Exponential sums and correctly-rounded functions

Nicolas Brisebarre, Guillaume Hanrot, Olivier Robert

Exponential sums and correctly-rounded functions

Nicolas Brisebarre, Guillaume Hanrot and Olivier Robert

Abstract—The 2008 revision of the IEEE-754 standard, which governs floating-point arithmetic, recommends that a certain set of elementary functions should be correctly rounded. Successful attempts for solving the Table Maker's Dilemma in binary64 made it possible to design `CRlibm`, a library which offers correctly rounded evaluation in binary64 of some functions of the usual `libm`. It evaluates functions using a two step strategy, which relies on a folklore heuristic that is well spread in the community of mathematical functions designers. Under this heuristic, one can compute the distribution of the lengths of runs of zeros/ones after the rounding bit of the value of the function at a given floating-point number. The goal of this paper is to change, whenever possible, this heuristic into a rigorous statement. The underlying mathematical problem amounts to counting integer points in the neighborhood of a curve, which we tackle using so-called exponential sums techniques, a tool from analytic number theory.

1 Introduction

On most current computer systems, real numbers are approximated and represented by floating-point numbers [29]. For many years, floating-point arithmetic has been a mere set of cooking recipes, a situation described for instance in [18], and numerical programs were not reliable nor portable.

The IEEE-754 [8], [1] standard, and its revision [16], [26], for binary floating-point arithmetic (and the radix independent IEEE-854 [7], [2] standard that followed) drastically improved the situation and put an end to this dangerous era. In particular, the IEEE-754 standard clearly specifies the formats of the floating-point representations of numbers, and the behaviour of the four arithmetic operations and the square root.

And yet, as of today, the standard still does not rule the behaviour of usual functions, such as the ones contained in the C mathematical library (`libm`), as precisely as it does for the four arithmetic operations and the square root. Part of a general effort to improve this situation[29], this paper presents a novel rigorous approach for estimating the amount of so-called bad cases for rounding. As we shall see below, it is useful to design function evaluation routines with a higher quality of computation called correct rounding, and in particular, it plays a key role in order to estimate their performances.

- *N. Brisebarre and G. Hanrot are with Université de Lyon, CNRS, ENS de Lyon, Inria, Université Claude-Bernard Lyon 1, Laboratoire LIP (UMR 5668), Lyon, France.*
  E-mail: nicolas.brisebarre@ens-lyon.fr, guillaume.hanrot@ens-lyon.fr
- *O. Robert is with Université de Lyon, CNRS, Université Claude-Bernard Lyon 1, INSA Lyon, Centrale Lyon, Université de Saint-Étienne, Institut Camille Jordan (UMR 5208), Saint-Étienne, France.*
  E-mail: olivier.robert@univ-st-etienne.fr
- *This work was partly supported by the TaMaDi project of the French* Agence Nationale de la Recherche.

1.1 Our arithmetic framework

Let $\mathcal{D}_p$ denote the set of exponent-unbounded, p-bit significand, radix-2 floating-point numbers (with $p \geqslant 1$):

$$\mathcal{D}_p = \left\{ j \cdot 2^{E-p+1}, 2^{p-1} \leqslant |j| \leqslant 2^p - 1, j, E \in \mathbb{Z} \right\} \cup \{0\}.$$

This set is not the set of available floating-point numbers on an existing system. It is an "ideal" system of floating-point numbers: a superset of an actual system, with no overflows, underflows, or subnormals [16], [29]. We prove results in $\mathcal{D}_p$. These results remain true in an actual system, provided that no overflows, underflows or subnormals occur.

We call significand of a nonzero element $x = j \cdot 2^{E-p+1}$ of $\mathcal{D}_p$ the number $j/2^{p-1}$ and integral significand of x the integer j. The exponent of an element $j \cdot 2^{E-p+1}$, with $2^{p-1} \leqslant |j| \leqslant 2^p - 1$, is the integer E.

The result of an arithmetic operation whose input values belong to $\mathcal{D}_p$ may not belong to $\mathcal{D}_p$ (in general it does not). Hence that result must be rounded. The IEEE standard defines 4 different rounding modes; in the sequel, x is any real number to be rounded:

- rounding towards $+\infty$, or upwards: $\circ_u(x)$ is the smallest element of $\mathcal{D}_p$ that is greater than or equal to x;
- rounding towards $-\infty$, or downwards: $\circ_d(x)$ is the largest element of $\mathcal{D}_p$ that is less than or equal to x;
- rounding towards 0: $\circ_z(x)$ is equal to $\circ_u(x)$ if $x < 0$, and to $\circ_d(x)$ otherwise;
- rounding to the nearest even: $\circ_n(x)$ is the element of $\mathcal{D}_p$ that is closest to x. If x is exactly halfway between two consecutive elements of $\mathcal{D}_p$, $\circ_n(x)$ is the one for which the integral significand j is an even number.

The first three rounding modes are called directed rounding modes.

1.2 Correct rounding

The standard requires that the user should be able to choose one rounding mode among these ones, called the *active rounding mode*. An active rounding mode being chosen, when

performing one of the 4 arithmetic operations, or when computing square roots, the obtained rounded result should be equal to the rounding of the exact result: this requirement on the quality of the computation is called *correct rounding*.

While the IEEE 754-1985 and 854-1987 standards required correctly rounded arithmetic operations, they did not do it for the most common mathematical functions, such as simple algebraic[1] functions like $1/\sqrt{\cdot}$, $\sqrt[3]{\cdot}$, ... and also a few transcendental[2] functions like sine, cosine, exponentials, and logarithms of radices e, 2, and 10, etc. More generally, a natural target is the whole class of elementary functions[3]. A subset of these functions is usually available from the `libm`s delivered with compilers.

This lack of requirement is mainly due to a difficult problem known as the Table Maker's Dilemma (TMD), a term coined by Kahan, which we present below.

Being able to provide correctly rounded functions is of utter interest:

- it greatly improves the portability of software;
- it allows one to design algorithms that use this requirement;
- this requirement can be used for designing formal proofs of pieces of software;
- one can easily implement interval arithmetic, or more generally one can get certain lower or upper bounds on the exact result of a sequence of arithmetic operations.

1.3 Implementation of correctly-rounded functions: Ziv's strategy

A. Ziv proposed a general methodology [37], implemented in the `libultim` library[4], that made possible the correctly rounded evaluation of the functions of the `libm`. It consists in iteratively improving the accuracy of the approximation until the correctly rounded value can be decided.

However, there was, up until recently, no practical bound on the termination time of Ziv's iteration: it may be proven to terminate for some transcendental functions, but the actual maximal accuracy required in the worst case was unknown. Note that in order to prove that Ziv's iteration terminates for a given real-valued function φ and a given floating-point number x, one has to guarantee a priori that $\varphi(x) \notin \mathcal{D}_p$ (directed rounding modes) or $\varphi(x)$ is not the middle of two consecutive elements of $\mathcal{D}_p$ (rounding to nearest mode). One has to deal separately with the exact cases such as $\exp(0), \log(1), \sin(0)$ or $\sqrt[3]{x^3}$ for $x \in \mathcal{D}_{\lfloor p/3 \rfloor}$ etc. [21], [22], [28].

Unfortunately, for many applications, this is not satisfactory, for several reasons:

- in `libultim`, the measured worst-case execution time is indeed three orders of magnitude larger than that of the usual `libm`s;
- A related problem is memory requirement, which is, for the same reason, unbounded in theory and much higher than the usual `libm`s in practice;
- when the approximations are evaluated using a fast pipelined multiplier or multiplier-accumulator, the tests required by the previous strategy would require to wait many cycles before being able to restart the computation;
- for real-time applications, the delay of computation must be bounded.

1.4 Fast and cheap correctly-rounded function evaluation in binary64

For most of elementary functions, when evaluating them, one has to compute some approximation to the exact result, with an accuracy (which is then a rational integer power of the radix 2) somewhat higher than the "target" precision p. The Table Maker's Dilemma is the problem of determining what the accuracy of this approximation should be to make sure that rounding that approximation will always be equivalent to rounding the exact result. Ideally, we aim at getting the minimal possible accuracy.

On the one hand, there are theoretical results that yield - not fully satisfactory - solutions for algebraic functions: the precision which the computations must be performed to is, in general, overestimated [17], [20], [6]. On the other hand, regarding transcendental functions, either no theoretical statement exists or they provide results that cannot be used in practical computations [30].

Therefore, algorithmic approaches to the TMD [23], [24], [34] had to be developed. They allowed for solving the TMD for the IEEE binary64 (also known as "double precision"). As a consequence, the revised IEEE-754 standard now recommends (yet does not require, due to the lack of results in the case of binary128) that the following functions should be correctly rounded: e^x, $e^x - 1$, 2^x, $2^x - 1$, 10^x, $10^x - 1$, $\ln(x)$, $\log_2(x)$, $\log_{10}(x)$, $\ln(1 + x)$, $\log_2(1 + x)$, $\log_{10}(1+x)$, $\sqrt{x^2 + y^2}$, $1/\sqrt{x}$, $(1+x)^n$, x^n, $x^{1/n}$ (n is an integer), $\sin(\pi x)$, $\cos(\pi x)$, $\arctan(x)/\pi$, $\arctan(y/x)/\pi$, $\sin(x)$, $\cos(x)$, $\tan(x)$, $\arcsin(x)$, $\arccos(x)$, $\arctan(x)$, $\arctan(y/x)$, $\sinh(x)$, $\cosh(x)$, $\tanh(x)$, $\sinh^{-1}(x)$, $\cosh^{-1}(x)$, $\tanh^{-1}(x)$.

Thanks to these results, it is now possible to obtain correct rounding in binary64 in two Ziv steps only, which one may then optimize separately. This is the approach used in `CRlibm`[5], a library which offers correctly rounded evaluation of the double-precision C99 standard elementary functions:

- the first quick step is as fast as a current `libm`, and provides an accuracy of 2^{-52-k} ($k = 11$ for the exponential function for instance), which is sufficient to round correctly to the 53 bits of binary64 in most cases;
- the second accurate step is dedicated to challenging cases. It is slower but has a reasonable bounded execution time, being tightly targeted at the hardest-to-round cases computed by Lefèvre et al. [25], [24], [33], [34], [32]. In particular, there is no need for arbitrary multiple precision anymore.

1. We say that a function φ is algebraic if there exists $P \in \mathbb{Z}[x, y] \setminus \{0\}$ such that $P(x, \varphi(x)) = 0$.

2. A function is transcendental if it is not algebraic.

3. An elementary function is a function of one variable which is the composition of a finite number of arithmetic operations ($+$, $-$, $\times$, $/$), exponentials, logarithms, constants, and solutions of algebraic equations.

4. `libultim` was released by IBM. An updated version is now part of the GNU `glibc` and available under the GNU General Public License.

5. http://lipforge.ens-lyon.fr/www/crlibm/

This approach [9], [10] leads to correctly-rounded function evaluation routines that are fast and have a reasonable memory consumption.

1.5 A heuristic probabilistic approach

This strategy relies on the following heuristic assumption about the first step: the proportion of "bad cases", that is to say the floating-point numbers whose evaluation cannot be correctly rounded if we use 2^{-52-k} as an intermediate accuracy, is around 2^{1-k}. One can find such an estimate used in [11] and a probabilistic study has been done in [13]. We now briefly recall the intuition behind this heuristic (see [28], [29] for a more detailed presentation).

