subseteq J_i$. Finally, since $I_m \subseteq I_j$ for all $j \leq m$ and $J_n \subseteq J_i$ for all $i \leq n$, the criterion $\forall i \leq n\ \exists j \leq m\colon\ f(I_j) \subseteq J_i$ is equivalent to $f(I_m) \subseteq J_n$.

This result shows that in some sense, the realisation of f by the loop $y = T(x, y)$ is optimal: once x is known to be in some interval I_m, the loop emits exactly the amount of digit matrices that corresponds to the information that $f(x)$ is in $f(I_m)$.

The result can be used to derive quantitative information about the number of emitted digits by the following general argument. Let ℓ be the length of $f(I_m)$. Since the length of J_n is 2^{1-n}, we get $\ell \leq 2^{1-n}$ in case of $f(I_m) \subseteq J_n$, which implies that at most $1 - \lceil \log_2 \ell \rceil$ digits can be emitted. On the other hand, if $\ell \leq 2^{-n}$, then at least n digits can be emitted, which means that at least $-\lceil \log_2 \ell \rceil$ can be emitted. This shows that the number of emitted digits can be predicted up to an uncertainty of 1, once the length of $f(I_m)$ is known.

Since the length of I_m is 2^{1-m}, the length ℓ of $f(I_m)$ can be estimated by the mean value theorem:

$$\ell \leq 2^{1-m} \sup_{x \in I_m} f'(x) \quad \text{and} \quad \ell \geq 2^{1-m} \inf_{x \in I_m} f'(x)\ .$$

The derivative f' is symmetric ($f'(-x) = f'(x)$), decreasing for $x < 0$, increasing for $x > 0$, and has global minimum $f'(0) = \frac{1}{2}$. While f' tends to ∞ as x goes to 1, we still have $f'(\frac{3}{4}) < 1$. Hence, we have $2^{-m} \leq \ell < 2^{1-m}$ for $I_m \subseteq [-\frac{3}{4}, \frac{3}{4}]$ which means that $m - 1$, m, or $m + 1$ digits may be emitted. Putting this the other way round, $n - 1$, n, or $n + 1$ digits have to be absorbed in order to emit n digits. The condition $I_m \subseteq [-\frac{3}{4}, \frac{3}{4}]$ for this result is satisfied for arguments starting with the digits A^2_0, or $A^2_1 A^2_{-1}$, or $A^2_1 A^2_0$, or $A^2_1 A^2_1 A^2_{-1}$, or the symmetric sequences starting with A^2_{-1}.

For intervals I_m not contained in $[-\frac{3}{4}, \frac{3}{4}]$, one has to do some more work. The m digits $A^2_{k_1}, \ldots, A^2_{k_m}$ multiply to a matrix $\left(\begin{smallmatrix} 1 & K \\ 0 & 2^m \end{smallmatrix}\right)$ where $K = \sum_{j=1}^{m} k_j \cdot$

2^{m-j}. Hence, I_m is the interval $[\frac{K-1}{2^m}, \frac{K+1}{2^m}]$. The possible values of K cover all integers with $|K| \leq 2^m - 1$. By symmetry, we restrict to the case $K \geq 0$. In this case, the supremum $\sup_{x \in I_m} f'(x)$ is $f'(\frac{K+1}{2^m})$. For $K = 2^m - 1$, this is infinite; we treat this as a special case below. For $K \leq 2^m - 2$, we have $f'(\frac{K+1}{2^m}) \leq f'(\frac{2^m-1}{2^m})$, so that we get

$$\begin{aligned}
\ell &\leq 2^{1-m} \cdot f'\left(\tfrac{2^m-1}{2^m}\right) \\
&= \frac{2}{2^m(1+\sqrt{1-(2^m-1)^2/2^{2m}})\sqrt{1-(2^m-1)^2/2^{2m}}} \\
&\leq \frac{2}{2^m\sqrt{(2^{m+1}-1)/2^{2m}}} \\
&= \frac{2}{\sqrt{2^{m+1}-1}} \;\leq\; 2^{1-m/2}
\end{aligned}$$

