



Available online at www.sciencedirect.com

SCIENCE @ DIRECT®

The Journal of Logic and
Algebraic Programming 66 (2006) 50–67

THE JOURNAL OF
LOGIC AND
ALGEBRAIC
PROGRAMMING

www.elsevier.com/locate/jlap

Exact real arithmetic using centred intervals and bounded error terms

J. Blanck

Department of Computer Science, University of Wales Swansea, Singleton Park, Swansea, SA2 8PP, UK

Accepted 20 July 2005

Abstract

Approximations based on dyadic centred intervals are investigated as a means for implementing exact real arithmetic. It is shown that the field operations can be implemented on these approximations with optimal or near optimal results. Bounds for the loss in quality of approximations for each of the field operations are also given. These approximations can be used as a more efficient alternative to endpoint based implementations of interval analysis.

© 2005 Elsevier Inc. All rights reserved.

1. Introduction

We will consider implementing exact computations on the real numbers.

Definition 1.1. A partial function $f: \mathbb{R}^n \rightarrow \mathbb{R}$ is (*exactly*) *computable* if there exists a computable function $\hat{f}: \mathbb{N}^2 \rightarrow \mathbb{N}$, such that for all $\mathbf{x} \in \text{dom } f$ and any representation $\hat{\mathbf{x}} \in \mathbb{N}$ of $\mathbf{x} \in \mathbb{R}$, $\hat{f}(\hat{\mathbf{x}}, k)$ converges to a representation of a real number approximation a satisfying

$$|a - f(\mathbf{x})| < 2^{-k}.$$

We now define the notion of exact real arithmetic.

Definition 1.2. An *exact real arithmetic* is an algebra of exactly computable operations defined on some subset of the real numbers.

Such an algebra should normally include at least the field operations addition, multiplication, negation and inverse, and will usually contain other operations, such as square roots, exponentials, logarithms, and trigonometric functions.

E-mail address: J.E.Blanck@swan.ac.uk

1567-8326/\$ - see front matter © 2005 Elsevier Inc. All rights reserved.
doi:10.1016/j.jlap.2005.07.002

Thus, in practise, an implementation of an exact real arithmetic provides a representation of the abstract data type of a field of real numbers.

We note that floating point arithmetic is *not* an exact real arithmetic. This is due to the limited amount of precision available in floating point numbers (causing round-off errors) and to the set of floating point numbers not being a subfield of the reals (causing over-flow).

There exist several implementations of exact real arithmetic. They are generally based on nested or intersecting rational intervals. For a survey of existing implementations see [12], but see also [8,17] and the MPFR web site.

In this paper we consider implementations using *centred dyadic intervals*. These are rational intervals represented by a pair of dyadic numbers giving the centre point and the radius of the interval. The next section gives some background to exact computation and interval representations. Section 3 discusses the centred intervals used in our implementation. The structure of these approximations of real numbers is compared to domain theory. Finally, Section 4 gives our optimal implementation of addition and multiplication and a near optimal implementation of inverse for an exact real arithmetic.

2. Background and preliminaries

2.1. The underlying theory

The theoretical foundation for exact real computation, and for this approach to an exact real data type, is the subject Computable Analysis [19,25,1]. There are several approaches when choosing concrete representations of real numbers. The most compelling reason for Computable Analysis is that all operations will be implementable on an ordinary computer (given that it has enough memory). This is in contrast to some abstract approaches based on programming with algebras, for example, Blum–Shub–Smale’s abstract model of computations on the reals [7]. The most notable difference between Computable Analysis and the BSS model is that equality, =, and the ordering, <, are not computable operations in Computable Analysis but are primitive in the BSS model. However, the abstract model of computation given by Tucker and Zucker [22,23], does truly model the behaviour of concrete computations of discontinuous functions such as equality.

2.2. Algebra of exact real arithmetic

Recall from Definition 1.2 that exact real arithmetic is an algebra of exact operations over a subset A of the real numbers. To be more specific, the algebra should be of the form

$$(A, +, -, \cdot, ^{-1}, \dots, 0, 1, \dots)$$

where $A \subseteq \mathbb{R}$ and the operations have the traditional type from the classical real numbers, e.g., $+: A^2 \rightarrow A$. The algebra will likely contain other operations and constants such as \sin , \log , π , e , and rational numbers.

When implementing an algebra of exact real arithmetic it is necessary to represent the real numbers in the computer. This means that we are actually computing using a representing algebra C that models the implementation method.

The representation that we have chosen for real numbers is intersecting rational intervals with rapidly diminishing radii. The rational intervals are in turn represented by pairs of

rational numbers. The centre-points of these intervals is a Cauchy sequence converging to the real number. Thus, real numbers are sequences of pairs of rational numbers, or functions from \mathbb{N} to \mathbb{Q}^2 .

Thus, the elements of the representing algebra C are functions, $C \subseteq (\mathbb{N} \rightarrow \mathbb{Q}^2)$. In the algebra C there are representations of the operations in A . For example, there will be an operation $+$ on C with type $+: C^2 \rightarrow C$ that represents the operation $+$ in A of type $+: A^2 \rightarrow A$. The algebra

$$C = (C \subseteq (\mathbb{N} \rightarrow \mathbb{Q}^2), +, -, \cdot, ^{-1}, \dots, \hat{0}, \hat{1}, \dots)$$

has the signature of A as a subsignature. The notion of representation is modelled by a homomorphism

$$\varphi: C \rightarrow A.$$

In addition, the representing algebra C is by necessity multi-sorted and contains sorts for natural and rational numbers, and of course, operations on these sorts.

