Bound on Run of Zeros and Ones for Images of Floating-Point Numbers by Algebraic Functions

Tomas Lang, Jean-Michel Muller

INSTITUT NATIONAL DE RECHERCHE EN INFORMATIQUE ET EN AUTOMATIQUE

Bound on Run of Zeros and Ones for Images of Floating-Point Numbers by Algebraic Functions

Tomas Lang , Jean-Michel Muller

No 4045

Novembre 2000

THÈME 2

ISSN 0249-6399 ISRN INRIA/RR--4045--FR+ENG

Bound on Run of Zeros and Ones for Images of Floating-Point Numbers by Algebraic Functions

Tomas Lang* , Jean-Michel Muller†

Thème 2 — Génie logiciel
et calcul symbolique
Projet Arénaire

Rapport de recherche n°4045 — Novembre 2000 — 19 pages

Abstract: This paper presents upper bounds on the number of zeros and ones after the rounding bit for algebraic functions. These functions include reciprocal, division, square root, and inverse square root, which have been considered in previous work. We here propose simpler proofs for the previously given bounds given and generalize to all algebraic functions. We also determine cases for which the bound is achieved for square root. As is mentioned in the previous work, these bounds are useful for determining the precision required in the computation of approximations in order to be able to perform correct rounding.

Key-words: Algebraic functions, Computer arithmetic, Table Maker's Dilemma, Correct rounding, Floating-Point Arithmetic.

(Résumé : tsvp)

* Dept. of Electrical and Computer Engineering, University of California at Irvine
† CNRS, Projet CNRS/ENS Lyon/INRIA ARENAIRE, Ecole Normale Supérieure de Lyon

Unité de recherche INRIA Rhône-Alpes
655, avenue de l'Europe, 38330 MONTBONNOT ST MARTIN (France)
Téléphone : 04 76 61 52 00 - International: +33 4 76 61 52 00
Télécopie : 04 76 61 52 52 - International: +33 4 76 61 52 52

Majoration du nombre de 0 et de 1 consécutifs dans l'écriture binaire de l'image d'un flottant par une fonction algébrique

Résumé : Nous donnons des majorations du nombre de zéros ou de uns consécutifs après le bit d'arrondi pour l'image d'un nombre virgule flottante par une fonction algébrique. Des résultats similaires ont été publiés pour l'inverse, la division, la racine carrée et l'inverse de la racine carrée. Nous proposons des preuves plus simples pour ces fonctions et généralisons à l'ensemble des fonctions algébriques. Nous déterminons également des cas pour lesquels la borne est effectivement atteinte pour la racine carrée. Ces bornes permettent de connaître la précision avec laquelle il faut approcher ces fonctions pour en fournir un arrondi correct.

Mots-clé : Fonctions algébriques, arithmétique des ordinateurs, Dilemme du fabricant de tables, Arrondi correct, Virgule flottante.

1 Introduction

This paper presents upper bounds on the number of zeros and ones after the rounding bit for algebraic functions. These functions include reciprocal, division, square root, and inverse square root, which have been considered in previous work [1] [2] [5] [4]. We here propose simpler proofs for the bounds given in [1] and generalize to all algebraic functions. We also determine cases for which the bound is achieved for square root. As already discussed for inverse square root in [1], except for reciprocal, division, square root, and norms, the bounds obtained are not achieved for precisions up to double precision and it is not known if there is a precision for which they are achieved.

As is mentioned in the previous work, these bounds are useful for determining the precision required in the computation of approximations in order to be able to perform correct rounding.

1.1 Basic idea and example

Our approach for the bound on the maximum number of zeros is based on the following:

1. Develop a radix-2 digit-recurrence algorithm for the function[1]. Since we are going to use the algorithm to determine the bounds, to simplify this analysis we concentrate on algorithms of the restoring type.

2. Determine a bound on the minimum value of the (non zero) residual after a one in the rounding bit position is computed.

3. Determine the number of consecutive zeros that are produced by this minimum residual.

This approach can be used for any function for which a digit recurrence algorithm can be developed. We show that this is possible for the class of algebraic functions.

We now develop an example of the use of this procedure and then generalize to the class of algebraic functions. Consider the computation of the cubic root. That is, compute $y = x^{1/3}$ with $1/2 \leq x < 1$ and operand and result of n bits.

[1] Note that the implementation of the digit-recurrence algorithm results in a simple method for correct rounding; this is true even in the case in which the result is obtained in signed-digit representation, where the computation of the rounding includes the sign of the final residual. Consequently, with this implementation the bounds presented in this paper are not necessary; however, this implementation might be too complicated or slow, so that other algorithms based on approximations are preferable.