Let φ be a real-valued function, assume that after the p^{th} bit, the bits of the significands of the values $\varphi(x)$, where x is a floating-point number, are sequences of independent random 0 or 1 with equal probability $1/2$. The probability that after bit p, we have

- in **rounding to nearest mode**, the bit sequence

$$\underbrace{100\cdots0}_{k \text{ bits}} \quad \text{or} \quad \underbrace{011\cdots1}_{k \text{ bits}}$$

- or, in **directed rounding modes**, the bit sequence

$$\underbrace{00\cdots0}_{k \text{ bits}} \quad \text{or} \quad \underbrace{11\cdots1}_{k \text{ bits}}$$

is 2^{-k+1}. Hence, if we have N floating-point numbers in the domain being considered, the number of values x for which we will have a bit sequence of the form indicated above is, under the probabilistic model stated above, around $N2^{-k+1}$.

1.6 Goal and outline of the paper

The assumption on which this approach relies is a very strong one, and, actually, is wrong in full generality (one can think, for instance, of the exponential function around 0). And yet, it seems to be satisfied in most cases and as of today, proving it seems completely out of reach. The goal of this paper is to give solid theoretical foundations to this heuristic probability for certain ranges of values of k. We target in particular the values of k that `CRlibm` uses in practice. For instance, our results prove in particular that:

- in binary64, for $k = 11$, the number of bad cases for
  - $\sqrt[3]{\cdot}$ over $[1/2, 1)$ lies between $(1 - 0.058) \cdot 2^{42}$ and $(1+0.058)\cdot 2^{42}$, which leads to a probability between $(1-0.058)\cdot 2^{-10}$ and $(1+0.058)\cdot 2^{-10}$ for the occurrence of the second accurate step;
  - $\exp$ over $[1, 2)$ lies between $(1-0.054)\cdot 2^{42}$ and $(1 + 0.054) \cdot 2^{42}$, which leads to a probability between $(1-0.054)\cdot 2^{-10}$ and $(1+0.054)\cdot 2^{-10}$ for the occurrence of the second accurate step;
- in binary128, for $k = 32$, the number of bad cases for
  - $\sqrt[3]{\cdot}$ over $[1/2, 1)$ lies between $(1 - 0.112) \cdot 2^{81}$ and $(1+0.112)\cdot 2^{81}$, which leads to a probability between $(1-0.112)\cdot 2^{-31}$ and $(1+0.112)\cdot 2^{-31}$ for the occurrence of the second accurate step;
  - $\exp$ over $[1, 2)$ lies between $(1 - 0.11) \cdot 2^{81}$ and $(1 + 0.11) \cdot 2^{81}$, which leads to a probability between $(1 - 0.11) \cdot 2^{-31}$ and $(1 + 0.11) \cdot 2^{-31}$ for the occurrence of the second accurate step.

We shall start with a mathematical formalization of the problem we address. This will be done in Section 2. In Section 3, we shall present estimates, proved thanks to the so-called exponential sums techniques, a mathematical tool at the core of our study. These objects and the related theory are powerful tools mainly used in Analytic Number Theory [14], [15], [27], [35], [19], [31]. Then, in Section 4, we shall apply the results of Section 3 to elementary functions. This will enable us to estimate the number of occurrences of the second accurate step of the two Ziv step strategy, like the one used in the `CRlibm` design. We shall also analyze the constraints imposed by our approach (in particular, the ranges the parameter k should belong to) and present in more detail the cases of the $\sqrt[3]{\cdot}$ and $\exp$ functions. We shall report experiments with these two functions in binary64 and binary128 in order to evaluate the quality of our estimates.

Up to a renormalization (which associates a function f to the function φ, see (18) and (19)), our problem can be seen as a question about the number of integers m in some interval of the form $[1, M]$ such that the distance of $f(m)$ to the integers is less than some $\delta > 0$. This problem, which amounts to counting integer points (m, n) in the δ-neighbourhood of the curve $y = f(x)$, will be treated in Section 3, which is the core of the paper. The main term of this number of points is expected to be $2M\delta$. We shall express the error term as a sum of periodic functions (namely fractional parts), and control this error term using an optimal truncation of the Fourier series introduced by Vaaler (Theorem 1). The problem then reduces to estimating exponential sums, for which we shall use van der Corput's inequality (Theorem 2).

Note that, actually, our results are valid for any $\mathcal{C}^2$ function[6], and not only for elementary functions.

Our work may be considered as complementary to the algorithmic determinations of the worst cases for correct rounding [23], [24], [34] in two ways:

- while their works allow for a worst case analysis of an elementary function implementation, our paper provides an attempt for a rigorous average case analysis of such an implementation. We shall come back to this in Remark 2.
- as the reader will notice in Subsection 4.1, our approach, roughly, works for values of k up to $p/3$ bits. In this range, the aforementioned algorithmic approaches may prove pointless since that there are too many values to determine, whereas our approach makes it possible to estimate the amount of bad cases for rounding in a satisfactory way.

2 Formalization of the problem

Assume we wish to correctly round an elementary function φ (actually, our formalization is valid for any real-valued function). We consider that all input values are elements of $\mathcal{D}_p$ with the same exponent e_1. A different analysis must be done for each possible value of e_1.

6. Recall that a $\mathcal{C}^\ell$ function over an interval I, $\ell \geqslant 0$, is a function that is ℓ-times differentiable and whose ℓ-th derivative is continuous.

Note that if x is a bad case for φ, then it is also a bad case for $-\varphi$ and $-x$ is a bad case for $t \mapsto \varphi(-t)$ and $t \mapsto -\varphi(-t)$. Hence we can assume that $x \geqslant 0$ and $\varphi(x) \geqslant 0$.

If the values of $\varphi(x)$, for $x \in [2^{e_1}, 2^{e_1+1})$, are not all included in the binade[7] $[2^{e_2}, 2^{e_2+1})$, we split the input interval into subintervals such that for each subinterval, there is a rational integer e_2 such that the values $\varphi(x)$, for x in the subinterval, are in $[2^{e_2}, 2^{e_2+1})$.

We now consider the processing of one subinterval I included in $[2^{e_1}, 2^{e_1+1})$.

In rounding to nearest mode, the problem to be solved is the following:

Problem 1 (Rounding to nearest mode)**.** *Given $k \in \mathbb{N}$, estimate the number of integers X, $2^{p-1} \leqslant X \leqslant 2^p - 1$ (and, possibly, the restrictions implied by $X/2^{-e_1+p-1} \in I$) such that there exists $Y \in \mathbb{N}$, $2^{p-1} \leqslant Y \leqslant 2^p - 1$,*

$$\left| \varphi\left(\frac{X}{2^{-e_1+p-1}}\right) - \frac{2Y+1}{2^{-e_2+p}} \right| < 2^{e_2-p-k}.$$

In directed rounding modes, the problem to be solved is the following:

Problem 2 (Directed rounding modes)**.** *Given $k \in \mathbb{N}$, estimate the number of integers X, $2^{p-1} \leqslant X \leqslant 2^p - 1$ (and, possibly, the restrictions implied by $X/2^{-e_1+p-1} \in I$) such that there exists $Y \in \mathbb{N}$, $2^{p-1} \leqslant Y \leqslant 2^p - 1$,*

$$\left| \varphi\left(\frac{X}{2^{-e_1+p-1}}\right) - \frac{Y}{2^{-e_2+p-1}} \right| < 2^{e_2-p-k}.$$

Now, we introduce the mathematical tools that will allow us to tackle Problems 1 and 2.

3 Some exponential sums-based estimates

First, let us mention that the reader who is mainly interested in the applications to the design of mathematical functions should focus on Theorem 3, Lemma 4 and Corollary 1, which are the technical results that will be used in practice.

Several deep results (see [3], [4], [5] for instance) in Analytic Number Theory rely on non-trivial bounding of the modulus of sums of the form $\sum_{a<m\leqslant a+M} \mathrm{e}(f(m))$, where $a \in \mathbb{Z}$, $M \in \mathbb{N}$, $f : [a+1, a+M] \to \mathbb{R}$ is a $\mathcal{C}^k$ function with $k \geqslant 1$ and $\mathrm{e}(t) := \exp(2i\pi t)$, $t \in \mathbb{R}$. The developed techniques prove very useful for estimating the number of integer points in the neighborhood of a curve, which actually is the usual context we face when we tackle the Table Maker's Dilemma. In-depth expositions of these techniques are presented in [14], [15], [27].

In the present paper we shall use techniques based on the second derivative, and thus assume f to be a $\mathcal{C}^2$ function; first-order techniques (à la Kuzmin-Landau) do not apply to the present context, while higher order methods (when applicable) yield worse upper bounds, probably useless in the formats we are interested in (typically binary64 and binary128).

Given $0 < \eta < 1/2$, we want to establish estimates of the value

$$\mathcal{R}(f;\eta) := \#\big\{1 \leqslant m \leqslant M,\ \|f(m)\| < \eta\big\}, \tag{1}$$

7. A binade is an interval of the form $[2^k, 2^{k+1})$ or $(-2^{k+1}, -2^k]$ for $k \in \mathbb{Z}$.

where $||x||$ denotes the distance of the real number x to $\mathbb{Z}$: $||x|| = \min(\{x\}, 1-\{x\})$, and $\{x\} = x - \lfloor x \rfloor$ is the fractional part of x.

When $\eta = 0$, we set

$$\mathcal{R}(f;0) := \#\left\{1 \leqslant m \leqslant M,\ f(m) \in \mathbb{Z}\right\}.$$

In our work, we follow the approach developed by Vaaler in [36]. Let $\psi(t)$ denote the normalized fractional part

$$\psi(t) := t - \lfloor t \rfloor - \tfrac{1}{2} \qquad (t \in \mathbb{R}).$$

The proof of our main result requires estimates of sums of fractional parts. Since our aim is to obtain upper bounds with suitable numerical constants, we shall use an appropriate approximation of $\psi(t)$, and a result due to Vaaler.

We consider the following function

$$\widehat{J}(t) := \begin{cases} 1 & \text{if } t = 0, \\ \pi t(1-|t|)\cot(\pi t) + |t| & \text{if } 0 < |t| < 1, \\ 0 & \text{if } |t| \geqslant 1. \end{cases}$$

The (Fourier series type) approximation of ψ we have in mind is the following:

$$\psi_H^*(t) := -\sum_{1\leqslant|h|<H} \frac{\widehat{J}\left(\frac{h}{H}\right)}{2i\pi h} \mathrm{e}(ht), \qquad t \in \mathbb{R},\ H \in \mathbb{N}.$$

We start by recalling an upper bound for the approximation error:

Theorem 1. *[36] Using previous notation, one has, for $t \in \mathbb{R}$, $H \in \mathbb{N}$,*

$$|\psi(t) - \psi_H^*(t)| \leqslant \frac{1}{2H} \sum_{|h|<H} \left(1 - \frac{|h|}{H}\right) \mathrm{e}(ht).$$

We shall now give an upper bound for $\mathcal{R}(f;0)$ in terms of exponential sums.