Input and output of elements of C is of course difficult, since the elements are functions. There must therefore exist conversions to and from real numbers. We assume that there exists an operation $\text{const}: \mathbb{Q} \rightarrow C$ that for any rational constant gives a real number (a function) with the same value. To access a real number we assume that there is an operation

$$\text{approx}: C \times \mathbb{N} \rightarrow \mathbb{Q}^2.$$

The operation approx extracts the n th interval in the sequence, which we assume to have a radius less than or equal to 2^{-n} .

For example, to compute the real number π^2 we might access a (hopefully) provided constant $\hat{\pi} \in C$, and compute the real $\hat{\pi} \cdot \hat{\pi}$ in the algebra C . To compute an interval approximation of π^2 with radius 2^{-n} or less we compute

$$\text{approx}(\hat{\pi} \cdot \hat{\pi}, n).$$

For the elements of the algebraic theory of data types see Meinke and Tucker [15].

2.3. Related concepts

Exact real arithmetic as presented here is similar to *Interval Arithmetic* (see [16]). One common interpretation of interval arithmetic is that it is a forward computation from a set of starting observations to compute an interval guaranteed to contain the forward image of these intervals. This is in contrast to exact real arithmetic where the starting observations must be real numbers, not interval approximations of real numbers.

Definition 2.1. An operation $f: \mathbb{R} \rightarrow \mathbb{R}$ is *implemented* by an interval mapping \hat{f} if, for every x in the interval \mathbf{x} , $f(x)$ belongs to the interval $\hat{f}(\mathbf{x})$.

There is no promise made that the resulting interval will be of a certain size.

Suppose \mathbf{x} is the interval $[3, 4]$ then the best possible answer we can get from a doubling operation $f(x) = 2x$ in interval arithmetic is the interval $[6, 8]$. If this resulting interval is

too wide, and if a better starting approximation \mathbf{x}' can be computed, say [3.14, 3.15], then the computation may be restarted to compute a better interval for the result, in this case [6.28, 6.30]. Repeating this until a sufficiently narrow interval is found is, in fact, one way of achieving exact arithmetic.

So, the basic difference between exact real arithmetic and interval arithmetic is that in order to compute a result with the required precision, it is assumed in exact real arithmetic that any values can be arbitrarily well approximated. We will look at the implementation of exact arithmetic under this assumption.

On the other hand, exact real arithmetic can replace floating point operations in implementing interval computations. Whether this is reasonable thing to do is not considered here.

Symbolic computations can also be used to achieve exact arithmetic for some expressions. For example, the expression $\sqrt{2} \sin \frac{\pi}{4}$ can be reduced to 1 without doing any numerical computations at all. We do not consider incorporating symbolic computations here but rely solely on doing numerical computations.

3. Centred intervals as approximations

In general, approximations are based on the existence of a countable dense set within the space. We will consider here mainly the case of approximations of real numbers as it is the most basic case.

3.1. Approximations of real numbers

Fundamentally, the rationals are dense in the reals. If one takes the rationals as the basis for approximations there are two equivalent ways of constructing approximations. To approximate a real point x either take a (closed) interval $[a, b]$ with rational endpoints that contains x , or take a (closed) *centred interval*, that is a pair of rational numbers (c, e) such that

$$|x - c| \leq e.$$

The two approaches are obviously equivalent. An approximation of the form $[a, b]$ can also be given as $((a + b)/2, (b - a)/2)$, and conversely, (c, e) can be given as $[c - e, c + e]$.

We have chosen a centred representation rather than an end-point based representation of the rational intervals. The rationale for this is that we will further on impose a bound on the term e , thereby achieving a more compact representation, essentially halving the storage space as a, b and c are of similar size. This more compact representation has several advantages:

1. Less storage requirements.
2. Improved efficiency.
3. Correlation between the information content and size of representation.

A disadvantage of a centred representation compared with an end-point based representation of intervals is that working out the centre point of an image interval is not the same as taking the image of the centre point, thus working through the numerical details is a greater undertaking.

3.2. Centred dyadic intervals and approximations

Using rational numbers (with arbitrary denominators) to approximate real numbers and performing computations on these approximations entails a need to represent exactly rational numbers with huge denominators. For example, the simple operation of squaring a rational approximation p/q results in a number p^2/q^2 that requires double the memory to store.

By approximating these rationals in turn it is possible to keep the size of the approximations down. This is also done in interval arithmetic and is there called *rounded interval arithmetic* because rational numbers are rounded to floating point numbers. The specific subset of the reals to be rounded to remains to be chosen. For example, within every interval there exists a *simplest rational*. The simplest rational is the one with a minimal positive denominator and among these the one with the smallest absolute value of the numerator. Another option is to use *dyadic numbers*, numbers of the form $m \cdot 2^{-s}$, where m and s are integers. It is generally more efficient to round to a dyadic number than to round to the simplest rational in some interval.

We use the above as motivation to use centred dyadic intervals as approximations.

Definition 3.1. A *centred dyadic interval* is represented by a triple (m, e, s) of the form

$$a = (m \pm e)2^{-s},$$

where the *mantissa* m and the *exponent* s are integers, and the *error term* e is a natural number. A real x is *approximated* by a if

$$|x - m2^{-s}| \leq e2^{-s},$$

or equivalently,

$$x \in [(m - e)2^{-s}, (m + e)2^{-s}].$$

Fix j . A *centred dyadic j -approximation* is a centred dyadic interval where the error term is strictly bounded by 2^j .

We will often assume that j is some fixed number and will simply write *centred dyadic approximation*.

Having approximations with bounded error terms ensures that there is a relationship between the size and the information content of an approximation.

Centred dyadic approximations of this form were also used by van der Hoeven [13] and a similar choice is made by Müller [18]. The error term e may be fixed to be 1, as is done by Lester [12], but by generalising the error term to take on other values, bits are less often thrown away because of rounding. The generalised error terms are related to the notion of guard bits in floating point computations. Another advantage of generalising the error term is that we may use an error term of 0 to denote an exact dyadic number.