We define a radix-2 digit-recurrence algorithm that produces one bit of the result each iteration. That, is after iteration j the partial result is

$$y[j] = \sum_{i=1}^{j} y_i 2^{-i}$$

The residual after iteration j is

$$w[j] = 2^j(x - y[j]^3)$$

resulting in the recurrence

$$w[j+1] = 2w[j] - 3y[j]^2 y_{j+1} - 3y[j]y_{j+1}^2 2^{-(j+1)} - y_{j+1}^3 2^{-2(j+1)} \tag{1}$$

with $w[0] = x$. The selection function has to satisfy

1. $w[j+1] \geq 0$
2. $w[j+1]$ is minimum.

Consequently,

$$y_{j+1} = \begin{cases} 0 & \text{if } 2w[j] < 3y[j]^2 + 3y[j]2^{-(j+1)} + 2^{-2(j+1)} \\ 1 & \text{otherwise} \end{cases}$$

Since $y < 1$, and by construction $y[j] \leq y$, we obtain $y[j] < 1$. Therefore, for $j \geq 1$

$$3y[j]^2 + 3y[j]2^{-(j+1)} + 2^{-2(j+1)} \leq 61/16 < 4$$

Consequently, we get

$$y_{j+1} = 1 \text{ if } \quad (2w[j] \geq 4) \tag{2}$$

We now determine a nonzero lower bound on $w[n+1]$, when $y_{n+1} = 1$. For this we determine the number of fractional bits of $w[n+1]$ (width of $w[n+1]$). The width of $w[j+1]$ is the maximum width of the terms in the recurrence. Each of the terms has the following width:

1. Since $w[0] = x$ has n bits and the effect of this initial value is "shifted" one position left each iteration, the number of bits of the contribution of $w[0]$ is $n - j$.
2. The term $3y[j]^2$ has $2j$ fractional bits.

3. The term $3y[j]2^{-(j+1)}$ has $2j+1$ fractional bits.

4. The term $2^{-2(j+1)}$ has $2j+2$ fractional bits.

Consequently, the width of $w[j+1]$ is

$$widthw[j+1] = \max[(n-j),\ (2j+2)]$$

For $j = n$ this is $2n+2$. Consequently, a nonzero lower bound on $w[n+1]$ is $2^{-(2n+2)}$

So, how many consecutive 0s in y can occur after this minimum $w[n+1]$? Each 0 multiplies the residual by 2. From equation (2) we conclude that a 1 will certainly occur when the value of the residual becomes 4. Consequently[2],

$$maxrun \leq 2n+3$$

We generalize the idea described above to any algebraic function, as defined below. From this generalization, we obtain upper bounds on the maximum runs of zeros and of ones, after the rounding position. The class of algebraic functions includes the functions considered previously. In Section 4, we compare with the previously reported results.

1.2 Algebraic functions

We generalize the idea described above to any algebraic function. An *Algebraic Function* f is a function for which there exists a 2-variable polynomial P with integer coefficients such that

$$y = f(x) \Leftrightarrow P(x,y) = 0 \tag{3}$$

Examples are:

Division, reciprocation $y = a/x$, with $P(x,y) = a - xy$;

Square roots $y = \sqrt{x}$, with $P(x,y) = x - y^2$;

Roots $y = x^{1/p}$, with $P(x,y) = x - y^p$;

Square root reciprocal $y = 1/\sqrt{x}$, with $P(x,y) = 1 - xy^2$;

Norms $y = \sqrt{x^2+v^2}$, with $P(x,y) = x^2 + v^2 - y^2$;

Normalization $y = x/\sqrt{x^2+v^2}$, with $P(x,y) = x^2 - y^2(x^2+v^2)$;

[2] In Section 3 we analize whether this bound is achieved

Throughout the paper, we assume that we perform calculations in an n-bit binary number system. We assume that $1/2 \leq x < 1$ and $1/2 \leq y < 1$. The generalization to other ranges is discussed later. We also assume that, in the considered domain of computation the function $y \rightarrow P(x, y)$ is decreasing (Otherwise, if the considered domain is narrow enough so that this function is monotonic, it suffices to exchange P and $-P$.), and that

$$\frac{\partial P}{\partial y}(x, y) \neq 0 \tag{4}$$

To illustrate the development we use as running example the function $y = x^{3/5}$ with $1/2 \leq x < 1$. The corresponding polynomial is