Lemma 1. *For any function $f : [1, M] \to \mathbb{R}$, one has*

$$\mathcal{R}(f;0) \leqslant \frac{1}{H} \sum_{|h|<H} \left(1 - \frac{|h|}{H}\right) \sum_{m=1}^{M} \mathrm{e}\left(hf(m)\right) \tag{2}$$

for any $H \in \mathbb{N}$.

Proof: We note that for any $H \in \mathbb{N}$ the term

$$\frac{1}{H^2}\left|\sum_{h=1}^{H} \mathrm{e}(ht)\right|^2 = \frac{1}{H} \sum_{|h|<H} \left(1 - \frac{|h|}{H}\right) \mathrm{e}(ht)$$

is nonnegative for any $t \in \mathbb{R}$, and is 1 for $t \in \mathbb{Z}$. It follows that

$$\mathcal{R}(f;0) \leqslant \sum_{m=1}^{M} \frac{1}{H} \sum_{|h|<H} \left(1 - \frac{|h|}{H}\right) \mathrm{e}\left(hf(m)\right),$$

which gives the expected result. □

We are now ready to state our main lemma.

Lemma 2. *Let $\delta \in \mathbb{R}_{>0}$, and $M, H \in \mathbb{N}$. Then for any function $f : [1, M] \to \mathbb{R}$ one has*

$$|\mathcal{R}(f;\delta) - 2M\delta| \leqslant E_1(H) + E_2(H) + \mathcal{R}(f+\delta;0) \tag{3}$$

and

$$|\mathcal{R}(f;\delta) - 2M\delta| \leqslant E_1(H) + 2E_2(H) \tag{4}$$

where we have set

$$E_1(H) := 2\sum_{h=1}^{H-1} \frac{|\sin(2\pi\delta h)|}{\pi h}\left|\sum_{m=1}^{M} \mathrm{e}\left(hf(m)\right)\right|, \qquad (5)$$

$$E_2(H) := \frac{1}{H}\sum_{|h|<H}\left(1-\frac{|h|}{H}\right)\left|\sum_{m=1}^{M} \mathrm{e}\left(hf(m)\right)\right|. \qquad (6)$$

Proof: We have

$$\begin{aligned}\mathcal{R}(f;\delta) &= \sum_{m=1}^{M} \#\{\nu\in\mathbb{Z}\colon f(m)-\delta<\nu<f(m)+\delta\}\\ &= \sum_{m=1}^{M}\left(\lfloor f(m)+\delta\rfloor - \lfloor f(m)-\delta\rfloor\right) - \mathcal{R}(f+\delta;0)\\ &= 2M\delta + \sum_{m=1}^{M}\left(\psi\left(f(m)-\delta\right)-\psi\left(f(m)+\delta\right)\right)\\ &\quad - \mathcal{R}(f+\delta;0).\end{aligned}$$

Now let

$$E := \sum_{m=1}^{M}\left(\psi\left(f(m)-\delta\right)-\psi\left(f(m)+\delta\right)\right)$$

and

$$E_H^* := \sum_{m=1}^{M}\left(\psi_H^*\left(f(m)-\delta\right)-\psi_H^*\left(f(m)+\delta\right)\right).$$

Then

$$\begin{aligned}|E-E_H^*| &\leqslant \sum_{m=1}^{M}\left|\psi\left(f(m)-\delta\right)-\psi_H^*\left(f(m)-\delta\right)\right|\\ &\quad + \sum_{m=1}^{M}\left|\psi\left(f(m)+\delta\right)-\psi_H^*\left(f(m)+\delta\right)\right|.\end{aligned}$$

Now Theorem 1 implies that for $\eta=\pm\delta$ one has

$$\begin{aligned}&\sum_{m=1}^{M}\left|\psi\left(f(m)+\eta\right)-\psi_H^*\left(f(m)+\eta\right)\right|\\ &\quad\leqslant \frac{1}{2H}\sum_{|h|\leqslant H}\left(1-\frac{|h|}{H}\right)\sum_{m=1}^{M}\mathrm{e}\left(h\left(f(m)+\eta\right)\right)\\ &\quad\leqslant \frac{1}{2H}\sum_{|h|\leqslant H}\left(1-\frac{|h|}{H}\right)\left|\sum_{m=1}^{M}\mathrm{e}\left(hf(m)\right)\right|,\end{aligned}$$

so that

$$|E-E_H^*| \leqslant E_2(H).$$

Next, we prove that $|E_H^*|\leqslant E_1(H)$. Indeed

$$\begin{aligned}E_H^* &= \sum_{m=1}^{M}\sum_{|h|=1}^{H-1}\frac{\widehat{J}\left(\frac{h}{H}\right)}{2i\pi h}\left(\mathrm{e}(h(f(m)-\delta))-\mathrm{e}(h(f(m)+\delta))\right)\\ &= \sum_{1\leqslant|h|<H}\frac{\widehat{J}\left(\frac{h}{H}\right)}{\pi h}\sin(-2\pi\delta h)\sum_{m=1}^{M}\mathrm{e}\left(h(f(m))\right)\end{aligned}$$

and the expected upper bound follows from the estimate $\left|\widehat{J}(t)\right|\leqslant 1$ for $t\in\mathbb{R}$.

Finally, writing

$$|\mathcal{R}(f;\delta)-2M\delta| \leqslant |E_H^*| + |E-E_H^*| + \mathcal{R}(f+\delta;0)$$

and using the previous estimates yields (3).

Now, to prove (4), it is sufficient to check that $\mathcal{R}(f+\delta;0)\leqslant E_2(H)$. This follows at once from (2). □

Remark 1. *The reader may have noticed that Lemma 2 provides two upper bounds for the same error $|\mathcal{R}(f;\delta)-2M\delta|$. The reason is the following: in practice, we shall set $\delta=2^{-k}, k\in\mathbb{N}$ and there are functions, such as* exp *or* cos *for instance, for which we will be able to estimate precisely the quantity $\mathcal{R}(f+2^{-k};0)$. In this case, inequality* (3) *will give us a better information than* (4)*. However, the latter has the advantage of being general.*

Theorem 3 and Corollary 1 are presented the same way, for the same reason.

We will need the following technical inequalities.

Lemma 3. *We have, for all $H\in\mathbb{N}$,*

$$\sum_{h=1}^{H}\frac{1}{h^{1/2}} \leqslant 2H^{1/2}; \qquad (7)$$

$$\sum_{h=1}^{H-1} h^{1/2}\left(1-\frac{h}{H}\right) \leqslant \frac{4}{15}H^{3/2}+H^{1/2}; \qquad (8)$$

$$\sum_{h=1}^{H-1}\frac{1}{h^{1/2}}\left(1-\frac{h}{H}\right) \leqslant \frac{4}{3}H^{1/2}; \qquad (9)$$

$$\sum_{h=1}^{+\infty}\frac{|\sin(2\pi\delta h)|}{h^{3/2}} \leqslant \frac{5}{2}(2\pi\delta)^{1/2}+(2\pi\delta)^{3/2}, \quad \delta>0. \qquad (10)$$

Proof: Note that the first three estimates are trivial for $H=1$. Hence in the sequel we suppose that $H\geqslant 2$.

1) For (7), we have

$$\sum_{h=1}^{H}\frac{1}{h^{1/2}} \leqslant \sum_{h=1}^{H}\int_{h-1}^{h}\frac{\mathrm{d}t}{t^{1/2}} = \int_0^H \frac{\mathrm{d}t}{t^{1/2}} = 2H^{1/2}.$$

2) For (8), we have

$$\sum_{h=1}^{H-1} h^{1/2} \leqslant \sum_{h=1}^{H-1}\int_h^{h+1} t^{1/2}\mathrm{d}t = \int_1^H t^{1/2}\mathrm{d}t \leqslant \frac{2}{3}H^{3/2}.$$

Moreover

$$\begin{aligned}\sum_{h=1}^{H-1}\frac{h^{3/2}}{H} &= -H^{1/2}+\sum_{h=1}^{H}\frac{h^{3/2}}{H}\\ &\geqslant -H^{1/2}+\sum_{h=1}^{H}\int_{h-1}^{h}\frac{t^{3/2}}{H}\mathrm{d}t\\ &= -H^{1/2}+\frac{1}{H}\int_0^H t^{3/2}\mathrm{d}t = -H^{1/2}+\frac{2}{5}H^{3/2}.\end{aligned}$$

Using these inequalities, we now have

$$\sum_{h=1}^{H-1} h^{1/2} - \sum_{h=1}^{H-1}\frac{h^{3/2}}{H} \leqslant \frac{2}{3}H^{3/2}-\frac{2}{5}H^{3/2}+H^{1/2}$$

which gives the expected result.

3) Similarly, for (9) we have

$$\sum_{h=1}^{H-1}\frac{1}{h^{1/2}}\left(1-\frac{h}{H}\right)=\sum_{h=1}^{H}\frac{1}{h^{1/2}}\left(1-\frac{h}{H}\right)$$
$$\leqslant\sum_{h=1}^{H}\int_{h-1}^{h}\frac{1}{t^{1/2}}\left(1-\frac{t}{H}\right)\mathrm{d}t$$
$$=\int_{0}^{H}\frac{1}{t^{1/2}}\left(1-\frac{t}{H}\right)\mathrm{d}t=\frac{4}{3}H^{1/2}.$$

4) We now estimate the last sum. We have

$$\sum_{h\leqslant 1/(2\pi\delta)}\frac{|\sin(2\pi\delta h)|}{h^{3/2}}\leqslant 2\pi\delta\sum_{h\leqslant 1/(2\pi\delta)}\frac{1}{h^{1/2}}$$
$$\leqslant 4\pi\delta\lfloor 1/(2\pi\delta)\rfloor^{1/2}\leqslant 2(2\pi\delta)^{1/2}$$

where we have used (8).

For the remaining terms, we write

$$\sum_{h\geqslant 1+\lfloor 1/(2\pi\delta)\rfloor}\frac{|\sin(2\pi\delta h)|}{h^{3/2}}\leqslant\sum_{h\geqslant 1+\lfloor 1/(2\pi\delta)\rfloor}\frac{1}{h^{3/2}}$$
$$=\frac{1}{(1+\lfloor 1/(2\pi\delta)\rfloor)^{3/2}}+\sum_{h\geqslant 2+\lfloor 1/(2\pi\delta)\rfloor}\frac{1}{h^{3/2}}$$
$$\leqslant(2\pi\delta)^{3/2}+\sum_{h\geqslant 2+\lfloor 1/(2\pi\delta)\rfloor}\int_{h-1}^{h}\frac{\mathrm{d}t}{t^{3/2}}$$
$$\leqslant(2\pi\delta)^{3/2}+\int_{1/(2\pi\delta)}^{+\infty}\frac{\mathrm{d}t}{t^{3/2}}\leqslant(2\pi\delta)^{3/2}+\frac{1}{2}(2\pi\delta)^{1/2}.$$

□

We now state an explicit version of van der Corput's inequality for exponential sums : this is Equation (6.14) of [35, p. 128].