We will use a and b to denote approximations, m and n to denote mantissas (or significands), s and t to denote exponents, and e to denote error terms.

These centred dyadic approximations are just floating point numbers with error terms. Recall that a floating point number is of the form

$$(-1)^s \cdot m \cdot 2^s,$$

where S is the sign bit, m is the mantissa, and s is the exponent. The mantissa in double precision floating point numbers (translating the ANSI/IEEE Standard 754/854 to our setting) is between 2^{52} and $2^{53} - 1$ (which fits into 52 bits since the leftmost bit, always 1, need not be stored). The exponent ranges from -1074 to 971 (there are also some special representations of entities such as zero, non-normalised numbers, and infinity). Thus, floating point numbers are special cases of the approximations above where both the mantissa and the exponent are bounded and the error term is missing. The inclusion of the error terms in the approximations is what will give us the ability to claim that we are doing exact real arithmetic.

3.3. Quality of approximations

We introduce two measures of quality or accuracy of approximations. The different notions behave well with regard to different sets of operations.

Definition 3.2. A centred dyadic interval $a = (m \pm e)2^{-s}$ has *precision*

$$s - (\lfloor \log_2 e \rfloor + 1),$$

and *significance*

$$\lfloor \log_2 |m| \rfloor - \lceil \log_2 e \rceil,$$

whenever these expressions are defined.

For example, the interval $(73 \pm 6)^{-8}$ has precision

$$8 - (\lfloor \log_2 6 \rfloor + 1) = 8 - (2 + 1) = 5,$$

and significance

$$\lfloor \log_2 73 \rfloor - \lceil \log_2 6 \rceil = 6 - 3 = 3.$$

That is, precision measures the number of correct bits in the fractional part, whereas significance measures the number of correct bits regardless of the magnitude. Recall from numerical analysis that precision works better with additive operations, and that significance works better with multiplicative operations.

It would seem that a more natural notion for precision is arrived at by rounding the logarithm upwards and not adding one, but the version presented here is easier to work with for our purposes.

3.4. Rounding approximations

To avoid an increase in size of the dyadic approximations during the course of a computation, rounding of intermediate results need to be done rather often. We will assume that rounding takes place after each basic operation of the algebra. For example, let x be included in the centred dyadic interval $a = (m \pm e)2^{-s}$, where $m \geq e \geq 0$. The centred dyadic interval $b = (m^2 + e^2 \pm 2me)2^{-2s}$ contains x^2 , but b takes double the storage space of that of a . But rounding b will give a centred dyadic approximation that loses very little significance and avoids increasing the size of the approximation, see Proposition 4.11.

3.5. Other approaches

Centred dyadic approximations are not necessarily the best choice for all applications but they do have two important properties that many other choices do not possess in conjunction. Firstly, efficient algorithms have been developed to compute all common operations on the binary representation of dyadic numbers, see, for example, [9]. Multiplication can be computed in $O(n \log n \log \log n)$, where n is the size of the operands, by, e.g., the Schönhage–Strassen method [14,20]. Secondly, for any bounded interval the storage space needed for an approximation with precision p of any point in the interval is merely $p + \log p + c$, where c is some constant. For unbounded intervals this is clearly not true since the representation of the integer part becomes unbounded. However, for a fixed significance the same bound holds for all reals.

Approximations based on arbitrary rational numbers do not have any bound on the amount of storage needed for approximations with a fixed precision or significance. The process of finding the simplest rational for any point will give bounds on the storage but will also incur a lot of computation.

Various signed digit representations of the reals can also be considered. These representations share the limited space requirement of dyadic representation. However, so far, there seems to be no complete implementation of all fundamental operations (including transcendental functions) using directly the signed digit representation. Thus, though an elegant representation, it should not outperform the dyadic representations, at least as long as the hardware does not directly support signed bits. Yet other alternatives are approximations that use continued fractions, either purely as [24], or as linear fractional transformation [11,10]. While both methods may require even less storage than dyadic representations, there still do not exist methods for many elementary operations that are as efficient as those for dyadic approximations.

3.6. Other spaces

Approximations for other metric spaces can easily be constructed. Choosing among different kinds of approximations for general metric spaces will be even more difficult than choosing approximations for real computations. In the real case, some subset of rational intervals is the obvious way to approximate reals, so we essentially had to consider how to represent the chosen dense set of rationals. For arbitrary metric spaces one has to decide on the dense set to use and sometimes even on which metric to use. Consider, for example, finding approximations of the real continuous functions on a compact interval. Approximations could be closed spheres (according to some norm) with centres in some dense set, e.g., polynomials or piecewise linear functions with rational coefficients or parameters. Selecting the best approximations will probably depend on what operations are to be performed. Hence, an extensive study is needed in order to decide on a set of approximations for general computations.

3.7. Computer representations of the approximations

To compute with the dyadic approximations considered here they need to be mapped to existing data types. The error term would normally be bounded so this may be represented by the native integer data type of the language/processor, i.e., in C the data type `int` can be

used. The exponent can be arbitrarily large in principle, however, there is no point in being able to represent exponents that would require the mantissas to be larger than the available memory. A 64 bit exponent is therefore enough for most practical implementations today. The mantissas are arbitrarily large integers, and must be represented with some form of big integer package unless a data type of this sort is included in the language.

An alternative that merits mention is the choice made by Müller [18] in his implementation. There, the mantissa together with the exponent is represented by an arbitrary precision floating point number. The error term then needs to be handled separately. We have chosen the form here with explicit exponents since we have some theoretical points to explain, there might also be some minor gains to be made by having the error term as well-defined as it is in our approach. On the other hand, the floating point approach can take advantage of already implemented transcendental functions and so on.

3.8. Structure of approximations

Interval approximations of real numbers can be ordered by how narrow they are. We let \perp be the trivial approximation that approximates any real number, that is it represents the whole real line.