$$P(x, y) = x^3 - y^5$$

1.3 Radix-2 restoring digit-recurrence algorithm for algebraic function

As indicated before, at step j of a digit-recurrence algorithm we have already computed

$$y[j] = 0.y_1 y_2 \ldots y_j = \sum_{i=1}^{j} y_i 2^{-i}$$

We define a *residual* $w[j]$ as

$$w[j] = 2^j P(x, y[j]) \tag{5}$$

Since the representation of y is non-redundant (restoring algorithm), the sequence $y[j]$ goes to y if and only if :

$$y_{j+1} = \begin{cases} 1 & \text{if} \quad y[j] + 2^{-(j+1)} \leq y \\ 0 & \text{otherwise} \end{cases} \tag{6}$$

Since the function $y \rightarrow P(x, y)$ is decreasing, (6) is equivalent to

$$y_{j+1} = \begin{cases} 1 & \text{if} \quad P(x,\ y[j] + 2^{-(j+1)}) \geq 0 \\ 0 & \text{otherwise} \end{cases} \tag{7}$$

In order to use (7), let us evaluate $2^{j+1}P(x,\ y[j]+2^{-(j+1)})$. We have

$$\begin{aligned} & 2^{j+1}P(x,\ y[j]+2^{-(j+1)}) \\ = \ & 2^{j+1}\left[P(x,\ y[j]) + \textstyle\sum_{i\geq 1}\frac{2^{-i(j+1)}}{i!}\frac{\partial^i P}{\partial y^i}(x,\ y[j])\right] \\ = \ & 2w[j] + \frac{\partial P}{\partial y}(x,\ y[j]) + \textstyle\sum_{i\geq 2}\frac{2^{(-i+1)(j+1)}}{i!}\frac{\partial^i P}{\partial y^i}(x,\ y[j]) \\ = \ & 2w[j] - C[j] \end{aligned}$$

Note that this is the definition of $C[j]$.

Consequently, the selection function and the next residual are computed as follows:

- If $2w[j] \geq C[j]$ then $y_{j+1} = 1$ and $w[j+1] = 2w[j] - C[j]$
- If $2w[j] < C[j]$ then $y_{j+1} = 0$ and $w[j+1] = 2w[j]$

For our example

$$C[j] = 5y[j]^4 + 10y[j]^3 2^{-(j+1)} + ...$$

2 Longest run of zeros and of ones for algebraic functions

2.1 Longest run of zeros starting from bit y_{n+2} .

As indicated before, we proceed as follows:

1. Determine a lower bound on the (non-zero) value of $w[n+1]$ when $y_{n+1} = 1$.
2. Use the selection function to determine an upper bound on the number of consecutive zeros.

Let us now determine the smallest possible nonzero value of $w[j]$. As said in the introduction, P is a 2-variable polynomial with integer coefficients. Define A_{pq} as the coefficient of $x^p y^q$ in P, so that

$$P(x,y) = \sum A_{pq} x^p y^q$$

and α_{pq} as the largest integer such that $|A_{pq}| \geq 2^{\alpha_{pq}}$ (with α_{pq} not defined for $A_{pq} = 0$).

Since x is an n-bit normalized fraction and $y[j]$ is a j-bit normalized fraction, the term $A_{pq}x^p y[j]^q$ is a multiple of $2^{\alpha_{pq}-np-jq}$. Hence, if we define ν_j as

$$\nu_j = \min_{p,q}\left(\alpha_{pq} - np - jq\right) \tag{8}$$

then $w[j] = 2^j P(x, y[j])$, if not equal to zero, is larger or equal to $2^{j+\nu_j}$. In particular,

$$w[n+1] \geq 2^{n+1+\nu_{n+1}}$$

For our example $A_{30} = A_{05} = 1$ so that $\alpha_{30} = \alpha_{05} = 0$ and

$$\nu_{n+1} = -5(n+1)$$

Now we use the selection function to determine the maximum number of zeros. This is based on the following two properties:

1. If $y_{j+1} = 0$ then $w[j+1] = 2w[j]$
2. When $2w[j] \geq C[j]$ then $y_{j+1} = 1$.

Consider that k zeros occur after $y_{n+1} = 1$. Then, from the lower bound of $w[n+1]$ we obtain

$$w[n+k] = 2^{k-1} w[n+1] \geq 2^{k+n+\nu_{n+1}}$$

Since the value y_{n+1+k} is still zero, we must have

$$2w[n+k] < C[n+k]$$

which implies

$$2^{k+n+\nu_{n+1}+1} < C[n+k] \tag{9}$$

Hence,

$$k < \log_2 C[n+k] - n - 1 - \nu_{n+1}$$

resulting in

$$k \leq \lceil \log_2 C[n+k] \rceil + n - 2 - \nu_{n+1} \tag{10}$$

where

$$\nu_{n+1} = \min_{pq}(\alpha_{pq} - np - (n+1)q)$$

This expression for k is not useful because of the k in the right-hand side. To avoid this we replace $C[n+k+1]$ by its maximum value for $j \geq n+1$. As we will see in the applications, in the practical cases this can be replaced by the maximum of the first partial derivative.