Theorem 2. *Let $M\in\mathbb{N}$, $\lambda\in\mathbb{R}_{>0}$ and $C\in[1,+\infty)$. Let $f\in\mathcal{C}^2\big([1,M],\mathbb{R}\big)$ such that*

$$\lambda\leqslant|f''(x)|\leqslant C\lambda\qquad\text{for all } x\in[1,M].$$

Then

$$\left|\sum_{m=1}^{M}\mathrm{e}(f(m))\right|\leqslant 3CM\lambda^{1/2}+6\lambda^{-1/2}.$$

We should now explain why we focus on the case $f\in\mathcal{C}^2$. The main term in the upper bound of Theorem 2 is the term $M\lambda^{1/2}$ which provides a saving $\lambda^{1/2}$ with respect to the trivial bound M (at least when λ is small). More generally, van der Corput's method asserts that for $k\geqslant 2$, when $f\in\mathcal{C}^k$ on $[1,M]$ and $|f^{(k)}|$ has the order of magnitude $\lambda_k>0$ small, the saving is $\lambda_k^{\theta_k}$ with $\theta_k=1/(2^k-2)$ (see Theorem 3 of [31]). In the applications we have in mind, $|f^{(k)}|$ has a size close to M^{1-k}, which provides a saving $M^{(1-k)\theta_k}$. It is now plain that this saving is maximal for $k=2$.

Now we can prove the main result of this section: we give an explicit bound which controls the difference between $\mathcal{R}(f;\delta)$ and the expected main term $2M\delta$. In Section 4, we shall use it, jointly with Lemma 4, or Corollary 1 to legitimate the heuristic probabilistic estimate approach from Subsection 1.5: $\mathcal{R}(f;\delta)$ will count the number of bad cases and the main term $2M\delta$ will play the role of the probabilistic estimate. Note that we may obtain simpler expressions for the upper bounds in (11) and (12). However, this option has been ruled out in order to keep the best estimate possible. Corollary 1 actually gives more ready-to-use bounds.

Theorem 3. *Let $M\in\mathbb{N}\setminus\{0\}$, $\lambda\in\mathbb{R}_{>0}$ and $C\in[1,+\infty)$. Let $f\in\mathcal{C}^2\big([1,M],\mathbb{R}\big)$ such that*

$$\lambda\leqslant|f''(x)|\leqslant C\lambda\qquad\text{for all } x\in[1,M].$$

Then for any $\delta>0$, one has

$$|\mathcal{R}(f;\delta)-2M\delta|\leqslant\Delta_1(M,\lambda)$$
$$+\left(\frac{60}{(2\pi)^{1/2}}+24(2\pi)^{1/2}\delta\right)\left(\frac{\delta}{\lambda}\right)^{1/2}+\mathcal{R}(f+\delta;0)\quad(11)$$

and

$$|\mathcal{R}(f;\delta)-2M\delta|\leqslant\Delta_2(M,\lambda)$$
$$+\left(\frac{60}{(2\pi)^{1/2}}+24(2\pi)^{1/2}\delta\right)\left(\frac{\delta}{\lambda}\right)^{1/2}\quad(12)$$

where we have set

$$\alpha_0=\frac{6}{\pi}+\frac{4}{5},\quad\beta_0=\frac{3}{\pi}+\frac{4}{5},$$

$$\Delta_1(M,\lambda,H)=\frac{M}{H}+2\alpha_0CM\lambda^{1/2}H^{1/2}$$
$$+6CM\frac{\lambda^{1/2}}{H^{1/2}}+16\frac{\lambda^{-1/2}}{H^{1/2}},$$

$$\Delta_2(M,\lambda,H)=\frac{2M}{H}+4\beta_0CM\lambda^{1/2}H^{1/2}$$
$$+12CM\frac{\lambda^{1/2}}{H^{1/2}}+32\frac{\lambda^{-1/2}}{H^{1/2}},$$

$$\Delta_1(M,\lambda)=\min_{H\in\mathbb{N}}\Delta_1(M,\lambda,H),$$

$$\Delta_2(M,\lambda)=\min_{H\in\mathbb{N}}\Delta_2(M,\lambda,H).$$

Proof: Using Lemma 2, we have, for any $H\in\mathbb{N}$

$$|\mathcal{R}(f;\delta)-2M\delta|\leqslant E_1(H)+E_2(H)+\mathcal{R}(f+\delta;0)$$

and

$$|\mathcal{R}(f;\delta)-2M\delta|\leqslant E_1(H)+2E_2(H),$$

where $E_1(H)$ and $E_2(H)$ are defined by (5) and (6) respectively.

Our aim is now to bound each of the terms E_j using Theorem 2 for the inner sums, and Lemma 3 for the sums involved.

For E_1, Theorem 2 yields

$$E_1(H)\leqslant 2\sum_{h=1}^{H}\frac{|\sin(2\pi\delta h)|}{\pi h}\left(3CM\lambda^{1/2}h^{1/2}+6\lambda^{-1/2}h^{-1/2}\right),$$

and we have

$$2\sum_{h=1}^{H}\frac{|\sin(2\pi\delta h)|}{\pi h^{1/2}}\leqslant\frac{2}{\pi}\sum_{h=1}^{H}\frac{1}{h^{1/2}}\leqslant\frac{4}{\pi}H^{1/2},$$
$$\sum_{h=1}^{H}\frac{|\sin(2\pi\delta h)|}{\pi h^{3/2}}\leqslant\left(\frac{5}{(2\pi)^{1/2}}+2(2\pi)^{1/2}\delta\right)\delta^{1/2}$$

by using (7) and (10) respectively. This gives

$$E_1(H) \leqslant F_1(H) + \left(\frac{60}{(2\pi)^{1/2}} + 24(2\pi)^{1/2}\delta\right)\left(\frac{\delta}{\lambda}\right)^{1/2}$$

where we have set

$$F_1(H) = \frac{12}{\pi} C M \lambda^{1/2} H^{1/2}.$$

Similarly, for $E_2(H)$, Theorem 2 gives

$$E_2(H) \leqslant \frac{M}{H} + \frac{2}{H}\sum_{h=1}^{H-1}\left(1-\frac{|h|}{H}\right)\left(3CM(h\lambda)^{1/2} + 6(h\lambda)^{-1/2}\right),$$

and using (8) and (9) we have

$$E_2(H) \leqslant F_2(H)$$

where we have set

$$F_2(H) = \frac{M}{H} + \frac{8C}{5} M\lambda^{1/2}H^{1/2} + 6CM\frac{\lambda^{1/2}}{H^{1/2}} + 16\frac{\lambda^{-1/2}}{H^{1/2}}.$$

Gathering the contribution of each $E_j(H)$, we have the bounds

$$|\mathcal{R}(f;\delta) - 2M\delta| \leqslant F_1(H) + F_2(H) + \mathcal{R}(f+\delta;0) + \left(\frac{60}{(2\pi)^{1/2}} + 24(2\pi)^{1/2}\delta\right)\left(\frac{\delta}{\lambda}\right)^{1/2}$$

and

$$|\mathcal{R}(f;\delta) - 2M\delta| \leqslant F_1(H) + 2F_2(H) + \left(\frac{60}{(2\pi)^{1/2}} + 24(2\pi)^{1/2}\delta\right)\left(\frac{\delta}{\lambda}\right)^{1/2}$$

valid for any $H \in \mathbb{N}$. The expected result follows from taking the minimum in H in both inequalities. □

The form of the statement of Theorem 3 allows us to choose, for a given function, a value of H that yields small values (and if possible, the smallest ones) of the upper bounds. But, it can prove convenient to give our reader some closed and ready-to-use expressions for the upper bounds for the error $|\mathcal{R}(f;\delta) - 2M\delta|$. This is the purpose of Lemma 4 and Corollary 1.

Lemma 4. *Let $M \in \mathbb{N}\setminus\{0\}$, $\lambda \in \mathbb{R}_{>0}$ and $C \in [1,+\infty)$. Then with the notation of Theorem 3, we have*

$$\Delta_1(M,\lambda) \leqslant \alpha_0^{2/3}C^{2/3}M\lambda^{1/3} + 2\alpha_0^{2/3}C^{2/3}\max(M\lambda^{1/3}, M^{1/2}) + 16\alpha_0^{1/3}C^{1/3}M^{1/2} + (6+2\alpha_0)\alpha_0^{1/3}C^{4/3}M\lambda^{2/3}$$

and

$$\Delta_2(M,\lambda) \leqslant 2\beta_0^{2/3}C^{2/3}M\lambda^{1/3} + 4\beta_0^{2/3}C^{2/3}\max(M\lambda^{1/3}, M^{1/2}) + 32\beta_0^{1/3}C^{1/3}M^{1/2} + (12+4\beta_0)\beta_0^{1/3}C^{4/3}M\lambda^{2/3}.$$

Proof: We treat the cases $M\lambda^{2/3} \geqslant 1$ and $M\lambda^{2/3} < 1$ separately. In order to bound some of the terms below, we shall use the following inequality

$$\sqrt{1+\lfloor t\rfloor} \leqslant \sqrt{t} + \frac{1}{\sqrt{t}} \qquad (t>0). \tag{13}$$

With the definition of α_0, we recall that Theorem 3 gives

$$\Delta_1(M,\lambda) \leqslant \frac{M}{H} + 2\alpha_0 CM\lambda^{1/2}H^{1/2} + 6C\frac{M\lambda^{1/2}}{H^{1/2}} + 16\frac{\lambda^{-1/2}}{H^{1/2}}$$

for any $H \in \mathbb{N}$.

If $M\lambda^{2/3} \geqslant 1$, we set $x := (\alpha_0 C)^{-2/3}\lambda^{-1/3}$ and we choose $H := 1 + \lfloor x\rfloor$. In particular, $H \geqslant x$, and (13) implies $H^{1/2} \leqslant x^{1/2} + x^{-1/2}$, so that

$$\begin{aligned}\Delta_1(M,\lambda) &\leqslant \frac{M}{x} + 2\alpha_0 CM\lambda^{1/2}x^{1/2} + (6+2\alpha_0)C\frac{M\lambda^{1/2}}{x^{1/2}} + 16\frac{\lambda^{-1/2}}{x^{1/2}}\\ &\leqslant 3\alpha_0^{2/3}C^{2/3}M\lambda^{1/3} + (6+2\alpha_0)\alpha_0^{1/3}C^{4/3}M\lambda^{2/3} + 16\alpha_0^{1/3}C^{1/3}\lambda^{-1/3}\end{aligned}$$

which implies the expected result since

$$3\alpha_0^{2/3}C^{2/3}M\lambda^{1/3} \leqslant \alpha_0^{2/3}C^{2/3}M\lambda^{1/3} + 2\alpha_0^{2/3}C^{2/3}\max(M\lambda^{1/3}, M^{1/2})$$

and

$$16\alpha_0^{1/3}C^{1/3}\lambda^{-1/3} \leqslant 16\alpha_0^{1/3}C^{1/3}M^{1/2}.$$

If $M\lambda^{2/3} \leqslant 1$, we set $x := (\alpha_0 C)^{-2/3}(M\lambda)^{-1}$ and we choose $H := 1 + \lfloor x\rfloor$. In particular, $H \geqslant x$, and (13) implies $H^{1/2} \leqslant x^{1/2} + x^{-1/2}$, so that

$$\begin{aligned}\Delta_1(M,\lambda) &\leqslant \frac{M}{x} + 2\alpha_0 CM\lambda^{1/2}x^{1/2} + (6+2\alpha_0)C\frac{M\lambda^{1/2}}{x^{1/2}} + 16\frac{\lambda^{-1/2}}{x^{1/2}}\\ &\leqslant \alpha_0^{2/3}C^{2/3}M^2\lambda + 2\alpha_0^{2/3}C^{2/3}M^{1/2} + (6+2\alpha_0)\alpha_0^{1/3}C^{4/3}M^{3/2}\lambda + 16\alpha_0^{1/3}C^{1/3}M^{1/2},\end{aligned}$$

which implies the expected result since

$$\alpha_0^{2/3}C^{2/3}M^2\lambda \leqslant \alpha_0^{2/3}C^{2/3}M\lambda^{1/3},$$

$$2\alpha_0^{2/3}C^{2/3}M^{1/2} \leqslant 2\alpha_0^{2/3}C^{2/3}\max(M\lambda^{1/3}, M^{1/2})$$

and

$$(6+2\alpha_0)\alpha_0^{1/3}C^{4/3}M^{3/2}\lambda \leqslant (6+2\alpha_0)\alpha_0^{1/3}C^{4/3}M\lambda^{2/3}.$$

For the estimates for $\Delta_2(M,\lambda)$, we notice that Theorem 3 gives

$$\frac{\Delta_2(M,\lambda)}{2} \leqslant \frac{M}{H} + 2\beta_0 CM\lambda^{1/2}H^{1/2} + 6C\frac{M\lambda^{1/2}}{H^{1/2}} + 16\frac{\lambda^{-1/2}}{H^{1/2}}$$

for any $H \in \mathbb{N}$. Reproducing the previous computations with β_0 instead of α_0 gives the expected result.