In the special case $K = 2^m - 1$, the length is computed explicitly without using the derivative:

$$\begin{aligned}
\ell &= f\left(\tfrac{2^m}{2^m}\right) - f\left(\tfrac{2^m-2}{2^m}\right) \\
&= 1 - \frac{1-\sqrt{1-(2^m-2)^2/2^{2m}}}{(2^m-2)/2^m} \\
&= 1 - \frac{2^m - \sqrt{2^{2m}-(2^m-2)^2}}{2^m-2} \\
&= \frac{-2+\sqrt{4\cdot 2^m - 4}}{2^m-2} \;=\; 2\frac{\sqrt{2^m-1}-1}{2^m-2} \\
&= 2\frac{w-1}{w^2-1} \text{ where } w = \sqrt{2^m-1} \\
&= \frac{2}{w+1} \;=\; \frac{2}{\sqrt{2^m-1}+1} \;\leq\; 2^{1-m/2}
\end{aligned}$$

In both cases, we have $\ell \leq 2^{1-m/2}$ which means that at least $m/2-1$ digits can be emitted. Put the other way round, this means that at most $2n+1$ digits must be absorbed to emit n digits. Looking at the last chain of (in)equations, one sees that this bound is sharp for $x = 1$.

Summarising, we see that the evaluation of $f(x)$ is possible for all x in $[-1, 1]$, even for the two end points. For $|x| \leq \frac{3}{4}$, about $n+1$ absorptions are needed to produce n digits, while for $|x| \approx 1$, about $2n+1$ absorptions are required. These results can be translated back to square root. With $S^1(\frac{3}{4}) = 7$ and $S^1(-\frac{3}{4}) = \frac{1}{7}$, one obtains:

Theorem 5.1 *The function $\sqrt{x}$ is computable for all x in $[0, \infty]$, including the two end points. To produce n digits of $\sqrt{x}$, about $n+1$ digits of x are needed if x is between $1/7$ and 7, and about $2n+1$ digits of x if x is close to 0 or ∞ (this is the worst case).*

References

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

[2] A. Edalat and P. Potts. A new representation for exact real numbers. In S. Brookes and M. Mislove, editors, *MFPS '97*, volume 6 of *Electronic Notes in Theoretical Computer Science*, 1997. URL: `http://www.elsevier.nl/locate/entcs/volume6.html`.

[3] W. Gosper. Continued fraction arithmetic. Technical Report HAKMEM Item 101B, MIT Artificial Intelligence Memo 239, MIT, 1972.

[4] R. Heckmann. The appearance of big integers in exact real arithmetic based on linear fractional transformations. In *Proc. Foundations of Software Science and Computation Structures (FoSSaCS '98)*, volume 1378 of *LNCS*, pages 172–188. Springer-Verlag, 1998.

[5] R. Heckmann. Big integers and complexity issues in exact real arithmetic. In *Third Comprox workshop (Sept. 1997 in Birmingham)*, volume 13 of *Electronic Notes in Theoretical Computer Science*, 1998. URL: `http://www.elsevier.nl/locate/entcs/volume13.html`.

[6] R. Heckmann. Contractivity of linear fractional transformations. In J.-M. Chesneaux, F. Jézéquel, J.-L. Lamotte, and J. Vignes, editors, *Third Real Numbers and Computers Conference (RNC3)*, pages 45–59, April 1998.

[7] A. Nielsen and P. Kornerup. MSB-first digit serial arithmetic. *J. of Univ. Comp. Scien.*, 1(7):523–543, 1995.

[8] P. J. Potts and A. Edalat. Exact real arithmetic based on linear fractional transformations. Draft, Imperial College, available from `http://www-tfm.doc.ic.ac.uk/~pjp`, December 1996.

[9] P. J. Potts and A. Edalat. Exact real computer arithmetic. Draft, Imperial College, available from `http://www-tfm.doc.ic.ac.uk/~pjp`, March 1997.

[10] P. J. Potts. Computable real arithmetic using linear fractional transformations. Draft PhD Thesis, Imperial College, available from `http://www-tfm.doc.ic.ac.uk/~pjp`, June 1996.

[11] P. J. Potts. Efficient on-line computation of real functions using exact floating point. Available from `http://www-tfm.doc.ic.ac.uk/~pjp`, May 1998.

[12] P. Potts, A. Edalat, and M. Escardó. Semantics of exact real arithmetic. In *Proc. Twelfth Annual IEEE Symposium on Logic in Computer Science*, pages 248–257. IEEE, 1997.

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