For our example

$$C[j] = 5y[j]^4 + 10y[j]^3 2^{-(j+1)} + ...$$

so that

$$\max(C[j]) = 5$$

Hence the bound is

$$k \leq 3 - n - 2 + 5(n+1) = 3 + 4n + 3 \leq 4n + 6$$

2.2 Longest run of ones

The longest run of ones starting from position $n+1$ in $f(x)$ can easily be deduced from the previous results, by noticing that a run of ones in $f(x)$ is a run of zeros in $1 - f(x)$.

2.3 Longest runs starting in bit y_{n+1}

These runs are used for the directed rounding modes. Since our approach can be used for runs starting at any position, a similar analysis as before shows that the bound for this case is

$$k \leq \lceil \log_2 C[n+k] \rceil - n - 1 - \nu_n \tag{11}$$

2.4 General ranges

Up to now we have assumed that $1/2 \leq x < 1$ and $1/2 \leq y < 1$. For the general case, we split the domain into subdomains such that $2^i \leq x < 2^{i+1}$ and $2^j \leq y < 2^{j+1}$. Then for each subdomain we apply the theorem with $X = 2^{-i-1}x$ and $Y = 2^{-j-1}y$. If $P(x,y) = 0$, we then have $P(2^{i+1}X, 2^{j+1}Y) = 0$. This $P(2^{i+1}X, 2^{j+1}Y)$, ONCE MULTIPLIED by the right power of 2 so that it has integer coefficients that are as small as possible (if they are not the smallest possible, the bounds are not wrong, they just are rougher) is a new polynomial $Q(X,Y)$ that is used for the theorem.

For instance, consider the reciprocal function for $1/2 \leq x < 1$. We have $i = -1$ (so $X = x$) and $j = 0$ (so $Y = 2^{-1}y$). Since $P(x,y) = 1 - yx$, we get

$$P(X, 2Y) = 1 - 2YX$$

$$A_{11} = -2, \;\; \alpha_{11} = 1$$

$$A_{00} = 1, \;\; \alpha_{00} = 0$$

$$\nu_{n+1} = 1 - n - (n+1) = -2n \;\; (w[n+1] \geq 2^{-(n-1)})$$

$$C[j] = 2x$$

$$\max(C[j]) = 2$$

$$k \leq log(2) - n - 2 - 2n = n - 1$$

2.5 Application to some usual functions

2.5.1 Reciprocal

This is illustrated in the previous section, resulting in

$$k \leq n - 1 \tag{12}$$

2.5.2 Division

For division $y = a/x$ and $1/2 \leq a, x < 1$, we have to consider two cases:

- $a < x$. In this case $1/2 \leq y \leq 1$ so

$$P(x, y = a - yx$$

$$A_{11} = -1, \;\; \alpha_{11} = 0$$

$$A_{00} = a, \;\; \alpha_{00} = -1$$

$$\nu_{n+1} = 0 - n - (n+1) = -2n - 1$$

$$C[j] = x$$

$$\max(C[j]) = 1$$

$$k \leq 0 - n - 2 + 2n + 1 = n - 1 \tag{13}$$

- $a > x$. In this case $1 \leq y < 2$. So, $Y = 2^{-1}y$ and we have the same bound as for the reciprocal.

2.5.3 Square root

$$P(x, y) = x - y^2$$

$$A_{02} = -1, \quad \alpha_{02} = 0$$

$$A_{10} = 1, \quad \alpha 10 = 0$$

$$\nu_{n+1} = -2(n+1)$$

$$C[j] = 2y[j] + 2^{-(j+1)}$$

$$\max(C[j]) = 2$$

$$k \leq 1 - n - 2 + 2(n+1) = n + 1 \tag{14}$$

2.5.4 q root

$$P(x,y) = x - y^q$$
$$A_{0q} = -1, \quad \alpha_{0q} = 0$$
$$A_{10} = a, \quad \alpha_{10} = 0$$
$$\nu_{n+1} = -(n+1)q$$
$$C[j] = qy[j] + ...$$
$$\max(C[j]) = q$$
$$k \leq \lceil log_2 q \rceil - n - 2 + (n+1)q = log_2 q + (n+1)(q-1) - 1 \tag{15}$$