This completes the proof. □

Combining Theorem 3 and Lemma 4, we obtain the following:

Corollary 1. *We have, under the assumptions of Theorem 3, the two estimates*

$$\begin{aligned}|\mathcal{R}(f;\delta) - 2M\delta| &\leqslant \alpha_0^{2/3}C^{2/3}M\lambda^{1/3} \\ &\quad + 2\alpha_0^{2/3}C^{2/3}\max(M\lambda^{1/3}, M^{1/2}) \\ &\quad + 16\alpha_0^{2/3}C^{2/3}M^{1/2} + (6+2\alpha_0)\alpha_0^{1/3}C^{4/3}M\lambda^{2/3} \\ &\quad + \left(\frac{60}{(2\pi)^{1/2}} + 24(2\pi)^{1/2}\delta\right)\left(\frac{\delta}{\lambda}\right)^{1/2} + \mathcal{R}(f+\delta;0) \quad (14)\end{aligned}$$

and

$$\begin{aligned}|\mathcal{R}(f;\delta) - 2M\delta| &\leqslant 2\beta_0^{2/3}C^{2/3}M\lambda^{1/3} \\ &\quad + 4\beta_0^{2/3}C^{2/3}\max(M\lambda^{1/3}, M^{1/2}) \\ &\quad + 32\beta_0^{2/3}C^{2/3}M^{1/2} + (12+4\beta_0)\beta_0^{1/3}C^{4/3}M\lambda^{2/3} \\ &\quad + \left(\frac{60}{(2\pi)^{1/2}} + 24(2\pi)^{1/2}\delta\right)\left(\frac{\delta}{\lambda}\right)^{1/2} \quad (15)\end{aligned}$$

for any $\delta > 0$.

As a final remark, let us point that the term $(\delta/\lambda)^{1/2}$, which may look somewhat singular in our bounds, has a very natural interpretation. Indeed, if f is a polynomial of the form $f(x) = a + \frac{u}{v}x + \frac{\lambda}{2}x^2$, with $a, u, v \in \mathbb{Z}$, v small, then for any $x \in (-(2\delta/\lambda)^{1/2}, (2\delta/\lambda)^{1/2})$ divisible by v, we shall have $\|f(x)\| < \delta$, which gives $\geqslant \frac{2}{v}\left(\frac{2\delta}{\lambda}\right)^{1/2}$ bad cases in this interval.

More generally, the same would hold for a function with $f(0) \in \mathbb{Z}$, $f'(0) = u/v$, and with derivatives of order $\geqslant 3$ vanishing quickly enough so that f is well approximated by a polynomial of degree 2 over $(-(2\delta/\lambda)^{1/2}, (2\delta/\lambda)^{1/2})$. In terms of bad cases, this is, e.g., the well-known situation of the cosine function close to 0, or of the cube root close to 1.

4 An estimate of the number of "bad cases" for the first quick step of the two step Ziv strategy

Let $p \geqslant 1$, $e_1, e_2 \in \mathbb{Z}$, $a, b \in \mathcal{D}_p$ two floating-point numbers such that $[a,b] \subset [2^{e_1}, 2^{e_1+1})$, $\delta > 0$ and $\varphi : [a,b] \to \mathbb{R}$ be a $\mathcal{C}^\ell$ function, $\ell \geqslant 2$, such that $\varphi([a,b]) \subset [2^{e_2}, 2^{e_2+1})$[8]. Let $A = a2^{p-1-e_1}$ and $B = b2^{p-1-e_1} \in \mathbb{N}$, and finally $M = B - A + 1$. In order to tackle Problems 1 and 2 of Section 2, we introduce $\mathcal{I}(\varphi; p, A, B, \delta) :=$

- $$\#\left\{A \leqslant j \leqslant B,\ \left\|2^{p-1-e_2}\varphi\left(\frac{j}{2^{p-1-e_1}}\right) - \frac{1}{2}\right\| < \delta\right\} \quad (16)$$
  in **rounding to nearest mode**,
- $$\#\left\{A \leqslant j \leqslant B,\ \left\|2^{p-1-e_2}\varphi\left(\frac{j}{2^{p-1-e_1}}\right)\right\| < \delta\right\} \quad (17)$$
  in **directed rounding modes**.

We call (A, B, δ)-bad cases (or simply bad cases when there is no ambiguity) the elements of these sets.

Our goal is to obtain fully effective estimates of the form $\mathcal{I}(\varphi; p, A, B, \delta) = 2M\delta + o(M\delta)$, or, at least, $\mathcal{I}(\varphi; p, A, B, \delta) = 2M\delta + O(M\delta)$. Though less accurate, the latter may still yield a fairly good upper bound for the number of bad cases. Recall that if $\delta = 2^{-k}$, $k \in \mathbb{N}$, $2M\delta$ is the number of bad cases suggested by the heuristic approach of Subsection 1.5, so that $\mathcal{I}(\varphi; p, A, B, 2^{-k})$ is exactly the number of bad cases for the first quick step of the two step Ziv strategy using intermediate accuracy 2^{1-p-k}. In the sequel, we shall thus use $\delta = 2^{-k}$ for some $k \in \mathbb{N}$ (the number of extra bits in the first step of Ziv strategy).

Remark 2. *In the context of a two step Ziv strategy, a good estimate on $\mathcal{I}(\varphi; p, A, B, \delta)$ allows one to give an average analysis of the cost of an evaluation of $\varphi(x)$ for $x \in [a,b]$.*

Indeed, if the first step uses relative accuracy 2^{1-p-k}, the second using relative accuracy $2^{1-p-k'}$ (with k' large enough so as to capture worst cases), the average cost of one evaluation of the function is $c_{\varphi,p}(k) + c_{\varphi,p}(k')\mathcal{I}(\varphi; p, A, B, 2^{-k})/M$, where $c_{\varphi,p}(j)$ stands for the cost of one evaluation of f with relative accuracy 2^{1-p-j}. For choices of parameters for which our estimates yield $\mathcal{I}(\varphi; p, A, B, 2^{-k}) \approx 2^{1-k}M$, we obtain an average cost $\approx c_{\varphi,p}(k) + 2^{1-k}c_{\varphi,p}(k')$ (the $\approx$ meaning here that we can determine a rigorous, small interval centered at $c_{\varphi,p}(k) + 2^{1-k}c_{\varphi,p}(k')$ containing the average cost).

In order to connect these questions and the results obtained on (1),

- In **rounding to nearest mode**, we introduce the function
  $$f(x) := 2^{p-1-e_2}\varphi\left(\frac{A-1+x}{2^{p-1-e_1}}\right) - \frac{1}{2}, \qquad 1 \leqslant x \leqslant M, \quad (18)$$
  and observe that $\mathcal{R}(f;\delta) = \mathcal{I}(\varphi; p, A, B, \delta)$.
- Similarly, in **directed rounding modes**, we introduce the function
  $$f(x) := 2^{p-1-e_2}\varphi\left(\frac{A-1+x}{2^{p-1-e_1}}\right), \qquad 1 \leqslant x \leqslant M, \quad (19)$$
  and observe again that $\mathcal{R}(f;\delta) = \mathcal{I}(\varphi; p, A, B, \delta)$.

We are thus in a position to apply Theorem 3 to derive rigorous and useful estimates for $\mathcal{I}(\varphi; p, A, B, \delta)$. Note that, whatever the rounding mode we choose, only the values of the second derivative of f matter in Theorem 3, hence the estimates that we obtain will be the same.

We now give the precise values of the constants involved in the statement of Theorem 3, namely C and λ.

$$\begin{aligned}\lambda &= \min_{x\in[1,M]} |f''(x)| \\ &= 2^{-p+1+2e_1-e_2} \min_{x\in[a,b]} |\varphi''(x)|\,.\end{aligned}$$

Recall that λ has to be positive. Now,

$$\begin{aligned}C\lambda &= \max_{x\in[1,M]} |f''(x)| \\ &= 2^{-p+1+2e_1-e_2} \max_{x\in[a,b]} |\varphi''(x)|\,,\end{aligned}$$

hence

$$C = \frac{\max_{x\in[1,M]} |f''(x)|}{\min_{x\in[1,M]} |f''(x)|} = \frac{\max_{x\in[a,b]} |\varphi''(x)|}{\min_{x\in[a,b]} |\varphi''(x)|}.$$

Note that this formula is legitimate since we assumed $\min_{x\in[a,b]} |\varphi''(x)| = 2^{p-1-2e_1+e_2}\lambda > 0$.

8. As already said at the beginning of Section 2, considering this case is sufficient to address the general case $[a,b] \subset \pm[2^{e_1}, 2^{e_1+1})$ and $\varphi([a,b]) \subset \pm[2^{e_2}, 2^{e_2+1})$.

Remark 3. *In order to have $\lambda > 0$, one might have to split the interval into sub-intervals over which φ'' has no zero. More generally, if φ'' has huge variations, one might consider splitting the interval to avoid too large a value of C.*

Before applying Theorem 3 in order to estimate the number of bad cases of the first quick step, we now wish to understand the constraints on the parameters M, k, λ, C, and in particular, the number of bits k for which we can derive a relevant result from Theorem 3. Therefore, we check the various constraints that the parameters must satisfy in order to make inequality (15) (from Corollary 1) relevant. This inequality is a consequence of Theorem 3 but, in our practical experiments in binary64 and binary128, the results it yields are very close to the one provided by Theorem (3).