Definition 3.3. Let a and b be rational intervals. Define an ordering \sqsubseteq by

$$a \sqsubseteq b, \quad \text{if } b \subseteq a.$$

That is, b is a *better* approximation than a if b approximated a subset of the real numbers approximated by a .

A *cusl* is a partial order with a least element that have suprema for each finite consistent subset.

Lemma 3.4. *The rational intervals form a cusl under \sqsubseteq .*

Proof. The least element is \perp . The supremum of a consistent set of intervals is their intersection. \square

The ideal completion of this cusl is the *interval domain*. Refer to [21,2] for more on the interval domain and the computability properties induced on the reals by this structure.

Restricting from rational intervals to centred dyadic intervals gives a substructure.

Lemma 3.5. *The centred dyadic intervals form a subcusl of the rational intervals.*

Proof. We need only verify that the supremum of two consistent centred dyadic intervals is again a centred dyadic interval. The supremum is the intersection of the two intervals. The left endpoint of the intersection is either of the dyadic left endpoints of the two intervals, hence the left endpoint is dyadic. Similarly for the right endpoint. The midpoint between two dyadic points are dyadic. Half the distance between dyadic points is also dyadic so the intersection is a centred dyadic interval. \square

However, restricting to centred dyadic approximations entails a loss of the supremum property.

Example 3.6. The least upper bound of two approximations is the intersection of the intervals approximated, which again is a dyadic interval. However, in the case of a bounded error term, this dyadic interval might not be representable with an error term within the bound. For example, consider centred dyadic j -approximations for $j > 1$. The least upper bound of $(0 \pm (2^j - 1))2^0$ and $(1 \pm (2^j - 1))2^0$ is the interval represented by $(1 \pm (2^{j+1} - 1))2^{-1}$, which clearly cannot be represented with an error term within 2^j . Hence, the approximations do not form a csl. There is, however, a complete set of minimal upper bounds, $(0 \pm (2^j - 2))2^0$ and $(1 \pm (2^j - 2))2^0$.

The last observation in the example hints at the possibility that centred dyadic approximations form an *SFP-domain*.

Lemma 3.7. *The ideal completion of the centred dyadic j -approximations ordered by \sqsubseteq is an SFP-domain.*

Proof. The supremum of a consistent finite set of centred dyadic j -approximations is the intersection, which again is a dyadic interval. Any dyadic interval is centred as its midpoint is dyadic. Hence, it is sufficient to show that any centred dyadic interval has a complete finite set of minimal upper bounds among the j -approximations.

Let $a = (m \pm e)2^{-s}$ be a centred dyadic interval. If $e < 2^j$ then a is itself a j -approximation and we are finished. Assume, therefore, that $e \geq 2^j$. Let

$$S = \{(n \pm f)2^{-t} : a \sqsubseteq (n \pm f)2^{-t}, f < 2^j, t \leq s\}.$$

The set S is finite and partially ordered. Let S' be the subset of S that contain all approximations of the form

$$(m + k \pm (2^j - 1))2^{-s},$$

where

$$|k| \leq e - 2^j + 1.$$

Note that the approximations in S' cover the centred dyadic interval a , and that adjacent approximations overlap by more than half their lengths. Any approximation of the form $b = (n \pm f)2^{-t}$ above a , where $t > s$ is above some of the approximations in S' as b has at most half the length of the approximations in S' . Thus, the set S of upper bounds is complete. The minimal elements of S form a complete finite set of minimal upper bounds completing the proof. \square

Thus, the computability on the set of computable reals can be modelled as in [2,3].

When considering the representations of centred dyadic approximations above, the ordering \sqsubseteq is only a pre-order as anti-symmetry fails, for example, $(m \pm e)2^{-s}$ approximates the same interval as $(2m \pm 2e)2^{-s-1}$. This minor problem may be addressed by dividing out common powers of 2 from m and e , and adjusting the exponent, thereby getting unique representations of each centred dyadic approximation.

4. Constructing approximations

In order to perform exact computations efficiently great care must be taken in choosing the strategy to be used in refining the computation, this is discussed in [5], but see also [13]. However, in [5], we can also notice the importance of finding the tightest possible approximations for each step during the computation.

Definition 4.1. Let \mathbf{a} be a k -tuple of centred dyadic intervals of real numbers. An operation $f: \mathbb{R}^k \rightarrow \mathbb{R}$ has an *optimal* implementation if there exists a maximal (with respect to \sqsubseteq) centred dyadic j -approximation b that represents every point in

$$f(\mathbf{a}).$$

Note that we only require that the returned approximation is maximal. This is since in general there does not need to exist suprema among centred dyadic j -approximations, see Section 3.8.

Clearly, not all operations can have optimal implementations since the image interval is not always a dyadic interval.

Definition 4.2. Let \mathbf{a} be a k -tuple of centred dyadic intervals of real numbers. An operation $f: \mathbb{R}^k \rightarrow \mathbb{R}$ has a *near optimal* implementation if for any dyadic number $d > \text{diam } f(\mathbf{a})$, a centred dyadic interval b can be computed such that

$$f(\mathbf{a}) \subseteq b, \quad \text{and} \quad \text{diam } b \leq d,$$

where $f(\mathbf{a})$ is the forward image of the set of points approximated by \mathbf{a} , and diam gives the diameter of an interval.

Near optimality gives centred dyadic intervals, these may be *rounded* to centred dyadic approximations as we will see presently.

4.1. Rounding

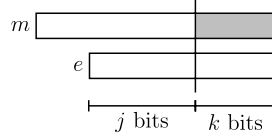
The operation of finding the best possible centred dyadic j -approximation from a centred dyadic interval will be referred to as rounding. It is an important operation since we have assumed that all intermediate results are rounded after each operation.