2.5.5 Inverse square roots

To obtain results that are valid for any floating-point input, we must find results for x belonging to two consecutive binades. Hence, we consider two cases:

- $y = 1/(2\sqrt{x})$, $x, y \in [1/2, 1)$. This corresponds to the case of floating-point inputs in intervals of the form $[2^{2i-1}, 2^{2i}]$. We get

$$P(x,y) = 1 - 4xy^2$$
$$A_{00} = 1 \;\; \alpha_{00} = 0$$
$$A_{12} = 4, \quad \alpha_{12} = 2$$
$$\nu_{n+1} = \min_{p,q} \{\alpha_{pq} - np - (n+1)q\} = \min(0, -3n) = -3n.$$
$$C[j] = 8xy[j] + ...$$
$$\max(C[j]) = 8$$
$$k \leq \log_2(8) - n - 2 + 3n = 2n + 1 \tag{16}$$

- $y = 1/\sqrt{2x}$, $x, y \in [1/2, 1)$. This corresponds to the case of floating-point inputs in intervals of the form $[2^{2i}, 2^{2i+1}]$. We get

$$P(x,y) = 1 - 2xy^2$$

and we easily deduce the same bound as in the previous case.

2.5.6 Norms

The bound is the same as for square root, namely,

$$k \leq n + 1 \tag{17}$$

2.6 Normalization of 2-D vectors

The function to compute is $y = x/\sqrt{x^2 + v^2}$ for $1/2 \leq x, v < 1$ resulting in $1/2 \leq y < 1$. The algebraic function is

$$P(x, y) = x^2 - (x^2 + v^2)y^2$$

Then

$$A_{20} = A_{22} = 1, \quad \alpha_{20} = \alpha_{22} = 0$$

$$A_{02} = v^2 \quad \alpha_{02} = 2log_2 v$$

$$\nu_{n+1} = \min(0 - 2n,\ 0 - 2n - 2(n+1), -2 - 0 - 2(n+1)) = -4n - 2$$

$$C[j] = 2(x^2 + v^2)y$$

$$\max(C[j]) \leq 4$$

$$k \leq 2 - n - 2 + 4n + 2 = 3n + 2 \tag{18}$$

3 Actual maxima for some functions

In the previous section, we have obtained bounds on k. It is worth considering whether these bounds are attained or not.

3.1 Reciprocal

For this function it is easy to find a case that attains the bound for the run of zeros. Namely, for $1/2 \leq x, y < 1$ if $x = 1 - 2^{-n}$ we get $y = 1/x = 1 + 2^{-n} + 2^{-2n} + ...$ Since this is larger than 1, it is necessary to normalize resulting in a 1 in position $n + 1$ and the next 1 in position $2n + 1$. Consequently, there are $k = n - 1$ consecutive zeros.

For the run of ones it can be shown that it is attained for n odd. Namely, for $x = 1 - 2^{-(n-1)/2} + 2^{-n}$ results in $y = 1 + 2^{-(n-1)/2} - 2^{-n} + 2^{-2n}....$ This can be easily shown by defining $t = 2^{-(n-1)/2}$ and obtaining the Taylor expansion of $1/(1 - t + t^2/2)$. For n even, by exhaustive search, we found that the bound is not attained for $n \leq 16$.

3.2 Division

For the run of zeros the same case as for reciprocal applies. For the run of ones consider the case $q = a/b$ with $A = 2^n a = 3 \times 2^{n-2} + 2$ and $B = 2^n b = 2^{n-1} + 1$. Then, defining $t = 1/(2^{n-1})$, the quotient A/b is $f(t) = (3+2t)/(1+t)$ with Taylor expansion $3 - t + t^2 - t^3 + t^4 - ....$ Hence $A/B = 3 - 2^{-n+1} + 2^{-2n+2} - ....$ So its first bits are: 1.100000..0 (n bits) 0 (round bit) 1111111...1 ($n-1$ ones) 000..

3.3 Square root

Here we show that the actual maximum number of consecutive zeros after y_{n+1} is $k = n-1$. We first tighten the bound and then show the existence of a case that satisfies this bound.