4.1 Analysis of the results established in Section 3: constraints on the parameters

The main term $2M\delta$ should not be smaller than any of the terms of the right-hand side of (15). So let us check the following conditions:

- we have $2M\delta \leqslant 4\beta_0^{2/3}C^{2/3}M\lambda^{1/3}$ if and only if
$$\frac{\log_2(1/\lambda)}{3} - \frac{2\log_2(\beta_0 C)}{3} - 1 \leqslant \log_2(1/\delta)$$
i.e.
$$p - 2e_1 + e_2 - \log_2\left(\min_{x\in[a,b]} |\varphi''(x)|\right) - 2\log_2(\beta_0 C) - 4 \leqslant 3k.$$
- We have $2M\delta \leqslant 32\beta_0^{2/3}C^{2/3}M^{1/2}$ if and only if
$$\frac{\log_2(M)}{2} \leqslant k + 4 + \frac{2\log_2(\beta_0 C)}{3}.$$
- We have $2M\delta \leqslant (12 + 4\beta_0)\beta_0^{1/3}C^{4/3}M\lambda^{2/3}$ if and only if
$$\frac{2\log_2(1/\lambda)}{3} - \log_2(3+\beta_0) - \frac{\log_2(\beta_0)}{3} - \frac{4\log_2(C)}{3} - 1 \leqslant \log_2(1/\delta)$$
i.e.
$$2(p-2e_1+e_2) - 2\log_2\left(\min_{x\in[a,b]} |\varphi''(x)|\right) - 3\log_2(3+\beta_0) - \log_2(\beta_0) - 4\log_2(C) - 5 \leqslant 3k.$$
- Now, we check the condition
$$2M\delta \leqslant \left(\frac{60}{(2\pi)^{1/2}} + 24(2\pi)^{1/2}\delta\right)\left(\frac{\delta}{\lambda}\right)^{1/2}.$$
Note that, as soon as $k \geqslant 2$, $60/(2\pi)^{1/2} \geqslant 24(2\pi)^{1/2}\delta$. Therefore, we focus on the condition $2M\delta \leqslant (60/(2\pi)^{1/2})\left(\frac{\delta}{\lambda}\right)^{1/2}$, which is equivalent to
$$2\log_2(M) - \log_2(1/\lambda) - 1 - 2\log_2(15) + \log_2(\pi) \leqslant k$$
i.e.
$$2\log_2(M) - p + 2e_1 - e_2 + \log_2\left(\min_{x\in[a,b]} |\varphi''(x)|\right) - 2\log_2(15) + \log_2(\pi) \leqslant k.$$

Remark 4. *It is probably worth commenting a little bit informally on the error term. Typically, for values of x such that x and $f(x)$ have exponent close to 0, in our context, λ is going to be of the order of 2^{-p}; assuming for the sake of simplicity that all constants including C are 1, the upper bound is of the order of $M2^{-p/3} + M^{1/2} + 2^{(p-k)/2}$ (recall that M is the number of floating-point numbers in the interval under consideration, so that the trivial upper bound would be M). As we can restrict to $k \leqslant p/3$, the middle term is always absorbed by one of the other two. The error term is thus either roughly $M2^{-p/3}$ for $M \geqslant 2^{2p/3}$ ("large intervals") or $2^{(p-k)/2}$ for $M < 2^{2p/3}$ ("small intervals").*

Now, we illustrate our results on two examples: the cube root and the exponential functions. In both cases, we will first have to decompose the input binade in order for the output values to fit in one single binade, then find explicit expressions for C and λ before being able to conduct actual experiments.

4.2 The $\sqrt[3]{\cdot}$ function

In this case, we have $\varphi(x) = x^{1/3}$. This function never overflows nor underflows in its exponent range. Note that it takes many exact values: for instance, if $x \in \mathcal{D}_{\lfloor p/3 \rfloor}$, $y = x^3 \in \mathcal{D}_p$ has x as an exact cube root in $\mathcal{D}_p$.

4.2.1 Explicit values of the parameters

Let us apply our method on an interval $[a, b]$ enclosed in a single binade of exponent e_1. Note that we can assume $a, b \geqslant 0$ as the cube root function is odd; further, as $\sqrt[3]{2^3 \cdot x} = 2\sqrt[3]{x}$, we can restrict our study to any 3 consecutive binades, for instance $[1/2, 1), [1, 2), [2, 4)$.

We have
$$\begin{aligned}\varphi([a,b]) \cap \mathcal{D}_p &= [a^{1/3}, b^{1/3}] \cap \mathcal{D}_p \\ &\subset [2^{e_1/3}, 2^{(e_1+1)/3}) \cap \mathcal{D}_p \\ &\subset [2^{\lfloor e_1/3 \rfloor}, 2^{\lfloor e_1/3 \rfloor + 1}).\end{aligned}$$

As such, we have $e_2 = \lfloor e_1/3 \rfloor$. Therefore, and as $a, b > 0$, we have
$$\lambda = \frac{2^{-p+2+2e_1-\lfloor e_1/3 \rfloor}}{9b^{5/3}},$$
$$C = \frac{\max_{x\in[a,b]} |\varphi''(x)|}{\min_{x\in[a,b]} |\varphi''(x)|} = (b/a)^{5/3}.$$

Overall, we can always treat a full input binade at a time, so that $M = 2^{p-1}$.

4.2.2 A simplified statement

We can now collect our results to illustrate the use of Lemma 4 to obtain a simple bound on the number of bad cases, depending only on the natural parameters p, δ of the problem. We actually give two bounds; the second one is more compact but loses a small constant factor as p grows large.

Corollary 2. *For the cube root function over the binade $[1/2, 1)$, if $p \geqslant 11$ the number of bad cases is in the interval $[\delta 2^p - \rho, \delta 2^p + \rho]$, where*
$$\rho \leqslant 5.72 \cdot 2^{2p/3} + 40.2 \cdot 2^{p/2} + 19.7 \cdot 2^{p/3} + 110 \cdot \delta^{1/2} 2^{p/2}.$$
If, further, $p \geqslant 24$, we have $\rho \leqslant 8.31 \cdot 2^{2p/3} + 110 \cdot \delta^{1/2} 2^{p/2}$.

Proof: Follows from the second part of Lemma 4 by specializing the values of C, λ as seen before. Note that we have $\lambda = 2^{1-p}/9(1-2^{-p})^{5/3}$, which is $\geqslant 2^{1-p}/9$ and $\leqslant 1.001 \cdot 2^{1-p}/9$ for $p \geqslant 11$.

The second claim simply uses $2^{p/2} \leqslant 2^{2p/3}2^{-4}$ and $2^{p/3} \leqslant 2^{2p/3}2^{-8}$ for $p \geqslant 24$. □

4.2.3 Experimental results

We shall treat the binary64 (Table 1) and binary128 (Table 2) cases. We chose not to deal with the binary32 case as it is so easy, in that case, to compute bad cases exactly that our work seemed irrelevant. On the other hand, for binary64 and binary128, it is not realistic to do the computation of the exact values for a whole binade; we thus compare our bounds with the expected value under the heuristic probabilistic model. We shall also limit ourselves to the binade $[1/2, 1)$, the results for the other two binades being slightly different, but very similar.

Our experiments are based on Theorem 3, but we checked that Lemma 4 gives the exact same tables (given the number of decimal figures displayed in the tables) in that setting.

The second column (*expected*) gives the expected number of bad cases under the probabilistic heuristic of Subsection 1.5, that is to say $2M\delta = M2^{-k+1}$.

The last column deserves further explanation: rather than giving the exact lower and upper bound, we prefer giving their "relative quality", which means the bound for $|\mathcal{R}(f;\delta)/2M\delta - 1|$. Thus, a bound close to 0% means that our bounds give a very precise order of magnitude, while a bound larger than 100% is not so good – though, while the corresponding lower bound is meaningless, the upper bound usually remains non trivial, even often useful as it still gives the correct order of magnitude. For instance, Table 1 tells us that, in binary64, if $k = 14$, the number of bad cases for $\sqrt[3]{\cdot}$ over $[1/2, 1)$ lies between $(1-0.46)\cdot 2^{42}$ and $(1+0.46)\cdot 2^{42}$, which leads to a probability between $(1-0.46)\cdot 2^{-10}$ and $(1+0.46)\cdot 2^{-10}$ for the occurrence of the second accurate step.

k	expected	rel. radius
5	281474976710656	0.1%
6	140737488355328	0.18%
7	70368744177664	0.36%
8	35184372088832	0.72%
9	17592186044416	1.5%
10	8796093022208	2.9%
11	4398046511104	5.8%
12	2199023255552	11.5%
13	1099511627776	23%
14	549755813888	46%
15	274877906944	92%

TABLE 1
Our bounds for the cube root function in the binary64 case, $x \in [1/2, 1)$

4.3 The exp function

In this case, we have $\varphi(x) = \exp(x)$. Note that in this section, we shall treat the exponential function both for positive and negative values of x. Equivalently, this can be seen as treating the two functions exp and $x \mapsto \exp(-x)$ for positive arguments, which is consistent with the formalization adopted in Section 2.

k	expected	rel. radius
8	4.0564819e31	6.8e-7 %
11	5.0706024e30	5.4 e-6 %
14	6.3382530e29	4.3 e-5%
17	7.9228162e28	3.5 e-4%
20	9.9035203e27	2.8 e-3%
23	1.2379400e27	0.022%
26	1.5474250e26	0.18%
29	1.9342813e25	1.5%
32	2.4178516e24	11.2%
35	3.0223145e23	90%

TABLE 2
Our bounds for the cube root function in the binary128 case, $x \in [1/2, 1)$

The exp function overflows for large, positive x and underflows for large, negative x; on the other hand, the only rational value of $\exp(x)$, with x a rational number, is $\exp(0) = 1$, so that for the rational δ we use we shall ignore the term $\mathcal{R}(f+\delta;0)$ in (3), (11) and (14). We shall use that repeatedly in this section, together with the fact that $\log 2$ is irrational so that $x/\log 2$ and $x \log 2$ can never be an integer when x is a nonzero rational number.

4.3.1 Explicit values of the parameters

Let $[a, b]$ be an interval such that $\pm[a, b] \subset [2^{e_1}, 2^{e_1+1})$ and $\exp([a, b]) \subset [2^{e_2}, 2^{e_2+1})$. First observe that obviously,

$$\lambda = 2^{-p+1+2e_1-e_2}\exp(a),$$
$$C = \exp(b-a).$$

We now discuss e_1, e_2. For $\exp([a, b])$ to be enclosed in a single binade of exponent e_2, we need to have $2^{e_2} \leqslant \exp(a) \leqslant \exp(b) < 2^{e_2+1}$, hence $[a, b] \subset [e_2 \log 2, (e_2+1)\log 2)$. For positive real numbers with $e_1 \leqslant -2$, we have $[2^{e_1}, 2^{e_1+1}) \subset [0, \log 2)$, hence we may deal with a whole binade at once and have $e_2 = 0$. The same is true for negative real numbers with $e_1 \leqslant -2$, where we can take $e_2 = -1$. In both cases, we can take $M = 2^{p-1}$.