Proposition 4.3 (Rounding). *Given a centred dyadic interval $(m \pm e)2^{-s}$ there exists an optimal centred dyadic j -approximation.*

Proof. We need to give an approximation of the form $(n \pm e')2^{-t}$ where e' is to be strictly bounded by 2^j , i.e., fit within j bits. Fig. 1 indicates that what needs to be done is to cut both m and e off so that the new error term, adjusted for the rounding error introduced (the grey area), is small enough. Let q be the number of bits in the representation of e . A cut of at least $k = q - j$ bits is required. Let

$$t = s - k,$$

$$n = \text{round} \left(\frac{m}{2^k} \right),$$

Fig. 1. Rounding an approximation $(m \pm e)2^{-s}$.

and r be the introduced rounding error

$$r = |m - n2^k|.$$

The rounding error is at most 2^{k-1} . Create e' by rounding upwards the total error term $e + r$ divided by 2^k , i.e.,

$$e' = \left\lceil \frac{e + r}{2^k} \right\rceil.$$

From the construction it is clear that $(m \pm e)2^{-s} \subseteq (n \pm e')2^{-t}$.

The representation of e' may have more than j bits. If this is the case the process must be repeated with the original values m and e , and with k increased by 1. We must check that k need not be increased indefinitely, in fact, at most two increments will be necessary, and this only occurs when $j = 1$.

Consider first when k has been incremented once, then $k = q - j + 1$. We have $r \leq 2^{k-1}$ and $e < 2^q = 2^{j+k-1}$, and hence

$$\begin{aligned} e' &= \left\lceil \frac{e + r}{2^k} \right\rceil \\ &\leq \left\lceil \frac{2^q + 2^{k-1}}{2^k} \right\rceil \\ &\leq \left\lceil \frac{2^{j+k-1}}{2^k} \right\rceil + \left\lceil \frac{2^{k-1}}{2^k} \right\rceil \\ &\leq 2^{j-1} + 1, \end{aligned}$$

which is less than 2^j if $j > 1$. For $j = 1$ consider when k has been increased twice, that is, $k = q - j + 2 = q + 1$, then

$$\begin{aligned} e' &= \left\lceil \frac{e + r}{2^k} \right\rceil \\ &\leq \left\lceil \frac{2^q + 2^{k-1}}{2^k} \right\rceil \\ &\leq \left\lceil \frac{2^{k-1} + 2^{k-1}}{2^k} \right\rceil \\ &\leq 1 \\ &< 2^j. \end{aligned}$$

The optimality is achieved by computing e' as small as possible and by considering the smallest possible value of k first. \square

Note that the returned approximation is not unique since n may be rounded either up or down if $m/2^k$ has a fractional part that is exactly $\frac{1}{2}$.

Computing the actual rounding error r above is important if j is small, 1 or 2. For larger j (> 10), it is often better to approximate this rounding by the worst case of 2^{k-1} , since this very seldomly will affect the size of e' .

The following example illustrates when k has to be increased. In fact, k needs to be increased twice, but this only occurs when $j = 1$.

Example 4.4. Consider rounding the centred interval $(1280 \pm 257)2^{-10}$ when $j = 1$. We have $q = 9$ since the binary representation of 257 is 100000001. The first choice for k is therefore $k = 8$. This gives

$$\begin{aligned} k = 8, \quad 2^k = 256, \quad n = 5, \quad r = 0, \quad e' &= \left\lceil \frac{257+0}{256} \right\rceil = 2 \geq 2^1; \\ k = 9, \quad 2^k = 512, \quad n = 2, \quad r = 256, \quad e' &= \left\lceil \frac{257+256}{512} \right\rceil = 2 \geq 2^1; \\ k = 10, \quad 2^k = 1024, \quad n = 1, \quad r = 256, \quad e' &= \left\lceil \frac{257+256}{1024} \right\rceil = 1 < 2^1. \end{aligned}$$

Thus, the resulting approximation is $(1 \pm 1)2^0$. This is the optimal centred dyadic 1-approximation containing the given interval.

Proposition 4.5. *The precision lost by rounding a centred dyadic interval to a centred dyadic j -approximation is at most 2 if $j = 1$, and at most 1 if $j > 1$.*

Proof. Using the notation in the proof of Proposition 4.3 the precision of the result $(n \pm e')2^{-t}$ is

$$t - (\lceil \log_2 e' \rceil + 1) = s - k - j = s - q - i = s - (\lceil \log_2 e \rceil + 1) - i,$$

where $s - (\lceil \log_2 e \rceil + 1)$ is the significance of the original interval $(m \pm e)2^{-s}$ and i is the number of increments of k that has been performed. Which we know is bounded by 2 if $j = 1$, and by 1 if $j > 1$. \square

Proposition 4.6. *The significance lost by rounding a centred dyadic interval to a centred dyadic j -approximation, unless the result is a zero-centred approximation, is at most 2 if $j = 1$, and at most 1 if $j > 1$.*

Proof. Again, using the notation in the proof of Proposition 4.3 the significance of the result $(n \pm e')2^{-t}$, where $n \neq 0$, is

$$\begin{aligned} \lceil \log_2 n \rceil - \lceil \log_2 e' \rceil &= \lceil \log_2 m \rceil - k - (\lceil \log_2(e+r) \rceil - k) \\ &= \lceil \log_2 m \rceil - \lceil \log_2(e+r) \rceil. \end{aligned}$$

If $j > 1$ then $e \geq r$ and hence

$$\lceil \log_2(e+r) \rceil \leq \lceil \log_2 2e \rceil = \lceil \log_2 e \rceil + 1.$$

Thus, the loss in significance is at most 1.