From the bound obtained in the previous section we get

$$w[n+1] \geq 2^{-(n+1)} = e2^{-(n+1)}, \;\; e\; integer \tag{19}$$

Moreover, since $y_{n+1} = 1$ (rounding bit), we obtain from the recurrence

$$w[n+1] = 2w[n] - 2y[n] - 2^{-(n+1)} \tag{20}$$

Replacing $w[n]$ by its definition ($w[n] = 2^n(x - y[n]^2)$) results in

$$2^{n+1}(x - y[n]^2) - 2y[n] - 2^{-(n+1)} = e2^{-(n+1)} \tag{21}$$

Calling $(2^n)x = b$ (integer) and $(2^n)y[n] = a$ (integer) we get

$$e + 1 = 4(2^n b - a^2 - a) = 4(2^n b - a(a+1)) \tag{22}$$

Since $2^n b$ and $a(a+1)$ are even we get

$$e + 1 \geq 8$$

resulting in

$$e \geq 7 \tag{23}$$

That is, the minimum residual is bounded by $7 \times 2^{-(n+1)}$ and consequently $k \leq n-1$. Now we need to show that this bound is achieved. Making $e = 7$ in (22) we get

$$(2^n)b - a(a+1) = 2$$

with $2^{n-1} \leq a < 2^n$ and $2^{n-2} \leq b < 2^n$

We now need to show that this equation has an integer solution in the range of a and b. It is possible to prove by induction the following solution recurrence:

$$a(2) = 2 \quad b(2) = 2$$

If $b(i)$ is odd THEN $a(i+1) = 2^i + a(i)$
If $b(i)$ even THEN $a(i+1) = 2^{i+1} - a(i) - 1$
Moreover, $b(i+1) = [a(i+1)(a(i+1)+1)+2]2^{-(i+1)}$
Now we can prove by induction that these values are inside the range:
If $2^{i-1} \leq a(i) < 2^i$ then
for $b(i)$ odd $2^i + 2^{i-1} \leq A(i+1) = 2^i + A(i) < 2^{i+1}$ (inside the range)
for $b(i)$ even $2^{i+1} - 2^i - 1 \leq a(i+1) = 2^{i+1} - a(i) - 1 < 2^{i+1} - 2^{i-1} - 1$ (inside the range).
Similarly, we can show that $b(i)$ is inside the required range.
As a consequence, there is always a solution inside the range and the bound is achieved.
This method can also be used to obtain an example, even for large precision. For instance for quad precision ($n = 114$) the values are (in hex)
a = '30720461FD6E2F325A24E31B39FA5'
b = '24ABD1B29988F5099209B6D03CC18'
(obtained very fast with Maple).
Finally, for the run of ones two simple cases that achieve $k = n+1$ are $\sqrt{0.11111...11111}$ and $\sqrt{1.0000...000001}$.

3.4 Cube root

The case of cube root seems quite different from square roots. We have not found a "general formula" for the worst case. For small values of n, we have performed an exhaustive search, the result of which is given in Table 1.

3.5 Inverse square root

As for cube roots, we have not found a general formula for the maximum value. Using an analysis similar to that of square root we obtain that the minimum residual $e = w[n+1]2^{2n+1}$ is given by the expression

$$e = 2^{3n+1} - x(2y[n]+1)^2$$

Using this formula, by exhaustive search, we have found the worst cases for n from 4 to 20, and for $n = 24, 32$ and 53 (see Table 2). For $n \leq 20$, we have used a Maple program

Table 1: Worst cases for cube roots for n from 4 to 20.

n	x (binary)	$x^{1/3}$	k
4	0.01001	$0.1010011110111\cdots$	4
5	0.0011101	$0.100111000001000\cdots$	5
6	0.0101110	$0.1011011000000001111\cdots$	8
7	0.01001111	$0.1010110011111110111\cdots$	7
8	0.10011101	$0.1101100110000000000001001\cdots$	10
9	0.00100010100	$0.1000001101000000000001110\cdots$	11
10	0.01000000001	$0.1010000101011111111110100\cdots$	9
11	0.010101010001	$0.1011000101010000000000000001111\cdots$	15
12	0.00111010010100	$0.1001110001011000000000000001001\cdots$	13
13	0.01100010100111	$0.1011101001000011111111111111110011\cdots$	14
14	0.011101011000001	$0.1100010101111001111111111111110101\cdots$	13
15	0.0100100101111001	$0.10101000110111001^{17}0101\cdots$	17
16	0.0010011011011 10000	$0.1000100010001111110^{21}1110\cdots$	21
17	0.10000011001001101	$0.1100110011010111010^{19}1010\cdots$	19
18	0.0101011111110000011	$0.1011001100111111010 1^{19}0100\cdots$	19
19	0.0111111010001000011	$0.1100101001101000011 10^{22}1010\cdots$	22
20	0.01101101111011101001	$0.1100000100100011010101^{23}0101\cdots$	23

and for the larger values of n, we have used a dozen of workstations in parallel during a few days, running a program implementing the algorithm presented in [3][3] Note that there are many values of n (at least, $n = 7, 9, 10, 11, 12, 13$ and 18) for which the worst cases are $0.111111\ldots110 \times 2^p$, where p is an even exponent (in Table 2, this corresponds to the values of the form 11.1111…110).