For real numbers with $e_1 \geqslant -1$, however, we have to split a given input binade into parts:

$$\begin{aligned}[2^{e_1}, 2^{e_1+1}) = {} & [2^{e_1}, \lceil 2^{e_1}/\log 2\rceil \log 2) \\ & \cup \bigcup_{\lceil \frac{2^{e_1}}{\log 2}\rceil \leqslant \ell < \lfloor \frac{2^{e_1+1}}{\log 2}\rfloor} [\ell \log 2, (\ell+1)\log 2) \\ & \cup \left[\lfloor 2^{e_1+1}/\log 2\rfloor \log 2, 2^{e_1+1}\right) \\ =: {} & I_0(e_1) \cup \bigcup_{\ell} I_\ell(e_1) \cup I_\infty(e_1).\end{aligned}$$

Over $I_0(e_1)$, we have $e_2 = \lfloor 2^{e_1}/\log 2\rfloor$, over $I_\infty(e_1)$ we have $e_2 = \lfloor 2^{e_1+1}/\log 2\rfloor$ while over $I_\ell(e_1)$ the corresponding exponent is ℓ. In this case, we are thus limited to values of M of the order of $2^{p-1-e_1}\log 2$, or even smaller for I_0 and I_∞ if 2^{e_1} is close to a multiple of $\log 2$.

A symmetric discussion holds similarly for negative x, where $I_0 = [-\lceil 2^{e_1}/\log 2\rceil \log 2, -2^{e_1}]$ gives $e_2 = -\lceil 2^{e_1}/\log 2\rceil$, $I_\ell = [-(\ell+1)\log 2, -\ell \log 2)$ gives $e_2 = -1-\ell$, $I_\infty = (-2^{e_1+1}, -\lfloor 2^{e_1+1}/\log 2\rfloor \log 2)$ gives $e_2 = -\lceil 2^{e_1+1}/\log 2\rceil$.

We sum up the previous discussion concerning e_1 and e_2 in the following table.

sign(x)	e_1	I	e_2
< 0	$\leqslant -2$		-1
	$\geqslant -1$	I_0	$-\lceil 2^{e_1}/\log 2 \rceil$
		I_ℓ	$-1-\ell$
		I_∞	$-\lceil 2^{e_1+1}/\log 2 \rceil$
> 0	$\leqslant -2$		0
	$\geqslant -1$	I_0	$\lfloor 2^{e_1}/\log 2 \rfloor$
		I_ℓ	ℓ
		I_∞	$\lfloor 2^{e_1+1}/\log 2 \rfloor$

4.3.2 A simplified statement

Again, we illustrate Lemma 4 to get a global bound over the binade $[1, 2)$ of a compact form.

Corollary 3. *For the exponential function over the binade $[1, 2)$, if $p \geqslant 11$ the number of bad cases is in the interval $[\delta 2^p - \rho, \delta 2^p + \rho]$, where*

$$\rho \leqslant 5.45 \cdot 2^{2p/3} + 37.2 \cdot 2^{p/2} + 27.7 \cdot 2^{p/3} + 68.2 \cdot \delta^{1/2} 2^{p/2}.$$

If, further, $p \geqslant 24$, we have $\rho \leqslant 7.89 \cdot 2^{2p/3} + 68.2 \cdot \delta^{1/2} 2^{p/2}$.

Proof: We split $[1, 2)$ as $[1, 2\log 2) \cup [2\log 2, 2)$.

Over $[1, 2\log 2)$, we have $a = 1$ and $b \leqslant 2\log 2$, $e_1 = 0$, $e_2 = 1$; this gives $\lambda = 2^{-p}e$, $C \leqslant 4/e$, $M \leqslant 2^p(\log 2 - 1/2) + 1 \leqslant 0.194 \cdot 2^p$ for $p \geqslant 11$. The first part of Lemma 4 then yields

$$\rho_1 \leqslant 2.05 \cdot 2^{2p/3} + 11.2 \cdot 2^{p/2} + 10.1 \cdot 2^{p/3} + 31.5 \cdot \delta^{1/2} 2^{p/2}.$$

Over $[2\log 2, 2)$ we have $a \in [2\log 2, 2\log 2 + 2^{1-p}]$ and $b = 2 - 2^{1-p}$, $e_1 = 0$, $e_2 = 2$; this gives $\lambda \in [2^{-p+1}, 1.001 \cdot 2^{-p+1}]$, $C \leqslant e^2/4$, $M \leqslant 2^p(1 - \log 2)$, and then

$$\rho_2 \leqslant 3.4 \cdot 2^{2p/3} + 26 \cdot 2^{p/2} + 17.6 \cdot 2^{p/3} + 36.7 \cdot \delta^{1/2} 2^{p/2}.$$

Adding the two estimates yields the claimed result. □

4.3.3 Experimental results

We now include similar tables as in Subsection 4.2. Those tables are obtained by using the decomposition above, and applying Theorem 3 for each of those intervals – mimicking the strategy of Corollary 3 (except that the slightly sharper Theorem 3 is used instead of the ready-to-use Lemma 4).

In order to demonstrate our bounds in various settings, we treat three different binades with a rather different behaviour: $[1, 2)$, $[32, 64)$ and $[1/64, 1/32)$ in order to illustrate the behaviour of our bounds for large/small operands of the exponential function. We do not illustrate our bounds for large, negative x as in that case we get large negative e_2, which yield a value of $\lambda \geqslant 1$ and thus a bound $\geqslant M$, hence empty.

Before going into this direction, we first mention that we stress-tested our bounds with the binade $[2^{-26}, 2^{-25})$ where, depending on k and the rounding mode, either none or all the floating-point numbers give rise to a hard-to-round case; as expected in this setting, our bounds give the trivial result that the number of bad cases is in $[0, 2^{p-1}]$.

Our results are collected in Table 3 and Table 4. We obtain, for instance, in binary128, for $k = 29$, that the number of bad cases for exp over $[1, 2)$ lies between $(1 - 0.014) \cdot 2^{84}$ and $(1 + 0.014) \cdot 2^{84}$, which leads to a probability between $(1 - 0.014) \cdot 2^{-28}$ and $(1 + 0.014) \cdot 2^{-28}$ for the occurrence of the second accurate step.

k	expected	rel. rad. [1, 2)	rel. rad. [32, 64)	rel. rad. [1/64, 1/32)
5	281474976710656	0.09%	1%	0.02%
6	140737488355328	0.17%	1.9%	0.02%
7	70368744177664	0.34%	3.7%	0.03%
8	35184372088832	0.67%	7.3%	0.05%
9	17592186044416	1.3%	15%	0.09%
10	48796093022208	2.7%	29%	0.2%
11	4398046511104	5.4%	58%	0.3%
12	2199023255552	11%	116%	0.6%
13	1099511627776	22%	231%	1.1%
14	549755813888	43%	462%	2.0%
15	274877906944	86%	924%	3.9%

TABLE 3
Our bounds for the exponential function in the binary64 case

k	expected	rel. rad. [1, 2)	rel. rad. [32, 64)	rel. rad. [1/64, 1/32)
8	4.0564819e31	6.4e-7%	6.9e-6 %	2.8e-8
11	5.0706024e30	5.1e-6%	5.6e-5%	2.2e-7
14	6.3382530e29	4.1e-5%	4.5e-4%	1.8e-6
17	7.9228162e28	3.3e-4%	3.6e-3%	1.5e-5
20	9.9035203e27	2.7e-3%	2.9e-2%	1.2e-4
23	1.2379400e27	2.1e-2%	2.3e-1%	9.0e-4
26	1.5474250e26	1.7e-1%	1.9%	7.2e-3
29	1.9342813e25	1.4%	15%	5.8e-2
32	2.4178516e24	11%	116%	0.46%
35	3.0223145e23	86%	924%	3.7%

TABLE 4
Our bounds for the exponential function in the binary128 case

We conclude this discussion by a binade which is of a particular interest for the `CRlibm` library, as the latter starts by performing an argument reduction to restrict to the interval $[0, \log(2) \cdot 2^{-12})$; we thus give a few values of our bounds for the binade $[2^{-14}, 2^{-13})$ in Table 5.

k	rel. radius $[2^{-14}, 2^{-13})$
11	14%
12	19%
13	26%
14	38%
15	54%

TABLE 5
Some more `CRlibm`-specific bounds for the exponential function, binary64 format

4.4 Discussion

We now try to summarize the teachings that our experiments bring regarding our bounds.

4.4.1 The good situations

In all situations presented above, our experiments show that, as expected, our bounds provide orders of magnitude which go from very precise to reasonable when k approaches $p/3$. As expected, the relative quality of our bounds improve a lot with p for fixed k – Remark 4 states that we expect the relative quality to be of the order $2^{k-p/3}$ when input and output exponents are close to 0.

Until this paper only the probabilistic model (and no rigorous result – except for binary32 where complete enumeration is easy) was available. Note in particular that Tables 1 and 3 cover ranges for k which correspond to the values of k used commonly for the implementation of the *quick phase* of CRLibm functions – as such, demonstrate that results are sufficiently sharp to allow for a rigorous analysis of Ziv's method in that setting (see also Table 5). One might wonder also how our bounds work for MPFR [12] implementation of Ziv's strategy; MPFR code suggests that the number of guard bits, depending on the function, is usually of the order of $a \cdot \log_2(p) + b$ for $a \in \{1, 2\}$, and not-too-large b; this corresponds to $\delta \approx 2^{-b}p^{-a}$, a range where our bounds will be even better than for `CRlibm` except for very small intervals or very small/large output exponents, see below.

4.4.2 The bad situations

In this subsection we specifically target a few examples where we expect our bounds to behave poorly. In view of Remark 4 and Section 4.1, it should come as no surprise that the bounds behave poorly in the situations where

- Phenomenon 1: λ is either larger than the expected 2^{-p} (which means either large e_1 or large negative e_2)
- Phenomenon 2: or, conversely, λ much smaller than expected – in which case the term $(\delta/\lambda)^{1/2}$ may grow wild (which means large negative e_1 or large e_2).
- Phenomenon 3: Also, if M is much smaller than 2^{p-1}, the term, independent of M, $(\delta/\lambda)^{1/2}$ may become larger than the main term $2M\delta$ which has a linear dependency in M; to fix ideas, when exponents are close to zero, this is bound to happen whenever M becomes smaller than $2^{(p+k)/2}$.

Note that Phenomena 2 and 3 are related with the somewhat unavoidable term $(\delta/\lambda)^{1/2}$, see the end of Section 3.

Phenomena 1 & 2 do not occur for the cube root function, as we can always restrict to exponents in the range $[0, 2]$.

Phenomenon 1 is obvious on the case of the binade $(-64, -32]$ of the exponential function where the results are terribly bad: in binary64 with $\delta = 2^{-15}$, we get a (ridiculous) bound for the number of bad cases of the order of 2^{128}, whereas the right order of magnitude is expected to be 2^{38}.

Phenomenon 2 depends on the relative order of magnitudes of p vs. e_1, e_2 and would be somewhat visible on the binades $[1/64, 1/32)$ and $[32, 64)$ in binary32. It becomes visible, for instance, in binary64 on the binade $[2^{-25}, 2^{-24})$ where for $\delta = 2^{-5}$ the bound is already trivial – and, in order to compare with previous tables, the relative radius is already $\approx 3660\%$.

What happens on the binade $[32, 64)$ in binary64 and binary128 is of a different nature and comes mostly from the fact that we have to split the interval (i.e., essentially Phenomenon 3) in many sub-intervals over which our bound is much worse. We come back to Phenomenon 2 in the final remark below.

Note that Phenomenon 1 corresponds to large (in absolute value) x or zeros of φ, and case Phenomenon 2 to small x (again in absolute value) or x close to a singularity of φ. Note also that if e_1 and e_2 are simultaneously large or small with $e_2 \approx 2e_1$, neither Phenomenon 1 nor 2 occurs; this may happen if x is a zero of order 2 of φ, or if $\varphi(x)$ tends to infinity as x^2.