If $j = 1$ then $r < 2e$ and therefore

$$\lceil \log_2(e+r) \rceil \leq \lceil \log_2 4e \rceil = \lceil \log_2 e \rceil + 2.$$

Thus, the loss in significance is at most 2. \square

Due to boundary effects the significance may actually increase with rounding. For example, the approximation $(7 \pm 6)2^0$ has significance -1 . Rounding this so that the error term is strictly bounded by 2^2 gives the approximation $(2 \pm 2)2^2$ which has significance 0. Clearly, the precision measure is not affected by this anomaly.

4.2. Field operations

We now turn to the field operations on centred dyadic approximations.

Proposition 4.7. *There exists an optimal implementation of addition of centred dyadic j -approximations.*

Proof. Let $a = (m \pm e)2^{-s}$ and $b = (n \pm e')2^{-t}$ be centred dyadic intervals and assume without loss of generality that $s \geq t$. The exact image interval of their sum is

$$a + b = (m + n2^{s-t} \pm (e + e'2^{s-t}))2^{-s},$$

An optimal centred dyadic j -approximation is obtained by rounding this centred dyadic interval using Proposition 4.3. \square

Proposition 4.8. *The loss of precision for the operation of addition on centred dyadic j -approximations, compared with the argument of least precision, is at most 2.*

Proof. Assume that $j > 1$. Using the notation in the proof, let p be the precision of the argument with least precision. Then

$$e < 2^{s-p}, \quad \text{and} \quad e' < 2^{t-p} \iff e'2^{s-t} < 2^{s-p}.$$

Thus

$$e + e'2^{s-t} < 2^{s-p+1} \implies \lfloor \log_2(e + e'2^{s-t}) \rfloor \leq s - p,$$

and hence the precision of the centred dyadic interval $(m + n2^{s-t} \pm (e + e'2^{s-t}))2^{-s}$ is

$$s - (\lfloor \log_2(e + e'2^{s-t}) \rfloor + 1) \geq s - (s - p + 1) = p - 1.$$

The result now follows from Proposition 4.5.

Consider the case $j = 1$, that is, when error terms are at most $1 = 2^0 = 2^{j-1}$. Now,

$$e \leq 2^{s-p-1}, \quad \text{and} \quad e' \leq 2^{t-p-1} \iff e'2^{s-t} \leq 2^{s-p-1},$$

and

$$e + e'2^{s-t} \leq 2^{s-p}.$$

If $s \neq t$, then the above inequality is actually strict and the result follows as before by Proposition 4.5.

Assume $s = t$, then the only non-trivial case is when $e + e' = 2$. If the sum $m + n$ is even then an approximation that loses only one in precision can be found by dividing

through by 2. If the sum $m + n$ is odd, then the rounding error r introduced in the proof of Proposition 4.3 is at most 1 when dividing by 4, so the possible error is at most 3, that is we can return an approximation with an error term of 4. This approximation will lose 2 in precision. \square

Addition of approximations with opposite signs and similar magnitude may result in unlimited loss of significance.

Proposition 4.9. *There exists an optimal implementation of negation on centred dyadic j -approximations.*

Proof. The negation of $a = (m \pm e)2^{-s}$ is $(-m \pm e)2^{-s}$, which already is a centred dyadic j -approximation. \square

Clearly, neither precision nor significance is affected by the operation of negation on approximations.

The product of two centred dyadic intervals $a = (m \pm e)2^{-s}$ and $b = (n \pm f)2^{-t}$ is

$$(m \pm e)2^{-s}(n \pm f)2^{-t} = (mn \pm mf \pm ne \pm ef)2^{-s-t},$$

which is always contained in the naïve centred dyadic interval

$$(mn \pm (|m|f + |n|e + ef))2^{-s-t}.$$

While correct this interval is overly conservative.

Proposition 4.10. *There exists an optimal implementation of multiplication of centred dyadic j -approximations.*

Proof. The exact image interval of the centred dyadic intervals $a = (m \pm e)2^{-s}$ and $b = (n \pm f)2^{-t}$ can be obtained by considering eleven cases similar to the cases in Moore [16, p. 12].

$$\left\{ \begin{array}{ll} (mn + ef \pm (|mf| + |ne|))2^{-s-t}, & \text{if } |m| \geq e, |n| \geq f, mn > 0; \\ (mn - ef \pm (|mf| + |ne|))2^{-s-t}, & \text{if } |m| \geq e, |n| \geq f, mn < 0; \\ (mn + mf \pm (|ne| + ef))2^{-s-t}, & \text{if } |m| < e, n \geq f; \\ (mn - mf \pm (|ne| + ef))2^{-s-t}, & \text{if } |m| < e, -n \geq f; \\ (mn + ne \pm (|mf| + ef))2^{-s-t}, & \text{if } m \geq e, |n| < f; \\ (mn - ne \pm (|mf| + ef))2^{-s-t}, & \text{if } -m \geq e, |n| < f; \\ (mn + |ne| \pm (|mf| + ef))2^{-s-t}, & \text{if } |m| < e, |n| < f, mn > 0, |mf| > |ne|; \\ (mn + |mf| \pm (|ne| + ef))2^{-s-t}, & \text{if } |m| < e, |n| < f, mn > 0, |mf| \leq |ne|; \\ (mn - |ne| \pm (|mf| + ef))2^{-s-t}, & \text{if } |m| < e, |n| < f, mn < 0, |mf| > |ne|; \\ (mn - |mf| \pm (|ne| + ef))2^{-s-t}, & \text{if } |m| < e, |n| < f, mn < 0, |mf| \leq |ne|; \\ (0 \pm (|mf| + |ne| + ef))2^{-s-t}, & \text{if } mn = 0. \end{array} \right.$$

Again, an optimal centred dyadic j -approximation is found using Proposition 4.3. \square

Note in the proof that only four multiplications are needed to compute the image interval, and only one of these is the product of two potentially large factors, the rest contain at least one bounded term. The four multiplications are also necessary for the naïve approximation above, hence the additional work is only a number of comparisons.