3.6 Norms

The square root, cube root or square root reciprocal of an n-bit number is never exactly halfway between two consecutive n-bit numbers. This is no longer true when we consider norms. The difference with square root is due to the fact that for $1/2 \le x, v < 1$ the norm can be larger than 1 and because $x^2 + y^2$ has $2n$ bits. For instance, consider the 5-bit binary numbers $x = 0.10101 = 21/32$ and $v = 0.11100 = 7/8$. A straightforward calculation shows that $N(x, v) = 35/32 = 1.00011$.

We now show that the bound $k = n + 1$ is achieved. We follow the same approach as for square root. Namely, similarly to (21), we have

[3]We thank Vincent Lefèvre for the help with this.

Table 2: Worst cases for inverse square roots for n from 4 to 20,and for $n = 24, 32$ and 53 and $1 \leq x < 4$ (which obviously suffices to deduce all worst cases).

n	x (binary)	$1/\sqrt{x}$	k
4	1.101	$0.110010001101\cdots$	3
5	11.110	$0.10000100001100\cdots$	4
6	11.0100	$0.10001110000000001101\cdots$	9
7	11.11110	$0.10000001000000100\cdots$	6
8	10.011011	$0.10100100011111111110100\cdots$	9
9	11.1111110	$0.10000000010000000001100\cdots$	8
10	11.11111110	$0.10000000001000000000001100\cdots$	9
11	11.111111110	$0.10000000000010000000000001100\cdots$	10
12	11.1111111110	$0.10000000000010000000000001100\cdots$	11
13	11.11111111110	$0.10000000000001000000000000001100\cdots$	12
14	1.0010001110011	$0.1110111111011101^{14}0101\cdots$	14
15	11.1000101000110	$0.1000100000010000^{16}10111\cdots$	16
16	10.11111110001000	$0.1001001111111011^{15}1000\cdots$	15
17	1.0111111000101100	$0.1101000110000101110^{19}1011\cdots$	19
18	11.1111111111111110	$0.1000000000000000010^{17}1100\cdots$	17
19	1.100010000011110011	$0.1100111011010001000010^{23}1100\cdots$	23
20	1.0000101100011111101	$0.1111101010011100111110^{20}1000\cdots$	20
24	10.111010000110001110001 1	$0.1001011000100000100111100^{127}0100\cdots$	27
32	1.000111100000110110001011 0101101	$0.11110010001011011101110101 00101010^{32}1011\cdots$	32
53	1.101001101010100111001100 00010101101010111100110011110	$0.11000111001110111110100001000110001011000110000100101010101^{57}0100\cdots$	57

$$2^{n+1}(x^2 + v^2 - y[n]^2) - 2y[n] - 2^{-(n+1)} = e2^{-(n+1)}$$

Calling now $2^(2n)(x^2 + v^2) = b$ (integer) and $2^n y[n] = a$ (integer) we get

$$b - a(a + 1) = 2$$

with $2^{n-1} \leq a < 2^n$. An integer solution inside the range is given by the following recurrence:

$$a(3) = 5 \quad b[3] = 32$$

$$a(i+1) = 2^{i+1} - a(i) - 1 \quad b(i+1) = a(i+1)[a(i+1)+1] + 2$$

This produces the following examples (in hex):
for quad (n=113)
a[113] = '15555555555555555555555555555'
b[113] = '1C71C71C71C71C71C71C71C71C71CE38E38E38E38E38E38E38E38E390'
for double (n=53)
a[53] = '15555555555555'
b[53] = '1C71C71C71C71CE38E38E38E390'

3.7 Normalization of $2 - D$ vectors

Table 3 shows the actual values of k for n from 3 to 10. Although the bound of $3n + 2$ is far from achieved, there are some values which are significantly larger than n, which is the value expected by probabilistic considerations.