It should probably be pointed that in the bad cases problem and Ziv's strategy, for most elementary functions the situations of very large/small input/output exponents are handled in a specific way, and we feel that those bad situations have thus little importance on the relevance of our results.

4.4.3 One further comment

Often in our tables, the relative radius seems to roughly get multiplied by 2 when k increases by 1. It comes from the fact that in practice the $M\lambda^{1/3}$ term of the error bound often seems to dominate over the term $(\delta/\lambda)^{1/2}$ even for moderately small values of δ. As the former term is independent of δ, this means that the error bound is close to constant when δ varies, explaining why the percentages displayed double when k increases by one – the error term does not change a lot while the main term decreases by a factor of 2.

Note that when Phenomenon 2 occurs (i.e. $(\delta/\lambda)^{1/2}$ dominates) this no longer happens and a similar analysis shows that the relative radius then increases by a factor of $\sqrt{2}$ when k increases by 1.

5 Conclusion

In this paper, we gave the first rigorous version of a heuristic commonly used by the designers of mathematical libraries. To do so, we established some theoretical statements, based on so-called exponential sums techniques. Our results apply to any $\mathcal{C}^2$ function and, though we only dealt with the radix 2 case, our results would remain valid for any other radix.

Our experiments demonstrate that, even though our bounds have their weaknesses, they provide sharp results for the analysis of the first phase of Ziv's strategy. They legitimate the two Ziv step strategy used for instance by the designers of the `CRlibm` library: it makes it possible to evaluate some elementary functions with correct rounding while keeping an average performance and a memory overhead due to correct rounding negligible [9], [10].

The current main limit in our work is the fact that, if p denotes the precision of the floating-point environment that we use, we can roughly estimate the numbers of bad cases up to $k \approx p/3$ bits but not beyond. Note that the nature of our techniques makes doubtful the fact that we shall ever be able to tackle the "worst cases" (which would correspond to $k \approx p$).

There are two reasons for the current $k \approx p/3$ limit: we lose the parameter 2^{-k} in some terms of the result stated in Lemma 2; and the use of Theorem 2. However, the latter has been shown to be optimal for $\mathcal{C}^2$ functions [31].

In the future, we shall focus on tackling some improvements of these two results, for instance in the more restrictive, and yet quite relevant and useful, framework of elementary and special functions. We shall also try to overcome to some extent some of the phenomena pointed in Subsection 4.4.2, in order for instance to be able to cope with large arguments of trigonometric functions.

ACKNOWLEDGEMENTS

The authors wish to warmly thank Christoph Lauter and Jean-Michel Muller for their very helpful answers to our various questions. Also, the careful rereading of the anonymous reviewers was a real help in improving the paper.

REFERENCES

[1] American National Standards Institute and Institute of Electrical and Electronic Engineers. *IEEE Standard for Binary Floating-Point Arithmetic*. ANSI/IEEE Standard 754–1985, 1985.

[2] American National Standards Institute and Institute of Electrical and Electronic Engineers. *IEEE Standard for Radix Independent Floating-Point Arithmetic*. ANSI/IEEE Standard 854–1987, 1987.

[3] E. Bombieri and H. Iwaniec. On the order of $\zeta(\frac{1}{2}+it)$. *Ann. Scuola Norm. Sup. Pisa Cl. Sci. (4)*, 13(3):449–472, 1986.

[4] E. Bombieri and H. Iwaniec. Some mean-value theorems for exponential sums. *Ann. Scuola Norm. Sup. Pisa Cl. Sci. (4)*, 13(3):473–486, 1986.

[5] J. Bourgain. Decoupling, exponential sums and the Riemann zeta function. *J. Amer. Math. Soc.*, 30(1):205–224, 2017.

[6] N. Brisebarre and J.-M. Muller. Correct rounding of algebraic functions. *RAIRO - Theoretical Informatics and Applications*, 41(1):71–83, Apr. 2007.

[7] W. J. Cody. A proposed radix and word length independent standard for floating-point arithmetic. *ACM SIGNUM Newsletter*, 20:37–51, Jan. 1985.

[8] W. J. Cody, J. T. Coonen, D. M. Gay, K. Hanson, D. Hough, W. Kahan, R. Karpinski, J. Palmer, F. N. Ris, and D. Stevenson. A proposed radix-and-word-length-independent standard for floating-point arithmetic. *IEEE MICRO*, 4(4):86–100, Aug. 1984.

[9] F. de Dinechin, A. V. Ershov, and N. Gast. Towards the post-ultimate libm. In *Proceedings of the 17th IEEE Symposium on Computer Arithmetic*, ARITH '05, pages 288–295, Washington, DC, USA, 2005. IEEE Computer Society.

[10] F. de Dinechin, C. Q. Lauter, and J.-M. Muller. Fast and correctly rounded logarithms in double-precision. *Theoretical Informatics and Applications*, 41:85–102, 2007.

[11] C. B. Dunham. Feasibility of "perfect" function evaluation. *SIGNUM Newsletter*, 25(4):25–26, Oct. 1990.

[12] L. Fousse, G. Hanrot, V. Lefèvre, P. Pélissier, and P. Zimmermann. MPFR: A Multiple-Precision Binary Floating-Point Library with Correct Rounding. *ACM Transactions on Mathematical Software*, 33(2), 2007. available at http://www.mpfr.org/.

[13] S. Gal and B. Bachelis. An accurate elementary mathematical library for the IEEE floating point standard. *ACM Transactions on Mathematical Software*, 17(1):26–45, Mar. 1991.

[14] S. W. Graham and G. Kolesnik. *Van der Corput's method of exponential sums*. Cambridge University Press, 1991.

[15] M. N. Huxley. *Area, lattice points, and exponential sums*, volume 13. Clarendon Press, 1996.

[16] IEEE Computer Society. *IEEE Standard for Floating-Point Arithmetic*. IEEE Standard 754-2008, Aug. 2008. available at http://ieeexplore.ieee.org/servlet/opac?punumber=4610933.

[17] C. Iordache and D. W. Matula. On infinitely precise rounding for division, square root, reciprocal and square root reciprocal. In Koren and Kornerup, editors, *Proceedings of the 14th IEEE Symposium on Computer Arithmetic (Adelaide, Australia)*, pages 233–240. IEEE Computer Society Press, Los Alamitos, CA, Apr. 1999.

[18] W. Kahan. Why do we need a floating-point standard? Technical report, Computer Science, UC Berkeley, 1981. Available at http://www.cs.berkeley.edu/~wkahan/ieee754status/why-ieee.pdf.

[19] E. Krätzel. *Lattice points*, volume 33 of *Mathematics and its Applications*. Springer, 1989.

[20] T. Lang and J.-M. Muller. Bound on run of zeros and ones for algebraic functions. In N. Burgess and L. Ciminiera, editors, *Proceedings of the 15th IEEE Symposium on Computer Arithmetic (ARITH-16)*, pages 13–20, June 2001.

[21] C. Q. Lauter. *Arrondi Correct de Fonctions Mathématiques*. PhD thesis, École Normale Supérieure de Lyon, Lyon, France, Oct. 2008. In French, available at http://www.ens-lyon.fr/LIP/Pub/Rapports/PhD/PhD2008/PhD2008-07.pdf.

[22] C. Q. Lauter and V. Lefèvre. An efficient rounding boundary test for $\mathrm{pow}(x, y)$ in double precision. *IEEE Transactions on Computers*, 58(2):197–207, Feb. 2009.

[23] V. Lefèvre. *Moyens Arithmétiques Pour un Calcul Fiable*. PhD thesis, École Normale Supérieure de Lyon, Lyon, France, 2000.

[24] V. Lefèvre and J.-M. Muller. Worst cases for correct rounding of the elementary functions in double precision. In N. Burgess and L. Ciminiera, editors, *Proceedings of the 15th IEEE Symposium on Computer Arithmetic (ARITH-16)*, pages 111–118, Vail, CO, June 2001.

[25] V. Lefèvre, J.-M. Muller, and A. Tisserand. Towards correctly rounded transcendentals. In *Proceedings of the 13th IEEE Symposium on Computer Arithmetic*, pages 132–137. IEEE Computer Society Press, Los Alamitos, CA, 1997.

[26] P. Markstein. The New IEEE-754 Standard for Floating-Point Arithmetic. In *Numerical Validation in Current Hardware Architectures*, number 08021 in Dagstuhl Seminar Proceedings, Dagstuhl, Germany, 2008. Internationales Begegnungs- und Forschungszentrum für Informatik (IBFI), Schloss Dagstuhl, Germany.

[27] H. L. Montgomery. *Ten lectures on the interface between analytic number theory and harmonic analysis*. Number 84 in Conference board of the Mathematical Sciences. American Math. Soc., 1994.

[28] J.-M. Muller. *Elementary Functions, Algorithms and Implementation*. Birkhäuser, Boston, 2016.

[29] J.-M. Muller, N. Brisebarre, F. de Dinechin, C.-P. Jeannerod, V. Lefèvre, G. Melquiond, N. Revol, D. Stehlé, and S. Torres. *Handbook of Floating-Point Arithmetic*. Birkhäuser, 2010. ACM G.1.0; G.1.2; G.4; B.2.0; B.2.4; F.2.1., ISBN 978-0-8176-4704-9.

[30] Y. V. Nesterenko and M. Waldschmidt. On the approximation of the values of exponential function and logarithm by algebraic numbers (in Russian). *Mat. Zapiski*, 2:23–42, 1996. Available in English at http://www.math.jussieu.fr/~miw/articles/ps/Nesterenko.ps.

[31] O. Robert. On van der Corput's k-th derivative test for exponential sums. *Indag. Math. (N.S.)*, 27(2):559–589, 2016.

[32] D. Stehlé. On the Randomness of Bits Generated by Sufficiently Smooth Functions. In F. Hess, S. Pauli, and M. E. Pohst, editors, *Algorithmic Number Theory, 7th International Symposium, ANTS-VII, Berlin, Germany, July 23-28, 2006, Proceedings*, volume 4076 of *Lecture Notes in Computer Science*, pages 257–274. Springer, 2006.

[33] D. Stehlé, V. Lefèvre, and P. Zimmermann. Worst Cases and Lattice Reduction. In J.-C. Bajard and M. J. Schulte, editors, *Proceedings of the 16th IEEE Symposium on Computer Arithmetic (ARITH-16)*, pages 142–147. IEEE Computer Society Press, Los Alamitos, CA, June 2003.

[34] D. Stehlé, V. Lefèvre, and P. Zimmermann. Searching Worst Cases of a One-Variable Function Using Lattice Reduction. *IEEE Transactions on Computers*, 54(3):340–346, Mar. 2005.

[35] G. Tenenbaum. *Introduction to analytic and probabilistic number theory*, volume 163. American Mathematical Society, 2015.

[36] J. D. Vaaler. Some extremal functions in Fourier analysis. *Bulletin of the American Mathematical Society*, 12(2):183–216, 1985.

[37] A. Ziv. Fast evaluation of elementary mathematical functions with correctly rounded last bit. *ACM Transactions on Mathematical Software*, 17(3):410–423, Sept. 1991.