Proposition 4.11. *The loss of significance for the operation of multiplication on centred dyadic intervals with positive significance, compared with the argument of least significance, is at most 2.*

Proof. Since the significance is positive for both arguments, only the first two cases for multiplication need be considered. We prove the first case and leave the similar second case to the reader.

Assume, without loss of generality, that $|mf| \geq |ne|$, and let p be the significance of the second argument $(n \pm f)2^{-s}$, i.e.,

$$p = \lfloor \log_2 |n| \rfloor - \lceil \log_2 f \rceil.$$

The significance of the resulting interval is

$$\begin{aligned} & \lfloor \log_2(|mn| + ef) \rfloor - \lceil \log_2(|mf| + |ne|) \rceil \\ & \geq \lfloor \log_2 |mn| \rfloor - \lceil \log_2 2 \cdot \max(|mf|, |ne|) \rceil \\ & = \lfloor \log_2 |m| \rfloor + \lfloor \log_2 |n| \rfloor - \lceil \log_2 2|mf| \rceil \\ & = \lfloor \log_2 |m| \rfloor + \lfloor \log_2 |n| \rfloor - \lceil \log_2 |mf| \rceil - 1 \\ & = \lfloor \log_2 |m| \rfloor + \lfloor \log_2 |n| \rfloor - \lceil \log_2 |m| + \log_2 f \rceil - 1 \\ & \geq \lfloor \log_2 |m| \rfloor + \lfloor \log_2 |n| \rfloor - \lceil \log_2 |m| \rceil - \lceil \log_2 f \rceil - 1 \\ & = \lfloor \log_2 |m| \rfloor + \lfloor \log_2 |n| \rfloor - \lceil \log_2 |m| \rceil - (\lfloor \log_2 |n| \rfloor - p) - 1 \\ & \geq \lfloor \log_2 |m| \rfloor + \lfloor \log_2 |n| \rfloor - (\lfloor \log_2 |m| \rfloor + 1) - (\lfloor \log_2 |n| \rfloor - p) - 1 \\ & = p - 2. \end{aligned}$$

Thus the result. Note that $|mf| \geq |ne|$ can only occur if the significance of the first argument is at least $p - 1$. \square

The result above is sharp since $(5 \pm 1)2^0$ with significance 2 multiplied by itself gives $(5 \pm 1)2^0 \cdot (5 \pm 1)2^0 = (26 \pm 10)2^0$ which has significance 0.

Proposition 4.12. *The loss of significance for the operation of multiplication on centred dyadic j -approximations with positive significance, compared with the argument of least significance is at most 3, if $j > 1$, and at most 4, if $j = 1$.*

Proof. By Proposition 4.11 and Proposition 4.6. \square

Consider now the operation of inverse when applied to centred dyadic intervals. The inverse of an interval $a = (m \pm e)2^{-s}$ not containing 0 is the centred interval

$$\left(\frac{m}{m^2 - e^2} \pm \frac{e}{m^2 - e^2} \right) 2^s.$$

The significance of this centred interval differs by at most one from the significance of the argument. The fractions above are not dyadic in general. For example, let $a = (5 \pm 1)2^0$, then the inverse is the centred interval

$$b = \left(\frac{5}{24} \pm \frac{1}{24} \right),$$

which is the correct (non-dyadic) image interval $[\frac{1}{6}, \frac{1}{4}]$. A dyadic interval containing this interval must first be found. This can be done arbitrarily close, optimal dyadic intervals containing b for some given exponents are

$$(1 \pm 1)2^{-3}, (3 \pm 1)2^{-4}, (6 \pm 2)2^{-5}, (7 \pm 2)2^{-5}, (13 \pm 3)2^{-6}, (26 \pm 6)2^{-7},$$

$$(53 \pm 11)2^{-8}, (106 \pm 22)2^{-9}, (107 \pm 22)2^{-9}, (213 \pm 43)2^{-10}, \dots$$

Thus, there is no optimal centred dyadic interval to return. There does exist optimal centred dyadic approximations, but we cannot guarantee that rounding an optimal centred dyadic interval for some given exponent will give an optimal centred dyadic approximation. For example, rounding the interval $(107 \pm 22)2^{-9}$ listed above to a centred dyadic 4-approximation gives $(54 \pm 12)2^{-8}$, which is not optimal.

By computing a centred dyadic interval where the error term has slightly more than j bits and rounding to a centred dyadic j -approximation the risk of getting a non-optimal approximation is reduced.

A centred dyadic interval can be computed by

$$\left(\text{round} \left(\frac{m2^t}{m^2 - e^2} \right) \pm \left[\frac{1}{2} + \frac{e2^t}{m^2 - e^2} \right] \right) 2^{s-t},$$

for some appropriate choice of t . The $\frac{1}{2}$ is added to allow for the error introduced by rounding the centre point. Since the significance is approximately the same as that of the argument, t can be chosen to be $\lceil 2 \log_2 |m| \rceil - \lceil \log_2 e \rceil + j + k$, where k is a small constant number to allow for rounding of the resulting interval.

The centred dyadic interval computed here will usually have an error term that is too large, so it will have to be rounded again. This double rounding is necessary if the aim is to use the error term to its full capacity.

Proposition 4.13. *There exists a near optimal implementation of the inverse of centred dyadic approximations.*

Proof. By increasing the number t in the centred dyadic interval constructed above, we get a tighter fit around the image interval. Clearly, the construction can give an interval with a diameter bounded by any dyadic number strictly greater than the diameter of the image interval. The resulting interval is rounded to a centred dyadic j -approximation. \square

Proposition 4.14. *The loss of significance for the operation of inverse on centred dyadic j -approximations, is at most 3, if $j > 1$, and at most 4, if $j = 1$.*

Proof. As observed before Proposition 4.13 the significance of the possibly nondyadic centred interval loses at most one in significance. A centred dyadic interval can be found at a cost of at most 1 again. The result now follows from Proposition 4.6. \square

Other operations, like transcendental functions, can be handled similarly to inverses, i.e., find a dyadic approximation of the centre point and a valid error term and then round the resulting interval.