Table 3: Worst cases for normalization of 2D-vectors (x, v) (that is, function $x/\sqrt{x^2+v^2}$) for n from 3 to 10, with $1 \leq x, v < 2$.

n	x (binary)	v (binary)	$x/\sqrt{x^2+v^2}$	k
3	0.111	0.101	$0.110100000101\cdots$	5
4	0.1010	0.1000	$0.110001111110011\cdots$	6
5	0.11001	0.10101	$0.11000100000001010\cdots$	7
6	0.100000	0.110010	$0.10001001111111110110\cdots$	9
7	0.1111011	0.1011100	$0.11001100111111111111010\cdots$	12
8	0.10010110	0.10101000	$0.10101010100000000000001001\cdots$	13
9	0.110010001	0.111011000	$0.10100101101111111111111110100\cdots$	16
10	0.1101010000	0.1111111111	$0.10100011010111111111111111110101\cdots$	17

4 Comparison with results given in [1] and conclusions

Table 4 shows a comparison with the results reported in [1]. As can be seen the specific results are essentially the same, with minor differences for reciprocal, square root, and inverse square root. As indicated, we provide expressions for the bound for the class of algebraic functions.

Table 4: Comparison with results given in [1]

Function	run of ones		run of zeros	
	I-M	Our	I-M	Our
Reciprocal	$\leq n-1$	$= n-1$(n odd) $\leq n-1$(n even)	$= n-1$	$= n-1$
Division	$= n-1$	$= n-1$	$= n-1$	$= n-1$
Square root	$= n+1$	$n+1$	$\leq n-1$	$= n-1$
Inv. square root	$\leq 2n-1$	$\leq 2n+1$	$\leq 2n-1$	$\leq 2n+1$
Norm	–	$= n-1$	–	$= n-1$
q-root	–	$\leq \lceil log_2 q \rceil + (n+1)(q-1) - 1$	–	(same)
2D normalization	–	$\leq 3n+2$	–	$\leq 3n+2$
algebraic	–	$\leq \lceil log_2(max(C[j])) \rceil - n - 2 - \nu_{n+1}$	–	(same)

The main issue that is still pending is to obtain better bounds for those cases, such as inverse square root, for which the bounds are far from being achieved for specific values of n and show whether they are achieved for some n.

In Section 1.3 we have given a radix-2 restoring digit-recurrence algorithm for algebraic functions. As is well known for division and square root, this can be extended by the use of a redundant digit set for y_j (usually signed-digit) to simplify the selection function. Then, the arguments of the selection function are low precision estimates of the residual and of $C[j]$.

The feasability of the corresponding implementation depends on the complexity of the selection function and of the computation of $C[j]$.

References

[1] C. S. Iordache and D. W. Matula, *Infinitely Precise Rounding for Division, Square root, and Square Root Reciprocal*, Proc. 14th IEEE Symp. on Computer Arithmetic, 1999, pp. 233-240.

[2] V. Lefevre, J-M. Muller, and A. Tisserand, *Towards Correctly Rounded Transcendentals*, Proc. 13th IEEE Symp. on Computer Arithmetic. 1997, pp. 132-137.

[3] V. Lefevre and J-M. Muller, *Worst Case for Correct Rounding of the Elementary Functions in Double Precision*, submitted to ARITH15.

[4] P. W. Markstein, *Computation of elementary Functions on the IBM RISC System/6000 Processor*, IBM J. Res. Develop., vol. 34, no. 1, Jan. 1990, pp. 111-119.

[5] M. Schulte and E.E. Swartzlander, *Exact Rounding of Certain Elementary Functions*, 11th Symp. on computer Arithmetic, 1993, pp. 138-145.

Unit´e de recherche INRIA Lorraine, Technopôle de Nancy-Brabois, Campus scientifi que,
615 rue du Jardin Botanique, BP 101, 54600 VILLERS LÈS NANCY
Unit´e de recherche INRIA Rennes, Irisa, Campus universitaire de Beaulieu, 35042 RENNES Cedex
Unit´e de recherche INRIA Rhône-Alpes, 655, avenue de l'Europe, 38330 MONTBONNOT ST MARTIN
Unit´e de recherche INRIA Rocquencourt, Domaine de Voluceau, Rocquencourt, BP 105, 78153 LE CHESNAY Cedex
Unit´e de recherche INRIA Sophia-Antipolis, 2004 route des Lucioles, BP 93, 06902 SOPHIA-ANTIPOLIS Cedex

Éditeur
INRIA, Domaine de Voluceau, Rocquencourt, BP 105, 78153 LE CHESNAY Cedex (France)
http://www.inria.fr
ISSN 0249-6399