4.3. Average behaviour

The bounds in loss of precision and significance given here are worst case bounds. The average case is much better, typically the loss is often 0, sometimes 1, and seldomly worse. However, giving an estimate of the average loss is not easy since the distribution of error terms in a computation is most likely not uniform. Further problems are that the average loss may differ between different operation, and what operations are used on average. We leave this as an open area.

5. Conclusions

The centred dyadic approximations used here have the merits of limited storage requirement and efficient implementation of all operations. They seem to be the most viable choice, which is also supported by the results in [4]. The bounded general error terms may be important in implementing exact real arithmetic since they may reduce the size of the approximations used in the computation, this is supported by [5].

The benefit of working with centred intervals compared with using endpoint representations is clearly that only one costly operation has to be performed per operation rather than two. In some cases centred dyadic approximations will give non-sharp intervals, but we have shown that very good approximations are returned for all operations considered. Moreover, we have a simple tool, rounding, that limits the size of intermediate results.

Acknowledgments

I wish to thank J.V. Tucker and three anonymous referees for useful criticisms of earlier drafts of this paper.

References

- [1] O. Aberth, *Computable Calculus*, Academic Press, 2001.
- [2] J. Blanck, Domain representability of metric spaces, *Ann. Pure Appl. Logic* 83 (1997) 225–247.
- [3] J. Blanck, Effective domain representations of $\mathcal{H}(X)$, the space of compact subsets, *Theor. Comput. Sci.* 219 (1999) 19–48.
- [4] J. Blanck, Exact real arithmetic systems: Results of competition, in: Blanck et al. [6], pp. 390–394.
- [5] J. Blanck, Efficient exact computation of iterated maps, *J. Log. Algebr. Program.* 64 (2005) 41–59.
- [6] J. Blanck, V. Brattka, P. Hertling (Eds.), *Computability and Complexity in Analysis*, Lecture Notes in Computer Science, vol. 2064, Springer, 2001.
- [7] L. Blum, M. Shub, S. Smale, On a theory of computation and complexity over the real numbers: NP-completeness, recursive functions, and universal machines, *Bull. Amer. Math. Soc.* 21 (1989) 1–46.
- [8] H.-J. Boehm, The constructive reals as a java library, *J. Log. Algebr. Program.* 64 (2005) 3–11.
- [9] J.M. Borwein, P.B. Borwein, *Pi and the AGM: A Study in Analytic Number Theory and Computational Complexity*, John Wiley & Sons, 1998.
- [10] A. Edalat, R. Heckmann, Computing with real numbers, in: G. Barthe, P. Dybjer, L. Pinto, J. Saraiva (Eds.), *Applied Semantics—Advanced Lectures*, Lecture Notes in Computer Science, vol. 2395, Springer, 2002, pp. 193–267.
- [11] A. Edalat, P.J. Potts, A new representation for exact real numbers, *Electronical Notes in Theoretical Computer Science*, vol. 6, Mathematical Foundations of Programming Semantics Pittsburgh, PA, 1997, 14pp.
- [12] P. Gowland, D. Lester, A survey of exact computer arithmetic, In Blanck et al. [6], pp. 30–47.

- [13] GMPX, an extension of the GMP package that contains safe reals. Available from: <http://www.math.u-psud.fr/~vdhoeven/Progs/gmpx.tar.gz>.
- [14] D.E. Knuth, *Seminumerical Algorithms, The Art of Computer Programming*, vol. 2, Addison-Wesley, 1969.
- [15] K. Meinke, J.V. Tucker, Universal algebra, in: S. Abramsky et al. (Eds.), *Handbook of Logic in Computer Science*, vol. I, Oxford University Press, 1992, 189–368.
- [16] R.E. Moore, *Methods and applications of interval analysis*, SIAM Studies in Applied Mathematics, vol. 2, Society for Industrial and Applied Mathematics (SIAM), Philadelphia, PA, 1979.
- [17] V. Ménessier-Morain, Arbitrary precision real arithmetic: design and algorithms, *J. Log. Algebr. Program.* 64 (2005) 13–39.
- [18] N. Müller, The iRRAM: Exact arithmetic in C++, in: Blanck et al. [[6]], pp. 223–252.
- [19] M.B. Pour-El, J.I. Richards, *Computability in Analysis and Physics, Perspectives in Mathematical Logic*, Springer, Berlin, 1989.
- [20] A. Schönhage, V. Strassen, Schnelle Multiplikation großer Zahlen, *Computing* 7 (1971) 281–292.
- [21] V. Stoltenberg-Hansen, J.V. Tucker, Effective algebra, in: S. Abramsky et al. (Eds.), *Handbook of Logic in Computer Science*, vol. IV, Oxford University Press, 1995, pp. 357–526.
- [22] J.V. Tucker, J.I. Zucker, Computable functions and semicomputable sets on many sorted algebras, in: S. Abramsky et al. (Eds.), *Handbook of Logic in Computer Science*, vol. V, Oxford University Press, 2000, pp. 317–523.
- [23] J.V. Tucker, J.I. Zucker, Abstract versus concrete computation on metric partial algebras, *ACM Trans. Comput. Logic* 5 (4) (2004) 611–668.
- [24] J. Vuillemin, Exact real computer arithmetic with continued fractions, INRIA Report 760, INRIA, France, 1987.
- [25] K. Weihrauch, *An Introduction to Computable Analysis*, Springer, 2000.